



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERIA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE
COMUNICACIÓN

TEMA:

“DISEÑO DE UN PROTOTIPO DE MONITOREO DEL ESTADO DE SOMNOLENCIA EN CONDUCTORES MEDIANTE EL ESTUDIO DE LA ACTIVIDAD CEREBRAL EN LA ONDAS ALFA Y THETA.”

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

AUTOR: EDGAR ALEXANDER ICHAU IPIALES

DIRECTOR: MSC. CARLOS ALBERTO VÁSQUEZ AYALA

DIRECTOR: MSC. JAIME ROBERTO MICHILENA CALDERÓN

IBARRA-ECUADOR

2023



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICA A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003737465		
APELLIDOS Y NOMBRES:	Ichau Ipiales Edgar Alexander		
DIRECCIÓN:	Santa Rosa del Tejar		
EMAIL:	eaichau@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0992809868

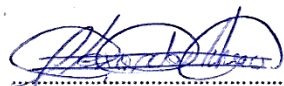
DATOS DE LA OBRA	
TÍTULO:	"DISEÑO DE UN PROTOTIPO DE MONITOREO DEL ESTADO DE SOMNOLENCIA EN CONDUCTORES MEDIANTE EL ESTUDIO DE LA ACTIVIDAD CEREBRAL EN LA ONDAS ALFA Y THETA"
AUTOR:	Ichau Ipiales Edgar Alexander
FECHA :	24/06/2023
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	Ingeniero en Electrónica y Redes de Comunicación
DIRECTOR:	MsC. Carlos Alberto Vásquez Ayala

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 24 días del mes de junio, 2023.

EL AUTOR:



Ichau Ipiales Edgar Alexander

CI: 1003737465



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

MAGÍSTER CARLOS VÁSQUEZ, DIRECTOR DEL PRESENTE TRABAJO
DE TITULACIÓN CERTIFICA:

Que el presente trabajo de Titulación: “DISEÑO DE UN PROTOTIPO DE MONITOREO DEL ESTADO DE SOMNOLENCIA EN CONDUCTORES MEDIANTE EL ESTUDIO DE LA ACTIVIDAD CEREBRAL EN LA ONDAS ALFA Y THETA.” Ha sido desarrollado por el señor Ichau Ipiales Edgar Alexander bajo mi supervisión.

Es todo cuanto puedo certificar en honor a la verdad.

A handwritten signature in blue ink, appearing to read 'Carlos Alberto Vásquez Ayala', is written over a horizontal dotted line.

MsC. Carlos Alberto Vásquez Ayala

DIRECTOR

DEDICATORIA

A mi familia.

AGRADECIMIENTOS

*Agradezco a Dios por siempre acompañarme en todo momento, y ser la luz que guía
cada uno de mis pasos.*

*Mi eterno agradecimiento a mis padres quienes siempre me inculcaron que con el
trabajo duro y la dedicación es posible cumplir los sueños y nunca hay que rendirse. Gracias
totales por depositar toda su confianza en mí, brindándome la seguridad necesaria para seguir
paso a paso con mi vida.*

*A mi hermana quien siempre me brinda su apoyo y consejos de manera incondicional en
cada momento.*

*Agradezco a los docentes de la carrera CIERCOM, en especial a mi director, el MsC.
Carlos Vásquez y a mi asesor, el MsC. Jaime Michilena quienes supieron compartir su
conocimiento y su tiempo para guiarme en el desarrollo y culminación de este trabajo de
titulación.*

*A mi compañeros y amigos que fueron parte de este proceso universitario con quienes
compartimos buenos y malos momentos, pero siempre brindándonos la motivación necesaria
para no decaer y seguir adelante.*

INDICE DE CONTENIDO

DEDICATORIA.....	IV
AGRADECIMIENTOS.....	V
INDICE DE CONTENIDO	VI
INDICE DE FIGURAS.....	XII
INDICE DE TABLAS.....	XVII
RESUMEN.....	XIX
ABSTRACT.....	XX
1 CAPITULO I: ANTECEDENTES.....	1
1.1 Tema	1
1.2 Problema.....	1
1.3 Objetivos	2
1.3.1 Objetivo general	2
1.3.2 Objetivos específicos	3
1.4 Alcance.....	3
1.5 Justificación.....	5
2 CAPITULO II: FUNDAMENTO TEÓRICO	7
2.1 El cerebro	7

2.1.1	Operatividad	7
2.1.2	Sistema nervioso central	7
2.2	<i>Ondas cerebrales</i>	8
2.2.1	Ondas alfa	10
2.2.2	Ondas beta	11
2.2.3	Ondas theta	12
2.2.4	Ondas delta.....	12
2.3	<i>Electroencefalograma (EEG)</i>	13
2.3.1	Captación EEG	13
2.3.2	Tipos de electrodos EEG	13
2.3.3	Sensores EEG	15
2.4	<i>El sueño</i>	22
2.4.1	Sueño de ondas lentas (no MOR)	22
2.4.2	Sueño paradójico (MOR)	23
2.5	<i>Somnolencia</i>	24
2.5.1	Concepto de somnolencia	24
2.5.2	Causantes del estado de somnolencia	24
2.5.3	Somnolencia en conductores	26
2.6	<i>Sistemas de detección de somnolencia</i>	27
2.6.1	Sistemas por parámetros.....	28
2.6.2	Sistemas por monitoreo cerebral	29
2.6.3	Sistemas por reconocimiento facial	29
2.7	<i>Tecnologías utilizadas para la detección de somnolencia</i>	30
2.7.1	El DSM de SITRACK	30
2.7.2	Optalert	30

2.7.3	Detector de fatiga Bosch	31
2.7.4	Vigo Technologies.....	31
2.8	<i>Software Libre</i>	32
2.8.1	Java	32
2.8.2	Python	33
2.9	<i>Hardware Libre</i>	35
2.9.1	Arduino	35
2.9.2	Raspberry pi.....	38
2.9.3	UDOO.....	39
2.10	<i>Estándar IEEE 802.15.1</i>	41
2.10.1	Arquitectura	41
2.10.2	Capas del protocolo Bluetooth.....	42
2.11	<i>Métodos de procesamiento de Datos</i>	42
2.11.1	Proceso KDD	42
2.11.2	Fases del proceso KDD	43
2.12	<i>Algoritmos de clasificación supervisada</i> :.....	44
2.12.1	Algoritmo KNN.....	45
2.12.2	Random Forest	46
3	CAPITULO III: DESARROLLO EXPERIMENTAL	50
3.1	<i>Metodología</i>	50
3.1.1	Modelo iterativo	50
3.2	<i>Fase 1: Análisis y especificación de requerimientos</i>	52
3.2.1	Situación actual	52
3.2.2	Definición de requerimientos.....	55

3.2.3	Requerimientos de Stakeholders	56
3.2.4	Requerimientos del sistema	59
3.2.5	Requerimientos de arquitectura	62
3.3	<i>Elección de hardware y software</i>	65
3.3.1	Elección de hardware	65
3.3.2	Elección de software	73
3.4	<i>Fase 2: Diseño</i>	74
3.4.1	Arquitectura del sistema	75
3.4.2	Diagrama de bloques.....	77
3.4.3	Diagrama esquemático del sistema.....	79
3.5	<i>Fase 3: Codificación y pruebas del sistema</i>	81
3.5.1	Bloque 1: Alimentación del sistema	81
3.5.2	Bloque 2: Captación EEG	84
3.5.3	Bloque 3: Comunicación	87
3.5.4	Bloque 4: Adquisición de datos EEG.....	89
3.5.5	Bloque 5: Procesamiento de datos.....	100
3.6	<i>Fase 4: Integración y testeo del sistema</i>	125
3.6.1	Bloque 6: Implementación	125
3.6.2	Testeo del prototipo.....	139
4	CAPITULO IV: PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS ...	140
4.1	<i>Delimitación del proceso de pruebas a ejecutar</i>	140
4.2	<i>Pruebas de cumplimiento de requerimientos: stakeholders, arquitectura, sistema</i>	141
4.2.1	Cumplimiento de Stakeholders	141
4.3	<i>Pruebas de funcionalidad por cada bloque</i>	144

4.3.1	Bloque 1: Alimentación	144
4.3.2	Bloque 2: captación EEG.....	148
4.3.3	Bloque 3: Comunicación	150
4.3.4	Bloque 4: adquisición y almacenamiento de los datos	153
4.3.5	Bloque 5: Procesamiento de datos.....	156
4.4	<i>Pruebas finales de implementación del prototipo completo.....</i>	<i>163</i>
4.5	<i>Análisis de resultados.....</i>	<i>168</i>
4.6	<i>Costo del Prototipo.....</i>	<i>169</i>
CONCLUSIONES.....		171
RECOMENDACIONES		173
REFERENCIAS.....		174
ANEXOS.....		181
<i>Anexo A.....</i>		<i>181</i>
	Encuesta.....	181
	Tabulación de encuesta	183
<i>Anexo B.....</i>		<i>188</i>
	Constancias de autorización de uso de datos.....	188
<i>Anexo C: CODIFICACIÓN</i>		<i>189</i>
	Adquisición de datos EEG en Arduino.....	189
	Código de Random Forest en Google Colab (Python).....	191
	Código final del prototipo para implementar en el Raspberry Pi	192
<i>Anexo D. Elaboración de placas.....</i>		<i>192</i>

Diagramas	192
Planchado de placas	193
Suelda de placas	193
Placas finales.....	193
<i>Anexo E. Fotografías de lo realizado</i>	<i>194</i>
<i>Anexo F. Imágenes de las pruebas de funcionamiento.....</i>	<i>196</i>
<i>Anexo G . Datasheet de dispositivos.....</i>	<i>199</i>
Mindwave Mobile 2	199
Bluetooth hc-05	199
Arduino mega	199
Raspberry Pi 3B+	199

INDICE DE FIGURAS

Figura 1 Partes SNC	8
Figura 2 Ondas EEG	9
Figura 3 Neurosky Mindwave	16
Figura 4 Emotiv Insight	18
Figura 5 Diadema Muse 2.....	19
Figura 6 iFocusBand.....	20
Figura 7 Dispositivo OpenBCI	22
Figura 8 Arduino nano.....	36
Figura 9 Arduino uno R3.....	37
Figura 10 Arduino Mega.....	38
Figura 11 Raspberry pi	39
Figura 12 Placa UDOO.....	40
Figura 13 Proceso KDD.....	43
Figura 14 Algoritmo KNN.....	45
Figura 15 Modelo Random Forest	48
Figura 16 Modelo Iterativo	51
Figura 17 Arquitectura del sistema.....	76
Figura 18 Diagrama de bloques de funcionamiento del sistema	77
Figura 19 Diagrama esquemático del sistema	79
Figura 20 Diagrama de flujo del sistema general	80
Figura 21 Alimentación del sistema	82
Figura 22 Diagrama de flujo bloque de alimentación.....	83

Figura 23 Componentes de Neurosky Mindwave Mobile 2	84
Figura 24 Identificación estados del sensor, izquierda sensor no conectado y derecha sensor conectado.	85
Figura 25 Diagrama de bloque de captación de datos	86
Figura 26 Sincronización bluetooth.....	87
Figura 27 Configuración módulo hc-05 con neurosky	88
Figura 28 Comunicación bluetooth.....	88
Figura 29 Estructura de paquete ThinkGear	89
Figura 30 Estructura carga útil.....	90
Figura 31 Codificación de estructura del paquete ThinkGear	92
Figura 32 Activación de envío de información ThinkGear	93
Figura 33 Impresión de datos obtenidos del sensor Neurosky	94
Figura 34 Izquierda usuario colocado sensor correctamente, derecha datos de calidad de señal y atención.....	95
Figura 35 Verificación de calidad de señal en el peor estado, izquierda usuario colocado sensor sin pinza eliminadora de ruido, derecha datos obtenidos de calidad de señal.	95
Figura 36 Lectura de datos del sensor Neurosky mediante monitor serial de Arduino... ..	96
Figura 37 Diagrama de flujo de la adquisición de datos.....	97
Figura 38 Interfaz CoolTerm	98
Figura 39 Izquierda configuraciones comunicación serial, derecha validación de conexión.....	99
Figura 40 Izquierda adquisición de datos EEG, derecha almacenamiento de BDD por usuario.....	99

Figura 41 Señal de atención escenario vigila.....	101
Figura 42 Señal theta escenario vigila	102
Figura 43 Señal Alpha escenario vigila	102
Figura 44 señal beta escenario vigila.....	102
Figura 45 señal gamma escenario vigila.....	103
Figura 46 señal atención escenario relajación	105
Figura 47 señal theta escenario relajación	105
Figura 48 señal alfa escenario relajación.....	105
Figura 49 señal beta escenario relajación	106
Figura 50 señal gamma escenario relajación	106
Figura 51 señal atención escenario sueño.....	108
Figura 52 señal theta escenario sueño.....	108
Figura 53 señal alfa escenario sueño	109
Figura 54 señal beta escenario sueño.....	109
Figura 55 señal gamma escenario sueño.....	109
Figura 56 Datos atención filtrados por Media Móvil	112
Figura 57 Filtro Savitzky Golay en atención	113
Figura 58 Filtro Savitzky Golay en señal theta.....	113
Figura 59 Señal atención con filtro FIR.....	114
Figura 60 Señal theta con filtro FIR	115
Figura 61 Diagrama de bloques de filtro digital	119
Figura 62 Instalación de LazyPredict	120
Figura 63 Importación de algoritmos de Machine Learning	121

Figura 64 Determinación de etiquetas	122
Figura 65 Variables de entrenamiento y prueba para ML	123
Figura 66 Exactitud de algoritmos de ML.	123
Figura 67 Proceso de algoritmo de clasificación con Lazy Predict.....	124
Figura 68 Implementación de prototipo.....	125
Figura 69 Diseño case nodo central.....	126
Figura 70 Elaboración e implementación case nodo central, parte inferior bloque de comunicación, parte superior bloque de procesamiento de datos.	127
Figura 71 Case nodo principal completo, en la parte inferior se observa la fuente de alimentación y la parte superior el case del prototipo.....	128
Figura 72 Ubicación de alerta.....	129
Figura 73 Instalación de librerías.....	130
Figura 74 Determinación de variables x para columnas y variable y para etiquetas.	131
Figura 75 Verificación de cuantos valores existen para cada etiqueta	131
Figura 76 BDD balanceada para etiqueta de somnolencia (1) y vigila (0).....	132
Figura 77 Determinación de datos para entrenamientos como para prueba	133
Figura 78 Número de datos de entrenamiento y prueba	133
Figura 79 Creación de algoritmo de clasificación Random Forest.....	134
Figura 80 Entrenamiento del algoritmo.....	134
Figura 81 Comparación de predicciones dadas por el algoritmo entrenado.....	135
Figura 82 Verificar nivel de exactitud del algoritmo.....	135
Figura 83 Identificación de métricas del algoritmo	136
Figura 84 Matriz de confusión.....	136

Figura 85 Programación de acciones	137
Figura 86 Diagrama de flujo software de Random Forest.....	138
Figura 87 Implementación prototipo en ambiente real.....	139
Figura 88 Medición de voltaje en Arduino mega	146
Figura 89 Alimentación de modulo hc-05	146
Figura 90 Línea de tiempo de voltaje	147
Figura 91 Estados del sensor EEG, izquierda sensor no conectado, derecha sensor conectado.	150
Figura 92 Conexión de bloque de comunicación.....	152
Figura 93 Etapa de emparejamiento bluetooth	152
Figura 94 Comunicación entre el sensor y modulo hc-05 correcta	153
Figura 95 Datos emitidos por el sensor Neurosky Mindwave Mobile 2 en el Arduino	155
Figura 96 Izquierda datos EEG, derecha almacenamiento en .txt con COOLTERM ...	156
Figura 97 Señales en estado de somnolencia(amarrillo) y vigila (verde) de la BDD total	159
Figura 98 Datos recibidos por el Raspberry desde el Arduino	160
Figura 99 Identificación de etiqueta de predicción del algoritmo Random Forest.....	161
Figura 100 Matriz de confusión.....	161
Figura 101 Toma de decisión de Random Forest	162
Figura 102 Izquierda predicción estado despierto(vigila), derecha predicción estado somnolencia (Alerta).....	166
Figura 103 Grafica de pruebas etapa 2	167

INDICE DE TABLAS

Tabla 1 Análisis situación actual	54
Tabla 2 Nomenclatura de requerimientos	56
Tabla 3 Lista de Stakeholders	57
Tabla 4 Niveles de prioridad de requerimientos	57
Tabla 5 Requerimientos Operacionales y de usuarios.	58
Tabla 6 Requerimientos del Sistema.....	60
Tabla 7 Requerimientos de software, hardware y eléctricos que determinan la arquitectura del sistema.....	62
Tabla 8 Puntuación de requerimientos.....	65
Tabla 9 Elección sensor neuronal	66
Tabla 10 Características Neurosky Mindwave Mobile2.....	67
Tabla 11 Selección módulo bluetooth.....	68
Tabla 12 Características técnicas modulo bluetooth HC-05.....	69
Tabla 13 Selección de dispositivo para la adquisición de datos	70
Tabla 14 Características técnicas Arduino mega	71
Tabla 15 Dispositivo de procesamiento de datos.....	72
Tabla 16 Características Raspberry Pi 3	73
Tabla 17 Selección de software de programación	74
Tabla 18 Consumo de corriente y voltaje del sistema	81
Tabla 19 Códigos ThinkGear	91
Tabla 20 BDD estado vigila.....	100
Tabla 21 BDD estado de relajación	104

Tabla 22	BDD estado de sueño.....	107
Tabla 23	Valores SRN, SDT de BDD estado de vigila	111
Tabla 24	Calculo SRN, SDT de BDD sueño	111
Tabla 25	Relación de validación de filtro	115
Tabla 26	Tabla comparativa señal original vs señal filtrada.....	116
Tabla 27	Valor digital de etiquetas	122
Tabla 28	Pruebas de funcionamiento	140
Tabla 29	validación de requerimientos	141
Tabla 30	Validación de Stakeholders.....	142
Tabla 31	Validación de requerimientos del sistema	142
Tabla 32	Validación de requerimientos de arquitectura	143
Tabla 33	Test del bloque de alimentación	144
Tabla 34	Test bloque captación EEG.....	148
Tabla 35	Test de comunicación	150
Tabla 36	Test de adquisición y almacenamiento de datos	153
Tabla 37	Test de bloque de procesamiento.....	156
Tabla 38	Cronograma de pruebas finales.....	164
Tabla 39	Pruebas etapa 1	165
Tabla 40	Pruebas etapa 2	166
Tabla 41	Porcentaje de datos de prueba.....	168
Tabla 42	Costo prototipo.....	170

RESUMEN

El presente trabajo trata de un prototipo el cual permite la detección del estado de somnolencia en conductores, utilizando un sensor que permite monitorear la actividad cerebral en las distintas ondas como lo son Alpha, theta, delta, beta, gamma y el nivel de atención para después ser sometido a un proceso de machine learning que permita identificar el estado del conductor en base a los datos obtenidos. El prototipo es desarrollado con el propósito que su implementación sea lo menos invasiva posible para el conductor y así no entorpezca ni obstruya ninguna de sus actividades.

Es fundamental la determinación de los requisitos del proyecto para realizar una correcta elección de los elementos usados en el mismo, de igual manera importante tomar en cuenta el análisis de la situación actual de donde parte la determinación de los parámetros para realizar el diseño del prototipo. En cuanto al funcionamiento, el prototipo monitorea la actividad cerebral del usuario mediante el sensor EEG, estos datos son emitidos al microcontrolador, mismo que envía los datos a la placa de procesamiento Raspberry la cual realiza el filtrado respectivo de los datos para así realizar la comparación con el algoritmo Random Forest previamente entrenado si los datos que se obtienen del usuario pertenecen a la etiqueta 0 indica que el usuario se encuentra en vigila mientras que si los datos pertenecen a la etiqueta 1 el prototipo detecta que se encuentra en estado de somnolencia por tanto se activa de manera inmediata la alerta auditiva. El prototipo va implementado en el habitáculo del automóvil con el propósito de realizar las pruebas de funcionamiento en tiempo real para así comprobar el correcto funcionamiento del sistema. Por tanto, los resultados que se obtuvieron, concluyen que se evidencia que los conductores son propensos a caer en estado de somnolencia en los horarios de altas horas de la noche como también en la madrugada.

ABSTRACT

The present work deals with a prototype which allows the detection of the state of drowsiness in drivers, using a sensor that allows monitoring the brain activity in the user in the different waves such as Alpha, theta, delta, beta, gamma as well as the level of attention to later be subjected to a machine learning process that allows the driver's status to be identified based on the data obtained. The prototype is developed with the purpose that its implementation is the least invasive possible for the driver and thus does not hinder or obstruct any of their activities.

It is essential to determine the requirements of the project to make a correct choice of the elements used in it, in the same way important to take into account the analysis of the current situation from which the determination of the parameters to carry out the design of the prototype starts. Regarding the operation, the prototype monitors the user's brain activity through the EEG sensor, these data are sent to the microcontroller itself that sends the data to the Raspberry processing board which performs the respective filtering of the data in order to make the comparison with the previously trained Random Forest algorithm, if the data obtained from the user belongs to label 1, the prototype detects that it is in a state of drowsiness, therefore the auditory alert is immediately activated; The prototype is implemented in the passenger compartment of the car with the purpose of carrying out functional tests in real time in order to verify the correct functioning of the system. Therefore, the results that were obtained conclude that it is evident that drivers are prone to fall into a state of drowsiness in the late hours of the night as well as in the early morning. Therefore, the prototype is perfectly functional in the work area because the auditory alarm system works correctly when it detects the possible state of drowsiness in the driver.

CAPITULO I: ANTECEDENTES

1.1 Tema

Diseño de un prototipo de monitoreo del estado de somnolencia en conductores mediante el estudio de la actividad cerebral en la ondas alfa y theta.

1.2 Problema

La fatiga y el estado de somnolencia reduce en gran medida la capacidad de vigila en la persona por tal motivo los sentidos se ven afectados, personas que se encuentran en este estado tienen indicadores fisiológicos los cuales se denotan un cansancio físico lo cual produce: el bostezo, parpadeo, la falta de atención, una baja respuesta en acciones que demandan una reacción inmediata (Veron, 2018). Por lo cual el desgaste mental que el conductor sufre al encontrarse en un estado de fatiga o somnolencia es motivo para que los accidentes que se ocasionan tengan un alto índice lesiones graves en los pasajeros como también de mortalidad.

Según los reportes de la Agencia Nacional de Tránsito (ANT, 2021) en Ecuador en lo que va desde enero a septiembre 2021 se presenta un total de 3166 accidentes de tránsito, por tanto, los vehículos livianos se ven involucrados en un alto índice de 1154 lo que equivale a un 36.44% del total de siniestros. Los accidentes causados debido que el conductor se encuentre en un estado tanto físico como mental bajo declina a un estado de somnolencia el cual ocupa el 0.6% del total de siniestros en la carretera siendo así uno de los 10 principales factores de accidentes de tránsito. El desconocimiento de este factor tan importante para el conductor trae gran cantidad de consecuencia de gravedad sumamente alta. A su vez actualmente existen muchas técnicas de aplicabilidad en la cual se puede detectar el estado de somnolencia en la persona una de ellas es mediante rasgos fisiológicos mediante un análisis por visión artificial, otra técnica muy eficaz para la detección de este factor en el conductor es mediante el análisis de electroencefalograma el

cual realiza un monitoreo de la actividad cerebral en las etapas del sueño en específico el análisis de las ondas alfa y theta en donde se manifiesta la somnolencia en la persona por los diferentes ambientes (B105 LAB, 2017).

Los vehículos de gama media y baja son los más habituales en sufrir accidentes por la fatiga en el conductor debido a que actualmente no cuentan con algún tipo de asistente de conducción el cual permita la monitorización del estado en el que se encuentra el conductor y alertar de ser necesario debido a los altos costos que tienen la implementación de estas nuevas tecnologías en los vehículos. La gran diferencia con los vehículos de gama alta como lo son tanto Mercedes Benz el cual cuenta con un asistente que analiza el manejo del volante por parte del conductor para detectar síntomas típicos de agotamiento y una falta de atención acusada, y advierte con señales ópticas y acústicas del peligro de micro sueños (MERCEDES-BENZ).

Por ende, se plantea el desarrollo de un prototipo con el cual permitirá la monitorización del estado conductor en tiempo real mediante el análisis de la electroencefalografía (EEG) en las ondas alfa, beta, teta detectando así de manera temprana un posible adormecimiento del conductor, determinando si este se encuentra en un estado de somnolencia y proceder a alertar al mismo para evitar accidentes en la carretera.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un prototipo electrónico con el cual se permita la monitorización de parámetros adecuados para la identificación del estado de somnolencia en el conductor mediante el estudio de la actividad cerebral en las ondas alfa y theta.

1.3.2 *Objetivos específicos*

Realizar una investigación bibliográfica del estado del arte con respecto al monitoreo de la somnolencia en la persona mediante la tecnología EEG.

Diseñar el prototipo basado en el modelo iterativo, por medio del uso de dispositivos que permitan la obtención de datos EEG, para la detección del estado de somnolencia en el conductor.

Implementar el prototipo para el monitoreo del estado de somnolencia en el conductor.

Ejecutar las respectivas pruebas de funcionamiento, en donde se permita la correcta validación de la detección de la somnolencia en el conductor.

1.4 Alcance

En el actual proyecto se tiene como el alcance, realizar el diseño e implementación de un prototipo electrónico con el cual se permita el monitoreo y detección del estado de somnolencia del conductor mediante un análisis de la actividad cerebral mediante la utilización de un dispositivo que capte dicha actividad. Para lo cual se utilizará la metodología iterativa la cual consta de 4 etapas que son el análisis de requerimientos, diseño, codificación y pruebas del sistema (Martins, 2021), dichas etapas se detallaran a continuación para el desarrollo y ejecución del proyecto.

En la primera etapa se realiza el análisis de requerimientos a utilizar, por lo cual es necesario la previa recopilación de información sobre la actividad cerebral, el sueño en conductores y así determinar cuáles son los parámetros necesarios para la detección del estado de somnolencia como lo son las ondas cerebrales tanto alfa y theta las cuales son indicadores de las etapas del sueño. Por lo cual para considerar los requerimientos adecuados se utilizará el

estándar ISO/ IEC/IEEE 29148 para la elección del software que se adapte a la necesidad del prototipo, para el hardware se utilizará por medio de la elección de requerimientos de Stakeholders en el cual permitirá la elección de dispositivos de acuerdo con las necesidades que se van planteando a lo largo de la realización del proyecto.

En la segunda etapa se procede con el diseño del prototipo el cual se utilizará un sensor EEG con el cual se puede captar los datos de la actividad cerebral en las ondas Alfa y theta debido a que estas ondas son las indicadores de un inicio del estado de relajación y por ende del sueño, los datos obtenidos desde este dispositivo serán enviados de manera inalámbrica con el estándar Bluetooth (802.15.1), estos datos serán recibidos por un módulo de comunicación inalámbrico para luego ser procesados en software y hardware adecuado, una vez los datos sean captados estos pasaran por el proceso KDD en el cual la información será tratada y a su vez procesada para proceder a analizarla mediante algoritmos de aprendizaje autónomo supervisado con el cual se realiza la determinación si el conductor se encuentra en un estado de somnolencia o de vigila y como último paso proceder a la interpretación de la información tratada si se encuentra en estado de somnolencia en las onda theta y alfa baja enviar una alerta al conductor que permita recuperar el estado de vigila de este. Se utilizará el estándar ISO/ IEC/IEEE 29148 para la elección del software y para el hardware por medio de la elección de requerimientos de Stakeholders.

En la tercera etapa se procede a realizar la implementación del prototipo en este caso se la realizará en un ambiente interno de un automóvil Aveo y así proceder a realizar pruebas validación de la codificación del algoritmo de aprendizaje autónomo escogido para proceder a validar el funcionamiento de este con una correcta determinación del estado de somnolencia en el conductor.

En la cuarta etapa se realizará las pruebas de funcionamiento de todo el prototipo, es decir, tanto el algoritmo de aprendizaje y determinación si el conductor se encuentra en un estado somnolencia como también la correspondiente alerta si el conductor se encuentra en este estado. La correcta validación de este tipo de sistemas permitirá que los vehículos de gama baja y media que no cuentan con algún tipo de asistentes de conducción puedan acceder a estos y así lograr una ayuda para los conductores dando mayor seguridad en la carretera

1.5 Justificación

El transporte ya sea de uso privado como público es indispensable en la sociedad por tal razón la cantidad de accidentes es muy alta según las estadísticas de la ANT en el cual los automóviles son directamente el porcentaje más alto en estos siniestros (ANT, 2021), dentro de estos siniestros una de las principales causantes es que el conductor pierda su estado de vigila por tal motivo la concentración disminuya en el volante y se reduzca notablemente la reacción del conductor ante posibles situaciones en la carretera lo cual denote en accidentes de tránsito.

Con el presente proyecto se busca realizar un asistente de conducción con el cual se permita la detección temprana de un posible caso de somnolencia en el conductor y así proceder a alertar al mismo para que este pueda recuperar su estado de vigila y concentración al volante, con este sistema se ayuda al conductor el alertar un posible estado de baja concentración y somnolencia de esta manera este pueda tomar la decisión de descansar de ser el caso y seguir con su ruta de destino.

En este proyecto se busca el beneficio para autos de gama media y baja, como lo es el modelo Aveo dado que según datos de la revista Autos Magazine que en diciembre de 2020 se registraron 1'267.640 vehículos publicados para su venta en donde los que más demanda de

busca y compra son el modelo Aveo (AutosMagazine, 2020). Mismos que no cuentan con un sistema de asistencia en conducción para el control del adormecimiento temprano o estado de somnolencia en el conductor y al ser un vehículo tan habitual en las carreteras se realizará la implementación de este sistema de monitoreo de somnolencia el cual puede ser compatible con modelos similares de gama media y baja para su implementación ayudando así a mejorar la seguridad en las carreteras tanto de conductores como también de los pasajeros.

CAPITULO II: FUNDAMENTO TEÓRICO

Para este capítulo se realizará la adquisición de información tanto técnica como científica sobre los parámetros más importantes en las etapas del sueño mediante un estudio de EEG y como también en cuál de estas etapas se puede determinar el estado de somnolencia o un comienzo de adormecimiento en la persona. La importancia de tener conocimiento del tema a investigar como también que tecnologías son factibles para utilización y así llevar a cabo el desarrollo del problema de estudio.

2.1 El cerebro

2.1.1 *Operatividad*

El cerebro trabaja mediante el envío y recepción de señales tanto químicas y eléctricas que van dirigidas a todo el cuerpo. Estas señales son las encargadas de controlar los diferentes procesos como lo son actividades realizadas o estados de ánimo entre otros. (Johns Hopkins University, 2021)

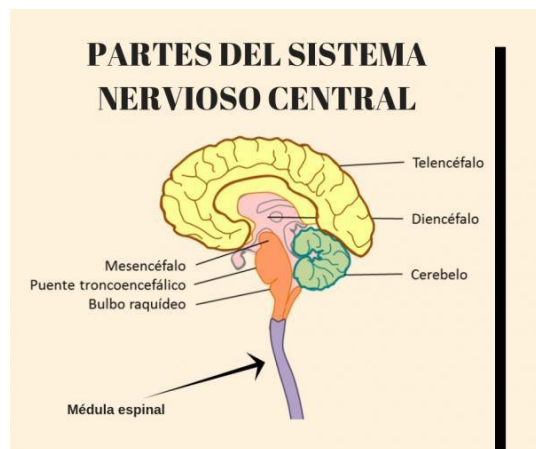
2.1.2 *Sistema nervioso central*

El sistema nervioso es aquel quien cumple con la función tanto de enviar como recibir distintas señales y estímulos en todo el cuerpo. Por ende, el sistema nervioso central se lo conoce por ser la parte central del procesamiento mental debido a que este es el que realiza el tratamiento de la información.

Por tal motivo, el sistema nervioso constituye de diversas partes que lo compone, como se muestra en la Figura 1; estas partes son: Telencéfalo, Diencefalo, Cerebelo, bulbo, mesencéfalo, medula espinal, tronco encefálico, los cuales al trabajar en conjunto son fundamentales porque permiten captar las señales y poder procesarlas de mejor manera. (Laguna , 2022)

Figura 1

Partes SNC



Nota. El SNC está constituido por diferentes partes las cuales son muy importantes pero el cerebro es el órgano más grande para el sistema nervioso central. Adaptada de *Sistema Nervioso Central*:

Funciones y partes [Fotografía], psicología-online, 2019, www.psicologia-online.com.

2.2 Ondas cerebrales

Es la actividad eléctrica la cual es producida por el cerebro y esta es detectada por un electroencefalograma. Las mismas que varían según lo que estemos realizando y sintiendo.

Tipos de Ondas Cerebrales

Las ondas EEG son de naturaleza eléctrica y las mismas están ubicadas en la corteza cerebral. Cada neurona constituye en dipolo eléctrico diminuto, el cual su polaridad depende con el impulso enviado a la célula sea de tipo excitatorio u inhibitorio. La obtención y registro de una señal de actividad eléctrica mediante el análisis en cada región cerebral a través de la superficie craneal en el cual se procede mediante la colocación de diversos electrodos mismos que permiten la captación de la diferencia de potencial entre ellos. (Ramos, 2009, pg. 2)

Las ondas EEG sin procesar generalmente son puntualizados en términos de bandas de frecuencias que son: Gamma (39-42 Hz) Beta (13-38 Hz), alfa (8-12 Hz), theta (4-8 Hz) y delta (menos de 4 Hz).

En la se puede observar las ondas EEG las cuales dan a conocer el estado de la actividad mental en el que está la persona dependiendo del tipo de actividad o ambiente en el que se este se encuentra, emitiendo rangos de frecuencias distintos.

En la **Figura 2**, se puede observar las ondas EEG las cuales dan a conocer cuál es el estado de actividad mental en el que está la persona dependiendo del tipo de actividad o ambiente en el que esta se encuentra, emitirá rangos de frecuencia distintos.

Figura 2

Ondas EEG

Ondas cerebrales	Frecuencia	Estado mental
Onda delta	0,5 - 3 Hz	sueño profundo
Onda theta	4 - 7 Hz	sueño ligero
Onda alfa	8 - 13 Hz	despierto, relajado
Onda beta	14 Hz	despierto, excitado

Nota. Las ondas EEG son determinadas por el rango de frecuencia en que estas se encuentran,

Adaptada de *Electroencefalograma EEG* [Fotografía], hipertextual, 2012, www.hipertextual.com

2.2.1 Ondas alfa

Estas ondas son aquellas que oscilan dentro del rango de 8 a 13 (Hz). Las mismas que tienen su valor más alto el cual se encuentra alrededor de los 10 Hz. Al estar la persona en una buena producción de ondas alfa ayuda al ingenio mental y la coordinación el individuo puede realizar cualquier tarea. Este tipo de onda es común que se encuentre el cerebro porque aparece cuando una persona se encuentra activa, mismo que puede ser un marcador de los estados de alerta y sueño. Cuando su alfa está dentro de los rangos normales, también tendemos a experimentar buen humor, ver el mundo con sinceridad y tener una sensación de calma. Una de las ondas más importantes en el cerebro es la de Alpha debido que en esta onda permite el aprender y el uso de la información que habitualmente se enseña en el aula y en el trabajo. Puede aumentar alfa cerrando los ojos o respirando profundamente o disminuir alfa pensando o calculando. El entrenamiento Alpha-Theta puede crear un aumento en la sensación, pensamiento abstracto y autocontrol. En nuestro escenario de automóvil, Alpha representaría neutral o inactivo. Alpha nos permite cambiar fácilmente de una tarea a otra. (NeuroHealth Associates, 2013)

- ALFA (8-12 Hz): Distribución: regional, generalmente involucra todo el lóbulo; occipital fuerte con ojos cerrados.

Estados emocionales subjetivos: relajado, no agitado, pero no somnoliento; tranquilo, consciente

Tareas y comportamientos asociados: meditación, sin acción

Correlatos fisiológicos: relajado, curativo

Efectos del entrenamiento: puede producir relajación

- Sub banda alfa bajo: 8-10: conciencia interna de uno mismo, integración mente/cuerpo, equilibrio

- Sub-banda alfa alto: 10-12: centrado, sanación, conexión mente/cuerpo.

2.2.2 *Ondas beta*

Este tipo de ondas por lo frecuente son de una actividad rápida la cual su rango de operación es de 14 Hz o mayor esto básicamente indica un estado de alerta. Cuando la persona se está generando dichas ondas se considera que se encuentra en un ritmo normal o en un momento de alerta o ansiedad, este es el estado en el que se encuentra el cerebro la mayor parte cuando tenemos los ojos abiertos y se realiza diversas actividades como escuchar, resolver problemas, toma de decisiones y el proceso de la información entre otras más.

Estas ondas al tener un rango de actividad sumamente extenso se encuentran clasificadas en tres tipos las cuales se dividen por los estados emocionales como fisiológicos que la persona se encuentra, determinando el rango de esta. (NeuroHealth Associates, 2013)

- Low Beta (12-15 Hz): Cuando se encuentra en este rango de actividad cerebral es debido a que la persona se encuentra relajado pero enfocado, integrado y todos los comportamientos coligados.
- Mid Beta (15-18 Hz): El rango intermedio para esta onda en la cual la persona se encuentra en un estado consciente, pensativo, alerta y activo, pero no agitado.
- High Beta (superior a los 18 Hz): Cuando el cerebro se encuentra produciendo este tipo de ondas se los puede localizar mediante el estado emocional de la persona que se encuentre en constante alerta o agitación; cómo también el estar realizando actividades mentales arduas producen la activación de este estado de onda beta en high.

2.2.3 Ondas theta

Estas ondas ocurren de manera más común durante el sueño, cómo también en esta onda se predomina la meditación de manera profunda. La onda theta es la puerta para el aprendizaje, de la memoria y la intuición. Cuando se encuentra en esta onda los sentidos se retiran del mundo externo y los mismo se focalizan en las señales que se originan desde adentro.

Este es el estado vespertino que solamente sucede por momentos fugaces que los podemos experimentar cuando nos despertamos o nos quedamos dormidos. En theta nos encontramos en un sueño; imágenes vividas, información que va más allá de nuestra conciencia normal. Es en donde guardamos cosas sumamente abstractas, nuestros miedos y pesadillas.

La onda theta se encuentra en actividad a una frecuencia de 3,5 a 7,5 Hz misma que se clasifica como actividad "lenta". Se encuentra correlacionada con la creatividad, la intuición, el soñar despierto y fantasear y es un depósito de recuerdos, emociones, sensaciones. Las señales theta durante el enfoque interno son fuertes. Debido a que se refleja el estado del sueño y la vigila lo que es relacionado con la mente en el estado subconsciente. La onda no es habitual en adultos despiertos, a diferencia que en niños hasta 13 años la presencia de esta onda es habitual.

Esta onda también es normal durante la etapa del sueño. Se dice que la onda theta refleja la actividad del sistema límbico (regula las emociones) y las regiones del hipocampo. La onda theta se la puede divisar la ansiedad. (NeuroHealth Associates, 2013)

2.2.4 Ondas delta

Son ondas cerebrales comúnmente lentas y ruidosas debido a que son de una frecuencia baja (menores de 4 Hz) estas ondas aparecen cuando la persona se encuentra en el sueño

profundo y en algunos procesos anormales. Frecuentemente estas ondas tienden a predominar hasta el año en lactantes; además para las etapas 3 y 4 de sueño están ondas aparecen.

Habitualmente se identifica a estas ondas por comportamientos, estado emocional y actividad de la persona como lo son el no moverse, sueño profundo, sueño no REM determinan que la persona se encuentra en ondas delta.

2.3 Electroencefalograma (EEG)

El electroencefalograma (EEG) trata en registro del cambio de la diferencia de potencial las cuales son producidas por las células cerebrales. La actividad eléctrica del cerebro es esencialmente una forma especializada del fenómeno general de la electricidad de toda la célula viviente. (Uribe & Arana, 2010)

2.3.1 Captación EEG

La actividad eléctrica que produce el cerebro se la puede captar actualmente con algunos métodos como:

- Sobre el cuero cabelludo.
- En la base del cráneo.
- En cerebro expuesto.
- En ubicaciones profundas cerebrales.

2.3.2 Tipos de electrodos EEG

2.3.2.1 Electroodos húmedos.

Los electrodos húmedos son aquellos en donde la impedancia electrodo-piel es un factor importante que determina las propiedades eléctricas de la interfaz electrodo-piel. Cuanto menor sea la impedancia electrodo-piel, mejor será la calidad de las señales EEG. Además, la baja

impedancia de la piel del electrodo puede ayudar a reducir la interferencia de la línea eléctrica y también hacer que las señales de EEG sean más inmunes a los artefactos de movimiento, incluido el movimiento del cable. Hoy en día, el electrodo más utilizado es el electrodo de plata/cloruro de plata (Ag/AgCl). Los electrodos de Ag/AgCl, también conocidos como electrodos húmedos, usan un gel de electrolito para formar una ruta conductora entre la piel y el electrodo para reducir la impedancia entre el electrodo y la piel. Los electrodos húmedos son el estándar de oro actual en la práctica clínica, sin embargo, tienen varias desventajas.

- Puede llevar mucho tiempo reducir la impedancia a un valor aceptable (5–20 $K\Omega$).
- Una vez que se alcanza una impedancia aceptable, el gel puede secarse en unas pocas horas, aumentando nuevamente la impedancia. Por ejemplo, se observó que la impedancia de los electrodos húmedos se deterioró de 5 a 15 $K\Omega$ dentro de las 5 horas posteriores a la aplicación del gel. Por lo tanto, los electrodos húmedos no son adecuados para EEG a largo plazo.
- El uso de muchos electrodos (es decir, EEG de conjunto denso) también puede ser problemático, porque la distancia entre los electrodos se reduce a medida que aumenta el número de electrodos. Esto puede hacer que los geles electrolíticos formen un puente conductor entre dos electrodos.
- Configurar el EEG con electrodos húmedos lleva mucho tiempo (preparación de la piel, esperar a que la impedancia alcance un valor aceptable, etc.) y requiere un profesional.

- Finalmente, el uso de pasta abrasiva (como parte de la preparación del cuero cabelludo) y el gel electrolítico puede ser inconveniente para el sujeto.

(Subramaniyam, 2021)

2.3.2.2 Electrodo en seco

Este tipo de electrodo está compuesto de un solo metal el cual trabaja como conductor interconectando tanto a la piel con el electrodo siendo de un material que es de acero inoxidable, además permite ser una alternativa bastante usada misma que tiene tantas ventajas como desventajas como por ejemplo este tipo de electrodos son propensos a tener niveles de ruido mucho más altos que con los electrodos húmedos. Pero si estos electrodos se encuentran colocados de una manera correcta, teniendo un contacto firme con la piel y el electrodo se puede determinar cómo confiable los datos EEG espectrales o sin pre-amplificación. El uso de los electrodos en seco permite sobrellevar las diversas actividades de investigación que son tratadas en ambientes internos a escenarios externos permitiendo llevar a cabo mediciones EEG en un escenario de mundo real. (BuscaEU, 2019)

2.3.3 Sensores EEG

Gracias a los avances tecnológicos que en los últimos años de investigación de neurociencia permite tener mayor conocimiento acerca del cerebro, principalmente en el análisis de las señales eléctricas que son enviadas por las neuronas que estas se producen en el cerebro, gracias a estos avances tecnológicos se puede conocer y aproximar los patrones y frecuencias de las señales emitidas por el cerebro mediante el uso de sensores los cuales van en el cuero cabelludo del usuario y así se puede monitorizar el comportamiento cerebral que la persona en cuestión tiene, el resultado es diferente dependiendo de las actividades que este mismo realice.

2.3.3.1 Mindwave Neurosky

El Mindwave Mobile 2 de la empresa Neurosky, es una diadema EEG auricular la cual permite la medición y la transferencia de los datos del espectro de potencia denominadas ondas EEG (ondas alfa, beta, teta, etc.) la misma que cuenta con Bluetooth con baja consumo de energía (BLE) para comunicarse de una forma inalámbrica con diversos dispositivos como son dispositivos Android, iOS o computadores. Con esta diadema EEG se puede controlar como trabajan las ondas y los diferentes cambios en tiempo real además que permite el monitoreo y control de los niveles de atención y relajación de la persona, permite también el aprender cómo es que el cerebro reacción al escuchar su música favorita. Este dispositivo es extraordinario para el ingreso del mundo de la interfaz cerebro computadora la cual permite realizar investigaciones sumamente interesantes. (Neurosky.com, 2014)

En la **Figura 3**, se puede observar que el mindwave neurosky es de un modelo ergonómico el cual es muy fácil su utilización y así hace más cómoda la experiencia de su uso el cual cuenta con su sensor principal el cual va ubicada en la frente del usuario como también su pinza la cual es de la tierra para este dispositivo.

Figura 3

Neurosky Mindwave



Nota. Neurosky Mindwave Mobile 2 es un auricular inalámbrico el cual permite la captura de la actividad cerebral en ondas EEG, Adaptada de *Neurosky Mindwave* [Fotografía], Eyecontec, 2019, www.eyecomtec.com

2.3.3.1.1 Medidores eSense

En todos los tipos de eSense es decir los algoritmos que permiten medir parámetros como (atención, meditación), el valor que brinda este medidor se lo determina en una escala que va de 1 a 100. En esta escala, un valor entre 40 y 60 en cualquier momento dado se considera "neutral", y tiene una noción similar a las "líneas de base" que se establecen en las técnicas de medición de EEG convencionales (aunque el método para determinar una línea de base de ThinkGear es patentado y puede diferir del EEG convencional). Se considera como "ligeramente elevado" al valor de 60 a 80, estos se los puede determinar cómo niveles que posiblemente están más altos de los normales (tanto para los niveles de Atención o Meditación que pueden llegar a ser más altos de lo normal para una persona determinada). Los valores de 80 a 100 se consideran "elevados", lo que significa que son fuertemente indicativos de niveles elevados de ese eSense.

De la misma manera, para el extremo opuesto de las escalas de denomina que los valores que varíen entre 20 y 40 muestran niveles "reducidos" de eSense, como también la variación de los valores que se encuentren en el rango de 1 y 20 muestra niveles "muy bajos" de eSense.

Dependiendo los niveles se puede identificar cada estado en el que se encuentra la persona como lo son la agitación, distracción o si tiene alguna anormalidad.

Por tal motivo el amplio rango con el cual se realiza la interpretación es porque algunas partes algoritmo eSense aprenden dinámicamente y, a veces, emplean algunos algoritmos de "adaptación lenta" para ajustarse a las fluctuaciones y tendencias naturales de cada usuario,

teniendo en cuenta y compensando el hecho de que el EEG en el cerebro humano está sujeto a rangos normales de variación y fluctuación. Por eso es una de las principales razones que estos sensores ThinkGear operan en gran variedad de condiciones tanto de ambiente como personales, al mismo tiempo que brindan una buena precisión y confiabilidad. Se alienta a los desarrolladores a interpretar y adaptar estos rangos de pautas para ajustarlos a su aplicación (por ejemplo, una aplicación podría ignorar valores por debajo de 60 y solo reaccionar a valores entre 60 y 100. (Neurosky.com, 2014)

2.3.3.2 Emotiv Insight

Es otra marca reconocida la cual brinda el servicio de venta de diferentes diademas que permiten la captación de la actividad cerebral como se observa en la Figura 4, para este caso es más robusta de acuerdo a su implementación que cuenta con 5 canales EEG que permite la detección de todo el cerebro además tiene un tiempo de configuración de 1 a 2 minutos es fácil de manejar y su uso de igual manera es sencillo, otro punto importante el cual es su conexión sencilla de forma inalámbrica la cual puede conectarse sin problema con dispositivos móviles como también con computadores; tiene una batería de larga duración un aproximado de 20 horas.

Figura 4

Emotiv Insight



Nota. De todos los dispositivos EEG Insight es el único que mide la actividad de todos los lóbulos corticales del cerebro mismo que permite obtener más información para realizar diversas investigaciones, Adaptada de *EMOTIV INSIGHT* [Fotografía], Emotiv, 2016, www.emotiv.com

2.3.3.3 Muse

Muse es una marca reconocida que realiza la venta de las diferentes diademas que permiten la captación de ondas cerebrales en este caso en particular diferentes acciones como lo es la meditación, estrés entre otras más. Se dará enfoque a la musa 2 el cual es un dispositivo actualizado para esta marca el mismo permite obtener los datos de cuando la persona se encuentra en diversos estados y a su vez con diferentes aplicaciones el poder entrenar a la mente en el cual se retroalimenta el aprendizaje en las ondas EEG.

En la Figura 5, se observa al dispositivo muse 2 que en particular un dispositivo en el cual se entra a la mente para poder obtener el estado de meditación de la persona el mismo trata de un estilo muy cómodo para el usuario siendo no invasivo teniendo sus sensores principales que irán ubicados en la parte de la frente del usuario.

Figura 5

Diadema Muse 2



Fuente: (Muse, 2018)

2.3.3.4 IFocusBand

El sensor iFocusBand es un sensor el cual es pensado para un uso deportivo por su estructura al ser de fácil uso y portátil mismo que cuenta con 3 sensores que son los utilizados para el entrenamiento cerebral, con gran ventaja que dichos sensores son resistentes a la transpiración que puede tener el usuario al usarlo en diferentes ambientes; los datos que se miden con el uso de este dispositivo se muestran mediante un teléfono móvil el cual debe tener la aplicación compatible para que los datos obtenidos lleguen al algoritmo IFocusBand mismo que es el que se encarga de realizar el proceso de las señales eléctricas obtenidas del cerebro como también la señales audiovisuales estas que se representaran con la actividad cerebral mediante el uso de neurofeedback el estilo de este dispositivo se lo puede ver implementado como una gorra como se muestra en la **Figura 6**. (TK, 2014)

Figura 6

iFocusBand



Nota. iFocus es un sistema portátil que contiene tres sensores EEG mismos que pueden conectarse para el entrenamiento y la retroalimentación cerebral, Adaptada de *Nueve proyectos de salud digital* [Fotografía], mobihealthnews, 2014, www.mobihealthnews.com

2.3.3.5 OpenBCI

Los dispositivos interfaz cerebro-computadora o también denominados (BCI) son de código abierto. Los mismos que cuentan en su placa con un convertidor de señales o datos analógicos a digitales de una manera rápida y asequible esta información se la utiliza para realizar un muestreo de la actividad eléctrica cerebral (EEG), como también la actividad muscular (EMG) y a su vez la frecuencia cardiaca (ECG) entre otras aplicabilidades que ofrecen estos dispositivos; además son compatibles con cualquier tipo de electrodos debido a que estos se encuentran respaldados por un marco de código abierto con aplicaciones que permiten el procesamientos de dichas señales.

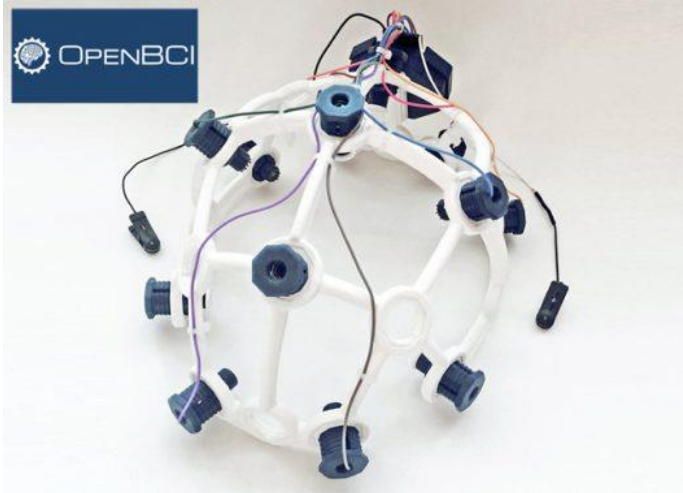
En opinión de los desarrolladores, los mayores desafíos que se enfrentan para comprender qué nos hace quienes somos no pueden ser resueltos por una empresa, una institución o incluso un campo completo de la ciencia. Creen que estos descubrimientos solo se realizarán a través de un foro abierto de conocimiento compartido y esfuerzo concertado, por personas de una variedad de disciplinas. Su visión es darse cuenta del potencial del movimiento de código abierto para acelerar la innovación en la ciencia del cerebro.

OpenBCI es una comunidad multidisciplinaria de investigadores, ingenieros, artistas, científicos, diseñadores y creadores, entre otros. Comparten la pasión por aprovechar las señales eléctricas del cerebro y el cuerpo humanos para comprender y expandir aún más la mente humana. A medida que la comunidad sigue creciendo, también lo hacen las posibilidades de lo que pueden descubrir y crear juntos. (Opensourceimaging, s.f.)

En la **Figura 7**, se observa los diferentes dispositivos que ofrece OpenBCI los cuales son utilizados en diversas actividades de investigación.

Figura 7

Dispositivo OpenBCI



Nota. La placa OpenBCI es un dispositivo robusto ya que permite captar diversas señales biológicas como la actividad cerebral (EEG), muscular (EMG) y cardíaca (EKG), Adaptada de *Acquiring biological signals (such as brain waves) Headsets and boards "OpenBCI products"* [Fotografía], tegakari, 2016, www.tegakari.net

2.4 El sueño

Es un estado alterado de la conciencia el cual es sumamente necesario para el bienestar del organismo. La supresión del sueño en seres humanos por periodos prolongados los altera desde el punto de vista emocional. (Afifi & Bergman, 2020)

2.4.1 Sueño de ondas lentas (no MOR)

Este tipo de sueño compone gran parte del sueño alcanzado un 75% aproximadamente en personas adultas, el mismo se caracteriza por tener sintomatologías físicas, alteraciones en conducta y electroencefalográficas (EEG). (Afifi & Bergman, 2020)

2.4.1.1 Etapas del sueño de ondas lentas (no MOR)

Etapa I: En esta etapa se denomina el apareamiento de la somnolencia la cual dura de uno a siete minutos. En la misma el individuo es fácil de despertar.

Etapa II: Para esta etapa se produce el sueño ligero a diferencia de la etapa anterior la persona requiere de estímulos más fuertes que la etapa I.

Etapa III: Para esta etapa se denomina la aparición del sueño moderadamente profundo, debido a que en esta etapa hay la se presentan las ondas EEG lentas que tienen un alto índice de voltaje.

Etapa IV: Para lograr el despertar esta etapa denominada sueño profundo se requiere estímulos muy potentes porque comprende a un 50% de la exploración electroencefalográfica de las ondas lentas EEG.

2.4.2 Sueño paradójico (MOR)

Esta etapa del sueño es en la que la persona sueña más. Cuando se encuentra en la etapa de sueño que consta de movimientos rápidos oculares denominado (sueño MOR), se hace más veloz la actividad del cerebro como también el aumento de velocidad en la respiración, la actividad cardiaca sube todo esto sucede mientras la persona permanece con los ojos cerrados. No es posible mover los músculos de los brazos ni de las piernas de manera temporal. Este tipo de sueño es fundamental porque se cree que ejerce una función demasiado fundamental en el aprendizaje y memoria. En el periodo en donde el sueño es normal el ser humano cruza por 4 o 5 ciclos diferentes de sueño los mismos que tienen una duración de unos 90 minutos cada uno e incluyen el sueño de movimientos oculares rápidos y este tipo de sueño en donde no se tiene los movimientos rápidos oculares la persona pasa de un sueño leve a un sueño profundo. También se

llama etapa MOR del sueño, fase REM del sueño, sueño de movimientos oculares rápidos y sueño REM. (Instituto Nacional del Cáncer, 2011)

2.5 Somnolencia

2.5.1 *Concepto de somnolencia*

La somnolencia es un estado de la persona en donde tiende a sentirse más somnoliento de lo normal en el transcurso del día. Las personas que sufren de este tipo de trastorno tienen a quedarse dormidas involuntariamente el cual es causante de problemas de seguridad dependiendo la actividad que se encuentren desarrollando. (Drowsiness, 2021)

2.5.2 *Causantes del estado de somnolencia*

El estado de somnolencia en las personas se da a diversas situaciones como pueden ser tanto de la salud de la persona como también a los aspectos de ambiente en el que se encuentre. (MedlinePlus, 2021)

2.5.2.1 El momento del día

Existen diversos horarios del día en donde se hace más pronunciada la aparición del estado de somnolencia como:

- La madrugada (2:00 am a 5:00 am) debido a que habitualmente el cuerpo se encuentra acostumbrado a descansar en este horario, el humano experimenta una cadencia del ritmo cardiaco por eso se presenta la probabilidad de que la persona pueda caer en somnolencia. (Unidad Nacional de Seguridad Vial, 2020)
- Las primeras horas de la tarde (13:00 pm a 15:00 pm) son momento en donde el sueño aparece con más facilidad, después del horario del almuerzo especialmente

si es una comida demasiado pesada, presenta una disminución de energía y presentar síntomas de sueño.

- Tarde-Noche (10:00 pm y medianoche) es un horario en donde también las personas se encuentran acostumbradas a dormir durante este horario por ende el cuerpo es probable que sufra de caer en estado de somnolencia.

2.5.2.2 Tipo de estimulación de ambiente

Uno de los factores que causan un posible estado de somnolencia en un conductor son los ambientes demasiados cálidos. Esto quiere decir que si la temperatura es más alta dentro del coche más probable la aparición de fatiga y somnolencia. (Tu-mejor-viaje.es, 2022)

2.5.2.3 Nivel de actividad del conductor

Si el entorno vial del conductor es muy monótono es probable que el conductor entre en estado de somnolencia debido a que este se relaja y los niveles de concentración bajan al contrario si el entorno vial del conductor es rico en estímulos favorece a que el conductor se encuentre despierto y concentrado. (Unidad Nacional de Seguridad Vial, 2020)

2.5.2.4 Carga de trabajo

El cansancio por exceso de trabajo genera un estrés que es de tipo especial, produciendo en la persona el agotamiento de energía física o también emocional. Las consecuencias de generar este tipo de estrés en la persona es la fatiga, el insomnio, vulnerabilidad a enfermarse entre otras por lo cual el exceso de trabajo más si son físicos produce consecuencias en la persona ya de manera inmediata o a corto plazo. (Mayo Clinic, 2020)

2.5.3 *Somnolencia en conductores*

La compleja tarea de conducir ocupa que el conductor preste toda su atención. Cuando el conductor no está completamente comprometido con esta tarea, muchos factores pueden reducir su capacidad para detectar y reaccionar ante situaciones de riesgo. La somnolencia es un estado peligroso que se ha demostrado que tiene un impacto significativo en la seguridad vial, lo que refleja el comportamiento inseguro de los conductores. (Soares et al., 2020)

El conductor cuando se encuentra en este estado tiende a sufrir de los siguientes efectos en el momento que conduce como lo son:

- Tiempo de reacción más alto
- Concentración disminuida propenso a distracciones.
- Toma de decisiones erráticas en escenarios complejos en donde se necesite respuestas rápidas.
- Alteraciones físicas en donde los músculos pasan a relajarse de una manera no adecuada por lo que hace que los movimientos sean mucho más retardados.
- Aparición de acciones o movimientos automatizados, realizar acciones por hábitos mas no por las necesidades del momento.
- Problemas de visión en donde es más complejo el enfocar la vista produciendo una visión borrosa y cansancio ocular.
- Micro sueños en donde el conductor tiende a quedarse dormido de manera ligera y este no percibe a lo que ocurre en la carretera en ese lapso.

2.6 Sistemas de detección de somnolencia

Los distintos avances en tecnología en conjunto con la industria automotora permiten llevar a un enfoque en el conductor y sus conductas, por este motivo se implementa en los últimos años al vehículo con el conductor (Human-Machine Interfaces) estas técnicas se llevan a cabo por la aplicabilidad de la programación de distinta información permitiendo lograr un asistente de conducción con el fin de mejorar la experiencia de manejo.

Actualmente en el mercado se puede encontrar gran cantidad de dispositivos electrónicos que permiten detectar factores fisiológicos del conductor como lo es el cansancio o la fatiga del mismo, recalcando que cada dispositivo se diferencia a su previo análisis por tanto los datos o características a tratar pueden ser diversas para establecer que el conductor se encuentra en buen estado para conducir.

El conductor suele establecerse en dos áreas en específico como lo es la detección del estado de distracción y también la detección del estado de somnolencia, estas dos áreas se encuentran correlacionadas ya que tienen conceptos bastante compartidos debido que el conductor puede presentar conductas que son asociadas tanto a la somnolencia y a la distracción respectivamente.

Por tal motivo los dispositivos de asistencia de conducción que actualmente hay se basan en la conducta del conductor mediante sistemas que evalúan la información que es emitida por el entorno que rodea al vehículo y también las medidas de conducción como los cambios de carriles constantes. Por otra parte, existen también los sistemas que son enfocados específicamente al monitoreo de actividad cerebral, rasgos faciales, monitoreo muscular del conductor al momento que este conduce.

2.6.1 Sistemas por parámetros

Los sistemas dedicados a la medición de parámetros al momento de conducir son orientados mediante la creación de perfiles específicos sobre la conducta o acciones que tiene el conductor al momento de conducir. Esta información se adquiere debido al trabajo y está relacionado con el nivel de actividad que tiene el conductor dentro del automóvil como lo son el uso de controles del vehículo y su influencia al conducir. El entorno de perfil que se genera en el automóvil es también monitoreado en donde se toma en cuenta diversos parámetros como lo es la variación de la velocidad, aceleración, que movimientos se realiza al volante, como se usa el acelerador y el freno entre otras acciones que se realizan por parte del conductor por otra parte también es analizado los factores externos al entorno del automóvil como lo es la medición del estado del semáforo, el viento y hasta incluso se llega a reconocer los límites de carril para saber en qué posición se encuentra el vehículo.

Estos sistemas llegan a ser muy útiles en los conductores, ya que si este se encuentra en un estado ya sea de fatiga o somnolencia los sensores incluidos en el vehículo interpretaran la información para realizar una corrección en la conducta del conductor. Ya que la información es almacenada y así se genera perfiles específicos en cada conductor y su conducta. Y los datos que son almacenados llegan a ser procesados y pasan a ser comparados con cada perfil de conducción permitiendo determinar si el conductor circula bajo un efecto de fatiga si llega a ser el caso el sistema advertirá al conductor por un sistema de alarma adecuada al automóvil. De mismo modo estos sistemas tienden a ser de costos muy altos imposibilitando que todo usuario llegue a adquirir este tipo de sistemas.

2.6.2 Sistemas por monitoreo cerebral

Estos sistemas permiten realizar el monitoreo exclusivo de la actividad cerebral del conductor utilizando sensores EEG, en este caso este proyecto se enfocará al análisis de parámetros que se obtienen por dispositivos específicos para señales eléctrica EEG, estos parámetros son utilizados para realizar la detección de la somnolencia o fatiga. Para la obtención de los datos se hace el uso de un sensor exclusivo EEG mismo que permite la monitorización de este tipo de actividad en el usuario, donde posteriormente son enviados los datos a una placa de procesamiento, la información será tratada y permitirá la determinación si el conductor se encuentra en un estado de vigila o somnoliento.

El uso del sensor EEG que sea ergonómico permite ser adaptable a cualquier ambiente de conducción y de la misma manera interfiere en la menor parte con las actividades realizadas por el conductor, lo que hace que este tipo de sistemas sean tan efectivos y precisos es que toman los datos en tiempo real y se realizan diversas comparaciones con las BDD creadas previamente para cada estado, donde el sistema pasará a tomar una decisión de predicción. Hay que aclarar que el uso de estas técnicas es aún nuevo y se encuentra en desarrollo por tanto los prototipos creados son sujetos a nuevas mejoras dependiendo también el sensor con el que se lleve la investigación.

2.6.3 Sistemas por reconocimiento facial

Este sistema permite realizar el monitorea de los rasgos faciales del conductor, y son directamente enfocados en rasgos fisiológicos mediante la utilización de técnicas de visión por computadora que permiten detectar sintomatologías de fatiga o somnolencia en el conductor.

Estos sistemas son basados en un análisis de diversas características emitidas por el conductor como el parpadeo lento, movimientos oculares más lentos entre otros. Mediante el uso

de cámaras que permiten capturar el rostro de la persona y procede a aplicar diversos algoritmos de procesamiento de imágenes permite identificar los patrones asociados con la somnolencia.

2.7 Tecnologías utilizadas para la detección de somnolencia

Actualmente existen diversas investigaciones en las cuales al aplicar diferentes técnicas permiten detectar el estado de somnolencia en los usuarios, a continuación, se muestra un apartado de proyectos involucrados en la detección del estado de somnolencia utilizando diversos métodos y tecnologías

2.7.1 *El DSM de SITRACK*

Como sus siglas DSM (Driver Status Monitor) o bien llamado sistema de detección de somnolencia y distracciones, es una herramienta robusta de la empresa SITRACK permitiendo ser una las diversas soluciones en seguridad de la conducción.

Este es un sistema único mismo que tiene 3 funcionalidades que son complementarias entre si permitiendo fortalecer a este sistema con el tercer ojo denominado (ADAS), la detección de la somnolencia (DSM) y una cámara opcional de video móvil (MDVR).

Teniendo en cuenta que el DSM también funciona de manera opcional con las otras dos funcionalidades como también de manera individual por tanto la implementación de este tipo de sistemas es una excelente opción que permite mejorar la conducción permitiendo una mejor seguridad vial. (Sitrack, 2021)

2.7.2 *Optalert*

Es una tecnología que es capaz de realizar una medición de forma eficaz y precisa del nivel de somnolencia del usuario en este caso el conductor, esta tecnología está enfocada en nuevos y avanzados sistemas de asistencia para el conductor. Realizando la cuantificación de la

somnolencia midiendo el abrir y cerrar de los ojos mediante el uso de algoritmos que son patentados por esta tecnología, por tanto, el algoritmo de Optalert permite la detección de un deterioro en el área cognitiva debido al estado de somnolencia lo que permite la activación de acciones que promuevan la recuperación del estado de vigila en el usuario. (Automotive - optalert)

2.7.3 *Detector de fatiga Bosch*

La empresa Volkswagen implemento el sistema de detección de fatiga de la marca Bosch, mismo sistema que se implementó en algunos de sus modelos como lo es el Volkswagen Passat entre otros a lo largo que estos sistemas se van actualizando. Este sistema utiliza diversos sensores y aplica algoritmos que son avanzados para permitir la monitorización y el análisis del comportamiento del conductor basado en el reconocimiento de patrones permitiendo una advertencia temprana de la fatiga y somnolencia. ({canalMOTOR}, 2022)

Esta es una tecnología innovadora que permite prevenir los accidentes en la carretera casados por estados fisiológicos del conductor proporcionando una solución bastante efectiva que permite mejorar la seguridad vial.

2.7.4 *Vigo Technologies*

Tecnología dedicada a realizar dispositivos que permiten el reducir los incidentes por fatiga en el conductor mediante la implementación de auriculares que realizan la supervisión del rendimiento y comportamiento de conducción con el panel de control en la nube de Vigo. Además del monitoreo en estado real del conductor permite medir los hábitos de este como lo es espejos laterales, la toma de descansos entre otros permitiendo realizar notificaciones si no se tiene un buen comportamiento por parte del conductor. (Vigo, s.f.)

2.8 Software Libre

Software que permite a los usuarios mirar libremente el código de programación, modificarlo, distribuirlo y usarlo sin ninguna restricción.

2.8.1 Java

Es un lenguaje de programación, que en 1995 fue su primera versión lanzada por la empresa Sun Microsystems. Con el pasar de los años ha evolucionado a sus inicios humildes hasta lograr potenciar en su gran mayoría el mundo digital que actualmente disfrutamos, debido a que proporciona una plataforma la misma que brinda la confiabilidad sobre la cual se realizan diversos trabajos de desarrollo de servicios y aplicaciones. Los nuevos productos como también los servicios innovadores digitales que son diseñados para el futuro cercano también en su mayoría siguen trabajando en Java.

El lenguaje de Java es orientado a objetos, lo que significa que cada uno de los programas realizados en este lenguaje son hechos en entidades las que representan cosas físicas mismas que son las denominadas como “objetos”. Actualmente los programas desarrollados en Java se los puede encontrar en computadores, servidores, dispositivos móviles e incluso tarjetas inteligentes mismas que ya son parte del vivir cotidiano actual de los seres humanos. (Java, 2011)

2.8.1.1.1 Donde se usa Java

El lenguaje de programación Java es muy reconocido a nivel mundial y por tanto es uno por el cual los desarrolladores eligen para realizar los diversos programas. Java solo necesita que se instale Java Run time Environment (JRE) para funcionar, independientemente de si está instalado en una PC de escritorio con Windows, Linux o Unix, una computadora Macintosh, un

teléfono inteligente o una computadora central. A continuación, se detallará algunos de los entornos en donde es más utilizado Java.

- **Aplicaciones de Android:** la mayoría de las aplicaciones de Android utilizan la API de Java o están escritas en Java, tanto que a menudo se considera que Java es el lenguaje de programación oficial para el desarrollo de aplicaciones móviles.
- **Aplicaciones GUI de escritorio:** muchas aplicaciones de escritorio se desarrollan en Java. Swing, Abstract Windowing Toolkit (AWT) y JavaFX son las principales herramientas utilizadas para facilitar el desarrollo de GUI.
- **Aplicaciones basadas en web:** Java se emplea a menudo para desarrollar una amplia gama de sitios web interactivos y aplicaciones basadas en web que se encuentran en los sectores de seguros, seguridad social, educación y salud.
- **Servicios financieros y minoristas:** Java se utiliza para escribir aplicaciones de gestión de transacciones y facturación, así como aplicaciones del lado del servidor.
- **Ciencia e investigación:** Java es el lenguaje favorito de la comunidad científica para una amplia gama de cálculos matemáticos y otras operaciones científicas. Puede manejar grandes conjuntos de datos y tecnologías de big data, ya que se usa para MATLAB y el marco Hadoop MapReduce. (Java, 2011)

2.8.2 *Python*

Es un lenguaje de programación el cual es usualmente utilizado en un amplio entorno tanto de aplicaciones web, el desarrollo de software como también en machine learning y todo lo que trata con la ciencia de los datos. En su gran mayoría los desarrolladores eligen la utilización de Python debido a que este es fácil de aprender como también sumamente eficiente y como

también es importante que el software se puede ejecutar en diversas plataformas volviéndolo sumamente útil. La descargar del software Python es de manera gratuita lo que atrae de mayor medida a los desarrolladores, como también el que se integra de manera eficaz a todos los tipos de sistemas y permite que la velocidad de desarrollo sea más veloz. (Python, 2017)

2.8.2.1 Beneficios del uso de Python

- Para los desarrolladores es más fácil el comprender los programas que se han hecho con Python ya que su sintaxis es de fácil comprensión.
- Permite optimizar los códigos y ser más productivos a los desarrolladores porque permite escribir los programas con menos líneas de códigos a razón de otros lenguajes de programación.
- Cuenta con una extensa biblioteca que tiene códigos que se pueden volver a usar para cualquier tarea permitiendo a los desarrolladores el no empezar desde cero un programa.
- La compatibilidad de Python con otros lenguajes de programación conocidos permite un mejor desarrollo de programas.
- Tiene una comunidad con una gran cantidad de desarrolladores en todo el mundo con los cuales se puede contar para resolver inconvenientes en la ejecución de proyectos permitiendo un gran soporte a diversas interrogantes.
- Existe gran información disponible en internet si se desea empezar de cero con este lenguaje de programación.
- A Python se lo puede utilizar en diversas plataformas de computadora como Mac, Linux o la más conocida Windows. Usos de Python

- Realización de scripts de manera automática. Los desarrolladores utilizan a menudo los scripts de Python para automatizar procesos.
- Realizar los procesos de machine learning. Los desarrolladores utilizan Python para ejecutar la ciencia de los datos como lo es la corrección y limpieza de datos, extraer características de los datos.
- Entrenar modelos de machine learning y crear algoritmos que permiten clasificar diversos datos de manera precisa.

2.9 Hardware Libre

2.9.1 *Arduino*

La plataforma de desarrollo Arduino es basada en una placa electrónica la cual es de hardware libre con su microcontrolador que es reprogramable además cuenta con una serie de puertos hembra mismos que permiten el establecer conexiones entre distintos dispositivos electrónicos como sensores y actuadores con el microcontrolador de manera simple. (Aguayo, 2019)

2.9.1.1 Tipos de Arduino

Existen algunos modelos de Arduino, con los cuales se puede realizar los diferentes proyectos dependiendo tanto en las necesidades del desarrollador, los mismos que varían en su precio el cual va relacionado con la robustez y especificaciones técnicas de cada dispositivo los módulos pueden ser desde el más simple hasta el más robusto.

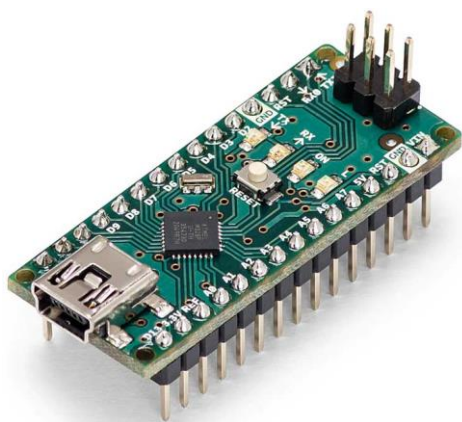
2.9.1.1.1 *Arduino nano*

Arduino Nano es una versión pequeña de Arduino UNO como se muestra en la **Figura 8**, misma que es compatible con placas de pruebas. Tiene más o menos funcionalidad del Arduino

UNO, pero en un factor de forma pequeño. Las únicas diferencias importantes con UNO son la falta de un conector de alimentación de CC, el uso de un puerto Mini USB en lugar de un puerto USB B y el chip convertidor USB-TTL. Nano usa un FT232, un chip de puente USB-UART dedicado de FTDI en lugar de un ATmega16U2. También es una opción muy popular entre los desarrolladores como UNO debido a su pequeño tamaño y precio económico. (Jobit, 2022)

Figura 8

Arduino nano



Nota. Dispositivo Arduino nano utilizado para diversos proyectos electrónicos, Adaptada de *Types of Arduino Boards* [Fotografía], Circuitdigest, 2020, www.circuitdigest.com

2.9.1.1.2 *Arduino uno R3*

Arduino Uno como se muestra en la **Figura 9**, la placa de desarrollo más popular y ampliamente utilizada. Está alimentado por un microcontrolador ATmega328P. Es la opción más popular entre la comunidad porque es económica, fácil de aprender y usar, y también hay una variedad de módulos prefabricados disponibles para esto, lo que facilita el desarrollo de nuevos proyectos o prototipos. Consta de 14 E/S digitales, de las cuales 6 pines son pines PWM

de 8 bits, 6 pines son entradas analógicas de 10 bits y puertos de comunicación básicos como SPI, I2C y UART. (Aguayo, 2019)

Figura 9

Arduino uno R3



Nota. Arduino uno es de las placas más utilizadas para la realización de proyectos electrónicos debido a su fácil entorno de aprendizaje, Adaptada de Arduino UNO [Fotografía], Arduino.cl, 2019, <https://arduino.cl/arduino-uno/>

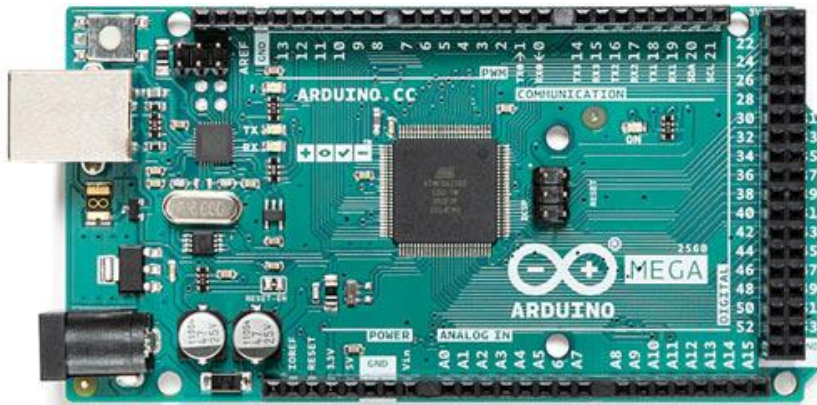
2.9.1.1.3 *Arduino mega*

Arduino Mega 2560 es la más grande de todas las placas. Está diseñado para aplicaciones en las que se necesitan muchas E/S o periféricos. Está alimentado por un procesador más grande y más capaz, el ATmega 2560. Esta placa tiene la mayor cantidad de E/S que la mayoría de las demás placas, 54 pines de E/S (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas y 4 UART. Tiene más almacenamiento flash y SRAM que la mayoría de las otras placas Arduino básicas. Es más popular entre la comunidad de impresoras 3D y CNC de código abierto, así como la comunidad de PLC de código abierto. (Jobit, 2022)

En la Figura 10, se denota que el módulo mega es más completo que los otros dispositivos que ofrece Arduino por lo cual con este dispositivo se permite realizar proyectos más robustos por sus especificaciones técnicas.

Figura 10

Arduino Mega



Nota. Dispositivo Arduino mega es uno de los más robustos de la gama Arduino debido a su mayor número de pines como también mejor capacidad de procesamiento muy utilizado para proyectos de electrónica avanzada, Adaptada de *Types of Arduino Boards* [Fotografía], Circuitdiges, 2020, www.circuitdigest.com

2.9.2 *Raspberry pi*

El Raspberry Pi se lo denominó con ese nombre debido a una serie de computadoras las cuales son de placa única mismas que son fabricadas por la fundación Raspberry pi, la cual es una organización benéfica situada en el Reino Unido misma que tiene como objetivo el educar a las personas en el área tecnológica y facilitar el acceso a la educación digital.

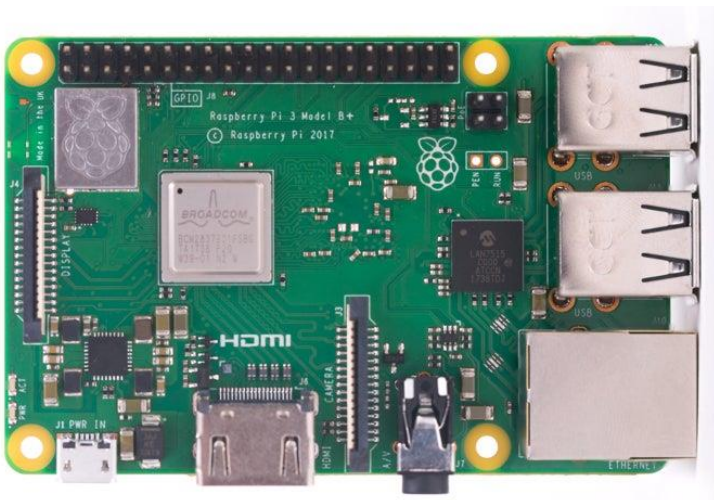
En el 2012 se lanzó la primera versión de la Raspberry pi y desde entonces hasta la actualidad son ya varias versiones cada vez más robustas de este dispositivo. El Pi original tenía

una CPU de 700 MHz de un solo núcleo y solo 256 MB de RAM, y el último modelo tiene una CPU de cuatro núcleos con más de 1,5 GHz y 4 GB de RAM. (What is a raspberry pi?, 2019)

En la **Figura 11**, se puede observar cual es la distribución y diseño de un Raspberry pi en su forma más general la cual cuenta con puertos USB como hdmi los cuales pueden ser conectados para sus diversos usos.

Figura 11

Raspberry pi



Nota. Dispositivo Raspberry pi utilizado para múltiples aplicaciones tanto de electrónica avanzada entre otros ya que es un minicomputador que brinda gran aplicabilidad de uso, Adaptada de What is a Raspberry Pi [Fotografía], OpenSource, 2019, www.opensource.com

2.9.3 UDOO

UDOO es una familia de mini PC de código abierto con tecnología Arduino compatible con Windows, Android y cualquier distribución de Linux; es una poderosa placa de creación de prototipos para el desarrollo y diseño de software, es fácil de usar y con unos pocos pasos puedes comenzar a usarla y crear tus proyectos con conocimientos mínimos. Fusiona diferentes mundos

informáticos en uno; cada mundo tiene sus fortalezas y debilidades, y todas ellas son útiles hoy en día en la educación y creación rápida de prototipos.

En sus características que más llama la atención es que UDOO es una computadora de hardware abierto y de bajo costo equipada con un procesador ARM i.MX6 Freescale para Android y Linux, junto con ARM SAM3X de Arduino DUE su tamaño es de 4,33 pulgadas x 3,35 pulgadas (11 cm x 8,5 cm) y tiene un bajo consumo de energía lo que lo vuelve ideal para realizar trabajo de manera portátil y ocupando el menor espacio posible teniendo una gran capacidad en su funcionamiento. (UDOO, 2021)

A continuación, en la **Figura 12** se puede ver cómo es la estructura de la placa UDOO, permitiendo evidenciar que esta tiene un tamaño pequeño por lo cual permite el llevar esta placa a cualquier lugar y poder realizar diversos proyectos sin ningún inconveniente.

Figura 12

Placa UDOO



Nota. Dispositivo UDOO es de la familia de mini pc con grandes características volviéndolo un dispositivo ideal para la ejecución de proyectos que requieran de procesamiento computacional

alto, Adaptada de *UDOO: Android Linux in a tiny single-board computer* [Fotografía], Kickstarter, 2021, www.kickstarter.com.

2.10 Estándar IEEE 802.15.1

El estándar IEE 802.15.1 denominado bluetooth específico de redes inalámbricas de área personal el cual permite enviar tanto datos y voz de distintos equipos mediante una conexión de radiofrecuencia de manera segura y libre a nivel global con su frecuencia de (2,4 GHz). A continuación, se enlistan los principales objetivos al utilizar este estándar.

- Simplificar las conexiones entre dispositivos móviles con los fijos
- Suprimir todo lo referente al uso de cables entre dispositivos
- Permitir crear redes de tamaño pequeño de forma inalámbrica con una sincronización fácil de envío de datos entre dispositivos conectados a esta red.

Los equipos que usan más a menudo esta tecnología son los de telecomunicaciones tales como PDAs, dispositivos móviles, computadoras portátiles, etc. (“Telecomunicaciones y Redes - INFORMÁTICA”)

"La tecnología bluetooth se encuentra diseñada exclusivamente para los dispositivos que son de bajo consumo ya que son de corto alcance de comunicación y su elaboración cuenta con transceptores de bajo costo." (“Estándar IEEE 802.15.1 (Bluetooth)”)

2.10.1 Arquitectura

La arquitectura de Bluetooth da servicios que ayuda a la interconexión de múltiples dispositivos dando así la posibilidad de la transferencia de datos entre dispositivos. Por tanto, el sistema principal consta de partes básicas, que son un transceptor que permite el envío y recepción de señales, para el establecer el enlace físico mediante un sistema de banda base.

2.10.2 Capas del protocolo Bluetooth.

"Bluetooth utiliza un sistema de capas dentro de la pila y cada una de ellas tiene su propio sistema de protocolos." ("Estándar IEEE 802.15.1 (Bluetooth)"). La finalidad de que el sistema se encuentre dividido en capas es acceder a la interconexión ya sea en aplicaciones o a su vez en dispositivos que sean elaboradas con especificaciones de este protocolo.

Bluetooth está dividido en capas de operación que son las siguientes:

- "Capa core o del sistema central se encarga de las funciones principales para los dispositivos que se encuentran en una piconet." ("Estándar IEEE 802.15.1 (Bluetooth)")
- Capa de reemplazo de cables que permiten la creación de puertos seriales virtuales.
- La capa que permite el control telefónico es desarrollada para permitir la realización de llamadas ya sean de voz o datos y también permite el control de dispositivos móviles.
- Capa de protocolos adoptados permita la flexibilidad de poder incorporarse de mejor manera con diversas tecnologías que son ya existentes en la actualidad.

2.11 Métodos de procesamiento de Datos

2.11.1 Proceso KDD

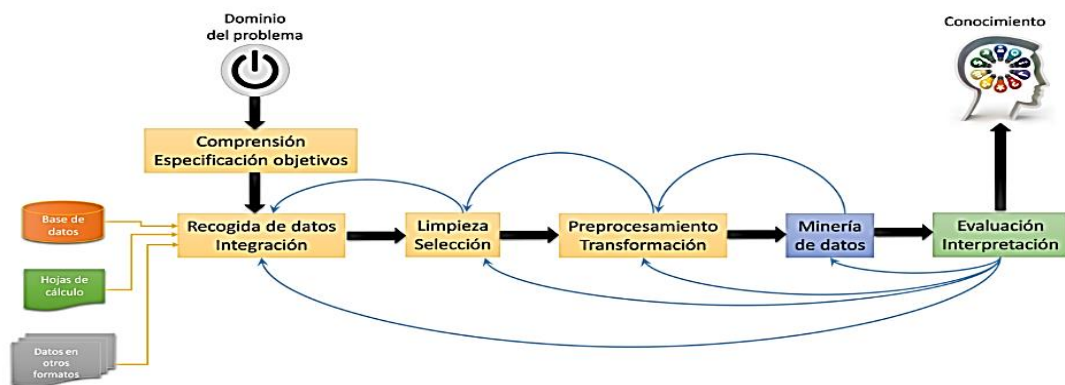
El proceso KDD se le desarrolla de manera secuencial con etapas a seguir con el fin de llevar la adquisición de manera automática de conocimiento el cual se inicial de una gran cantidad de datos sin procesar o de naturaleza iterativa, por lo cual se puede repetir este proceso las veces necesarias hasta lograr obtener la información requerida.

Habitualmente este proceso tiene como principal característica la identificación de la información para poder resolver problemas o necesidades que surgen a menudo. El conocimiento que se busca extraer en este proceso es de naturaleza desconocido para que sea aplicable o útil.

Por eso es indispensable el cumplir con este proceso que consta de diversas etapas a desarrollar de manera secuencial para obtener un resultado óptimo como se muestra en la **Figura 13** .

Figura 13

Proceso KDD



Nota. El proceso KDD tiene diversas etapas que se deben desenvolver, Adaptada de *Cómo es el proceso de extraer conocimiento a partir de bases de datos* [Fotografía], campusmvp, 2020, www.campusmvp.es

2.11.2 Fases del proceso KDD

Recopilación de Datos. La primera fase trata de la recopilación de datos de diferentes fuentes integrándose en un único reposito. Esta información obtenida se debe almacenar de manera segura y confiable. Las subfases que se detallan a continuación son sumamente importantes para el desarrollo de este proceso del tratamiento de los datos.

- **Datos seleccionados:** Con el uso de diversos métodos de filtrado de los datos se procede a realizar la eliminación de datos que son irrelevantes para el análisis buscando optimizar los resultados.

- **Limpiar datos:** Esta etapa es sumamente importante en el proceso ya que al tener almacenado los datos de grandes tamaños se procede a eliminar todos los valores que son innecesarios o inexistentes para así obtener un conjunto de datos de menor tamaño, pero más optimizado con valores reales.
- **Datos transformados:** Para esta etapa se realiza el preprocesamiento de los datos preparándolos para las últimas etapas de la minería de los datos en el cual se analiza los datos obtenidos mediante algunos algoritmos dependiendo la aplicabilidad que se le da a los datos.
- **Minería de datos:** La parte más importante de KDD ya que aquí se realiza el proceso respectivo en el cual se obtendrán los modelos ya sea de clasificación o regresión para los datos y estos sean interpretados. (“Breve explicación del proceso KDD - Laboratorio de Certificación”)
- **Interpretación y evaluación de modelos:** Una vez que se obtiene el mejor modelo en la etapa anterior esta es evaluada con su nivel de exactitud como también en la aplicabilidad requerida.

2.12 Algoritmos de clasificación supervisada:

El algoritmo de clasificación se denomina como una metodología la cual permite la reorganización de datos en gran cantidad dando un orden a estos, ya puede ser numéricos, valores mayores como valores menores. Existen algoritmos que se encargan en la clasificación estos trabajan tomando listas como un conjunto de datos de entrada, se procede a realizar las operaciones matemáticas respectivas para entregar una o diversas matrices que son ordenadas como salida.

2.12.1 Algoritmo KNN

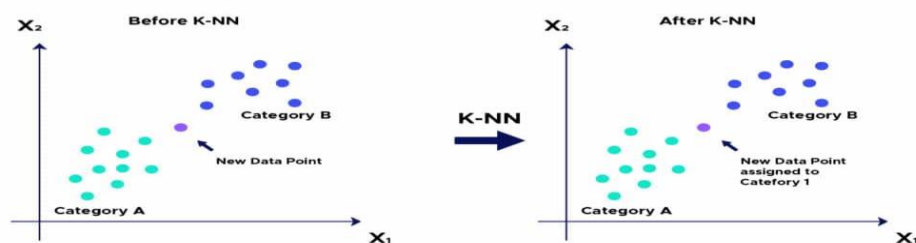
KNN denominado de vecinos más cercanos, es un algoritmo que se encarga en la clasificación mediante aprendizaje supervisado que tiende a ser no paramétrico, utilizando la cercanía para realizar sus predicciones sobre la conjunción de un punto de datos individual. Este algoritmo es útil tanto para problemas de clasificación como de regresión lineal, es más utilizado como clasificador ya que puede encontrar puntos parecidos cerca uno del otro.

El proceso para realizar diversos problemas de clasificación se debe asignar una etiqueta de la clase la cual representara a un conjunto de datos determinado.

En la **Figura 14**, se puede observar cómo es el comportamiento para el tratamiento de la información mediante la aplicación del algoritmo de vecinos más cercanos denominado KNN, en donde realiza la clasificación respectiva mediante a la información dada mediante dos etiquetas.

Figura 14

Algoritmo KNN



Nota. KNN es un algoritmo supervisado de Machine Learning y permite realizar proyectos tanto de clasificación como de regresión, Adaptada de ¿Qué es el algoritmo KNN? [Fotografía], DataScientest, 2021, www.DataScientest.com

2.12.1.1 Cálculo de métricas de distancia KNN

Los cálculos matemáticos son la principal función en este algoritmo ya que sirve para identificar la cantidad de vecinos más cercanos de un punto que se encuentra analizando, para así poder dar una etiqueta a este punto. Para lograr realizar esto, KNN necesita cumplir con algunos requisitos.

- Calcular las métricas de distancia
- Para identificar los puntos de datos que se encuentran más cerca del punto de análisis se necesita calcular las distancias que se tiene entre el punto de consulta y los otros puntos de datos. Las métricas de la distancia permiten ayudar a tener formados los límites de decisión, los cuales se dividen en los puntos de consulta en diferentes lugares.
 - Si la distancia euclidiana (p=2): Esta es una de las medidas que se utiliza más ya que se encuentra limitada a vectores que tienen un valor real. Usando la Ecuación [1], la cual mide una línea recta entre el punto de consulta y el otro punto que se mide.

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2} \quad \text{Ecuación [1]}$$

2.12.2 *Random Forest*

Es uno de los algoritmos de mayor popularidad y usualmente más usados por los analistas de datos. Debido a que este es un algoritmo de aprendizaje automático que es supervisado que es muy usado para resolver problemas tanto de regresión como de clasificación. (Sruthi, 2021)

2.12.2.1 Funcionamiento de Random Forest

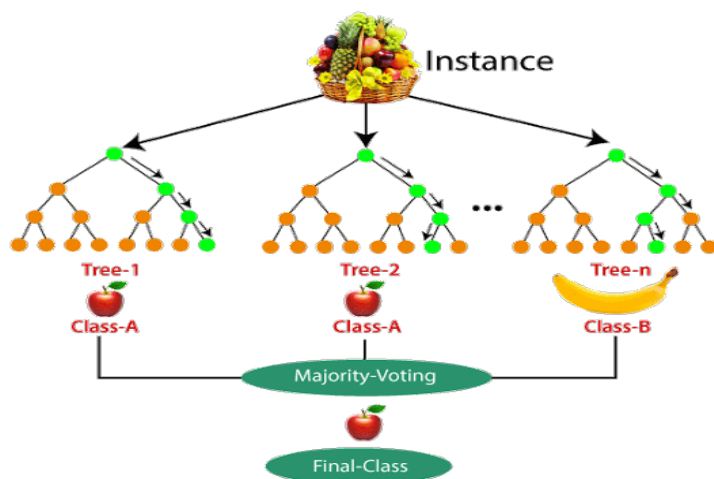
Para comprender el funcionamiento de este algoritmo hay que considerar que se basa en un método de ensemble mismo que cuenta con dos tipos de métodos para su funcionamiento.

- Embolsado: Cuando funciona por este método se crea un subconjunto de datos de entrenamiento que es diferente y a partir de estos datos el resultado se denomina en la votación por mayoría Random Forest funciona de esa manera.
- Impulso: Método por el cual combina lo que se denomina aprendices que son débiles a estos se los transforma en fuertes, creando modelos que funcionan de manera secuencial aspirando que el modelo final sea el más preciso. (Sruthi, 2021)

2.12.2.2 Ciclo de funcionamiento del algoritmo Random Forest

- Se escoge un subconjunto de datos y también un subconjunto de características con los cuales se elabora cada árbol de decisión.
- Los árboles de decisión son individuales para cada muestra.
- Se genera una única salida individual para cada árbol de decisión
- La solución del algoritmo es basada a la votación de manera mayoritaria o su promedio tanto para regresión como para clasificación.

En la Figura 15, se muestra el proceso del algoritmo para realizar la decisión final de predicción esto es basado al voto mayoritario.

Figura 15*Modelo Random Forest*

Nota. El funcionamiento del algoritmo Random Forest es complejo para la detección del resultado final, pero de igual manera sumamente preciso, Adaptada de Understan Random Forest Algorithms [Fotografía], Analytics Vidhya, 2023, www.analyticsvidhya.com

2.12.2.3 Hiperparámetros más importantes en Random Forest

Existen diversos Hiperparámetros que son utilizados por este algoritmo con el fin de mejorar su rendimiento y así mejorar el poder predictivo de estos haciéndolos más rápidos.

A continuación, se detallan los hiperparámetros esenciales que ayudan a mejorar el poder predictivo del algoritmo.

- **n_estimators:** Es el número total de árboles que se construyen antes de que se realice el promedio de las predicciones.
- **Max_features:** Es la cantidad máxima de características que son consideradas para dividir un nodo.

- **Mini_sample_leaf:** Son el menor número de hojas que son necesarias para la división de un nodo.
- **Criterio:** Que se considera para denominar el criterio (entropía, impureza de Gini, pérdida de registro)
- **Max_leaf_nodes:** es la cantidad máxima de nodos hoja por cada árbol.

A continuación, se detallan los hiperparámetros esenciales que ayudan a mejorar la velocidad en el algoritmo.

- **N_jobs:** permite elegir cuantos procesadores se puede utilizar. Si el valor es 1 se estará utilizando únicamente 1 procesador, pero si el valor es -1 indica que no hay límite y se utilizaran los procesadores disponibles.
- **Random_state:** Permite controlar la aleatoriedad de las muestras tratadas. El modelo siempre dará los mismos resultados si los hiperparámetros no son modificados.
- **Obb_score:** OBB indica fuera de bolsa. Trata de un método de validación cruzada de bosque aleatorio. (Sruthi, 2021)

CAPITULO III: DESARROLLO EXPERIMENTAL

En el siguiente capítulo se realiza el análisis de la situación actual en la cual se realiza la descripción de manera general de cada etapa para la realización del diseño y a su vez la elaboración del sistema de monitorización y alerta se explicará tanto en funcionamiento como distribución de hardware y software que se utilizará en el transcurso de la realización del sistema utilizando la metodología iterativa.

Para la elección tanto de software y hardware se realizara mediante los diferentes procesos que permiten la elección de requerimientos en este caso para el software se realizara mediante la utilización de la técnica ISO/ IEC/IEEE 29148 que la cual permitirá la elección de software adecuado para la realización de este proyecto, mientras para la elección del hardware se utilizara mediante la técnica de Stakeholders el cual permite la elección de los dispositivos que se adapten a las necesidades de la realización del proyecto.

3.1 Metodología

3.1.1 Modelo iterativo

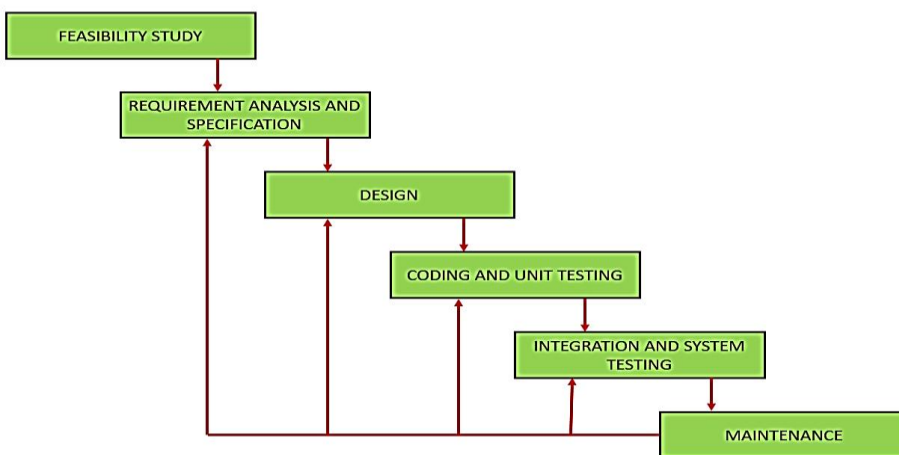
Para la realización de este proyecto se utilizará el “modelo iterativo” el cual es una variante al modelo comúnmente conocido y muy utilizado en proyectos denominado “modelo en cascada”. El modelo iterativo denota agregados algunos cambios al modelo antes mencionado, por lo tanto, se denominaría casi similar al modelo en cascada, pero este busca el aumentar la eficiencia para el desarrollo del proyecto en el software. Por lo cual el modelo iterativo permite la retroalimentación desde cada ciclo a sus antecesores permitiendo el mejoramiento del proyecto y denotando así una gran diferencia al modelo en cascada.

Este modelo es una variante al modelo en cascada en el cual se ven incorporados diferentes cambios al modelo antes mencionado, por lo tanto, es casi similar al modelo en cascada, pero en este se busca el aumento de la eficiencia para el desarrollo del software. Este modelo permite retroalimentarse desde cada ciclo a sus anteriores la cual conlleva a ser la principal diferencia con el modelo en cascada.

Como se muestra en la Figura 16, existe la retroalimentación de cada etapa a sus antecesores, por lo tanto, cuando sean detectados errores de alguna etapa siguiente gracias a las rutas de retroalimentación se permite corregirlos. Las rutas de retroalimentación permiten reelaborar la etapa se en la que se cometen errores y estos cambios se reflejan en las fases posteriores.

Figura 16

Modelo Iterativo



Nota. Modelo Iterativo, Adaptada de *8 Software Engineering- Iterative Waterfall Model*

[Fotografía], GeeksforGeeks, 2011, www.geeksforgeeks.org

3.2 Fase 1: Análisis y especificación de requerimientos

En esta fase se realizara un barrido para conocer de cada factor involucrado en el proyecto, mediante la implementación de un análisis investigativo, con el cual se permitirá el facilitar el acceso a la información necesaria sobre los problemas y necesidades a resolver en la investigación, seguido a esto, se denominara los requerimientos los cuales cumplan de manera satisfactoria las necesidades del prototipo, para lo cual se lo realizara mediante la investigación de los requerimientos de usuarios, de sistema, operacionales, de arquitectura, entre otros más que son necesarios para la elaboración del proyecto.

3.2.1 Situación actual

De acuerdo con la Organización Mundial de la Salud (OMS), el cansancio es uno de los factores principales para los siniestros en carretera, por otro lado, los reportes emitidos por la Agencia Nacional de Transito (ANT) registra que el 0,6% de los accidentes fatales son causados porque el conductor se encuentra en un estado somnoliento durante el 2021 y el 0.2% resultaron heridos. (Primicias, 2022)

Los conductores que se dedican al comercio, son propensos a sufrir somnolencia esto se debe a diversos factores que son asociados a sus actividades diarias, horarios irregulares, falta de descanso adecuado. Anund et al. (2008) argumenta que “el 13% de los conductores que realizan largos tramos son propensos a caer en estado de somnolencia”. Además, el estudio realizado por Philop et al. (2015) “Determina que la somnolencia en vehículos de comerciantes tiene una alta prevalencia con el 20% de los conductores son propensos a quedarse dormidos”. Por tanto, es necesario el realizar un análisis de manera exhaustivo de todos los parámetros que están ligados a la determinación de la somnolencia en el conductor, para detectar y prevenir posibles fatalidades en la carretera. Para el desarrollo del presente proyecto se establece algunos

requerimientos importantes para el correcto diseño del prototipo. Con el fin de determinar los requerimientos necesarios de los participantes en el desarrollo del sistema. Se establece requisitos que permiten un óptimo desarrollo del proyecto y la interacción del mismo con el usuario.

3.2.1.1 Técnica de investigación

Para la realización y obtención de información para este estudio se la lleva a cabo mediante la realización de una encuesta como se muestra en el Anexo A, ya que es una técnica descriptiva, que ayuda para la recopilación de datos sobre la somnolencia en conductores dedicados al comercio. La encuesta se encuentra diseñada para obtener información relevante y precisa, sobre experiencias y percepciones de la implementación de un prototipo de detección del estado de somnolencia.

La elección de esta técnica es basada por su capacidad en obtener datos de una muestra representativa de conductores, lo que permitirá tener datos más representativos y precisos que ayudaran de mejor manera al desarrollo del diseño del prototipo.

3.2.1.2 Tamaño de la muestra.

La utilización del tamaño de la muestra brinda una mayor confiabilidad y da validez de la encuesta. Por lo cual, se calcula el número de muestras que son necesarias para cumplir con la confiabilidad de validación de este proceso. A continuación, se muestra la ecuación utilizada para dicho cálculo.

$$n = \frac{N\sigma^2 Z^2}{(N - 1)e^2 + \sigma^2 Z^2}$$

Donde:

n =número de muestras

N = Tamaño de la población

Z = Nivel de confianza es una constante determinada (99%=2.58; 95%1.96). El valor de dicho valor es escogido por el encuestador dependiendo el enfoque al que se aplique.

e =error muestral (No debe ser mayor al 9%)

σ = Constante de desviación estándar de la población es de 0.5

$$n = \frac{20(0.5)^2(1.96)^2}{(20 - 1)0.09^2 + (0.5)^2(1.96)^2} = 13.35$$

El tamaño de la muestra para que sea validada es de 13 personas.

A continuación, de manera más detallada se muestra el método desarrollado para la recolección de información primordial misma que sirve para la determinación de los requerimientos importantes para el desarrollo del prototipo como se muestra en la Tabla 1.

Tabla 1

Análisis situación actual

Situación actual	
Método:	Para la realización de la tesis se propone la utilización del método de investigación descriptiva mediante esta se recolectará información actual mediante la implementación de una encuesta.
Formato:	La encuesta tiene como meta, el comprender porque factores por los cuales el conductor del automotor puede ser propenso a sufrir de estar en un estado de somnolencia ya sea por carga de trabajo o viajes largos buscando brindar una solución que ayude a este problema.

Por tal motivo la encuesta estuvo conformada por preguntas abiertas o cerradas. Las preguntas permitieron conocer si los conductores están al tanto sobre que es estado de somnolencia y a su vez si lo mismos permitirán el uso de un dispositivo que permita la detección de este problema al conducir. En el Anexo A del documento se puede visualizar el formato de la encuesta realizada.

Fuente: Elaborado por el autor

La encuesta se la realizo a futuros usuarios en este caso conductores, quienes son las personas que realizan esta actividad, por tanto, la información obtenida permite una ayuda significativa para el desarrollo del proyecto, teniendo los siguientes resultados.

Con las preguntas efectuadas permitieron comprender cual es la situación actual del conductor y sus actividades diarias. Ya que estas actividades están relacionadas al tipo de trabajo que el usuario realiza por tanto los horarios de conducción pueden ser más largos o más cortos como también los ambientes por la hora del día en la que estos realizan su trabajo siendo propensos a caer en somnolencia o cansancio dependiendo de los factores antes mencionados.

3.2.2 Definición de requerimientos

Mediante el análisis de la información obtenida de las distintas tecnologías y modelos de información las cuales tienen relación en la implementación para este proyecto, se determina tanto los elementos y requerimientos que son necesarios para cumplir con cada uno de los objetivos planteados, por lo cual se evaluará los requerimientos de sistema y requerimientos de arquitectura.

3.2.2.1 Construcción de atributos de requerimientos

Para realizar la construcción de los atributos se la realiza en base a tres requerimientos Stakeholders, sistema y arquitectura. Los mismos que tienen atributos propios por tal motivo es indispensable que cada uno de los requerimientos sea verificable y a su vez se cumplan.

3.2.2.2 Nomenclatura de requerimientos

Para cada requerimiento necesario para la ejecución del proyecto es importante el otorgarles una abreviatura con la cual se permita un mejor manejo de datos. En la *Tabla 2* que se muestra a continuación están las abreviaturas a utilizar.

Tabla 2

Nomenclatura de requerimientos

Requerimiento	Abreviatura
Stakeholders	STSR
Sistema	SYSR
Arquitectura	SRSR

Fuente: Elaborado por el autor

3.2.3 *Requerimientos de Stakeholders*

Los requerimientos de Stakeholders (STSR) tienen como objetivo la identificación de los requisitos para los interesados por el sistema. Particularmente se realiza un previo análisis de un conjunto de requerimientos los cuales son tanto operacionales como de usuario el cual tiene que ver directamente con la interacción de los involucrados con el sistema como se muestra en la *Tabla 3*.

Tabla 3*Lista de Stakeholders*

Lista de Stakeholders	Descripción
1. Usuarios del Sistema	Conductores.
2. MsC. Carlos Vásquez	Director trabajo de titulación.
3. MsC. Jaime Michilena	Asesor trabajo de titulación.
4. Edgar Ichau	Desarrollador del proyecto

Fuente: Elaborado por el autor.

Se denomina las necesidades y prioridades las cuales se encuentran clasificadas de acuerdo con la relevancia del requerimiento. En la Tabla 4, se muestra los niveles de prioridad especificados para el diseño de los requerimientos que en este caso son 3 prioridades.

Tabla 4*Niveles de prioridad de requerimientos*

Prioridad	Descripción
Alta	Sera de un nivel de alta importancia el cual debe ser incluido en el desarrollo del sistema.
Media	Se lo considera con un nivel medio de relevancia, pero él no incluirlo puede afectar al desarrollo y funcionamiento del sistema.
Baja	Se lo considera con un nivel prioridad mínimo, por lo tanto, se lo puede excluir porque no genera un efecto negativo y significativo al funcionamiento del sistema.

Fuente: Elaborado por el autor.

En la Tabla 5, se muestra los requerimientos operacionales y de usuarios mismos que conforman parte de los Stakeholders con los cuales se permite definir el correcto funcionamiento del sistema y prioridad a cada uno de estos. Para determinar dichos requerimientos que son indispensables se lo realizo en base a la encuesta (Anexo A).

Tabla 5

Requerimientos Operacionales y de usuarios.

STSR					
REQUERIMIENTOS DE STAKEHOLDERS					
#	REQUERIMIENTO	PRIORIDAD			RELACION
		ALTA	MEDIA	BAJA	
REQUERIMIENTOS OPERACIONALES					
STSR1	El sistema debe implementarse en un automóvil.	■			
STSR2	El sistema debe ser no invasivo para el conductor.	■			
STSR3	El sistema de captar las señales EEG		■		
STSR4	Se debe mostrar las mediciones de las frecuencias cerebrales del usuario.			■	
STSR5	El sistema debe tener bajo consumo de energía	■			
STSR6	La conexión inalámbrica debe estar en correcto funcionamiento para el envío de datos.		■		
STSR7	Disponer del dispositivo para anunciar estados del usuario	■			
STSR8	El prototipo debe ser lo más pequeño posible.		■		

STSR9	El dispositivo debe permanecer activo durante las horas de conducción		■
STSR10	El sistema debe ser de bajo costo de implementación	■	
STSR11	El sistema debe adaptarse a los diferentes ambientes para el conductor.		■
STSR12	Sistema debe tener la capacidad de envío de gran cantidad de datos.	■	
STSR13	El sistema debe ser resistente a la manipulación.	■	

REQUERIMIENTOS DE USUARIOS

STSR14	El usuario debe cambiar la batería del sensor frecuentemente.	■	
STSR15	El dispositivo cumplirá con su trabajo siempre y cuando el usuario tenga puesto correctamente el sensor.	■	
STSR16	Acceso y manipulación total del sistema para el usuario.		■
STSR17	El usuario no debe modificar ningún parámetro del sistema.	■	

Fuente: Elaborado por el autor.

3.2.4 *Requerimientos del sistema*

Para los requerimientos del sistema se realiza las limitaciones en el ámbito funcional del proyecto, en el cual se referencia a todos los procedimientos y cada propiedad que se debe dar, con estos requerimientos que debe tener el sistema se determina las funciones a ejecutar. Por lo cual con el apoyo del estándar ISO/IEC/IEEE 29148:2001, el cual brinda las pautas para la aplicación de los requisitos y procesos relacionados con los que se van a implementar principalmente para los sistemas y productos de software que brindan servicios a largo ciclo de vida, mismo que ayuda para definir cada una de las funciones requeridas y necesitadas en la ejecución del proyecto.





La Tabla 6 en la cual se poseen los requerimientos iniciales que son fundamentales en el sistema, requerimientos de Stakeholders y los requerimientos de arquitectura; como objetivo es mostrar de manera clara y concisa la información la misma que permitirá realizar la mejor elección tanto para software como para hardware para el desarrollo puntual del sistema.

Tabla 6



Requerimientos del Sistema

SYSR					
REQUERIMIENTOS DEL SISTEMA					
#	REQUERIMIENTO	PRIORIDAD			RELACION
		ALTA	MEDIA	BAJA	
REQUERIMIENTOS DE USO					
SYSR1	Conexión de fuente de 12 Vcc.	■			
SYSR2	Conversor de 12 Vcc a 5 Vcc para prototipo embebido.	■			
SYSR3	Colocación correcta del sensor neuronal.	■			
SYSR4	Acceso y manipulación total del sistema.			■	




REQUERIMIENTOS RENDIMIENTO

SYSR5	El sistema debe tener bajo consumo de energía.	
SYSR6	Información confiable de la lectura de las ondas EEG.	
SYR7	Procesamiento correcto de datos obtenidos por el sensor.	
SYR8	Respuesta instantánea de las acciones a realizar.	
SYR9	Sistema debe operar correctamente días continuos sin problemas.	

REQUERIMIENTOS DE INTERFACES

SYR10	Módulo de comunicación inalámbrico bluetooth entre sensor y sistema embebido	
SYR11	Interfaz serial de 57600 baudios entre hc-05 y nodo central	

REQUERIMIENTOS DE MODOS Y ESTADOS

SYR12	Modo on/off del prototipo embebido.	
SYR13	Modo on/off del sensor según su utilización.	
SYR14	La alerta debe activarse y desactivarse cuando el sistema detecte el estado de somnolencia o no	

REQUERIMIENTOS FISICOS

SYR15	El sistema debe estar cerca de la fuente de alimentación del automóvil.	■
SYR16	El tamaño del prototipo embebido debe ser pequeño para que no ocupe demasiado espacio al interior del automóvil y no entorpezca las actividades del conductor.	■

Fuente: Elaborado por el autor

3.2.5 *Requerimientos de arquitectura*

Los requerimientos de arquitectura (SRSH) son los que se van a incluir para cada requisito de diseño tanto para el software como para el hardware; esos serán los que después servirán para escoger los componentes, programas a utilizar para hardware y software que serán parte esencial del sistema para el prototipo. A continuación, en la Tabla 7 se detallan los requerimientos necesarios para la elaboración del sistema para su arquitectura.

Tabla 7

Requerimientos de software, hardware y eléctricos que determinan la arquitectura del sistema.

SRSH

REQUERIMIENTOS DE ARQUITECTURA

#	REQUERIMIENTO	PRIORIDAD	RELACION
---	---------------	-----------	----------

ALTA MEDIA BAJA

REQUERIMIENTOS LOGICOS

	El sistema debe ser compatible con una placa embebida		
SRSH1	Soporte y compatibilidad de la tecnología Bluetooth para la recepción y envío de datos entre el sensor neuronal y el microprocesador.	■	
SRSH2	Ubicación del prototipo respetando los 10m de rango establecido para su comunicación inalámbrica.	■	
SRSH3	Conexión cableada entre los dispositivos a interactuar		■

REQUERIMIENTOS DE DISEÑO

SRSH4	El sistema pueda adaptarse a diferentes condiciones climáticas.		■
SRSH5	Envío y recepción de datos de manera inalámbrica.	■	
SRSH6	La ejecución del sistema debe ser a diario con intervención del conductor.	■	
SRSH7	El sistema debe tomar datos de manera constante para realizar el tratamiento de los mismos.	■	
SRSH8	El relevador debe estar conectado con los dispositivos a interactuar.	■	
SRSH9	El sistema debe ser de fácil manipulación para su movilización sin temor a daños en el mismo.		■

REQUERIMIENTOS DE HARDWARE

SRSH10	Se requiere que los módulos de comunicación inalámbrica sean de bajo consumo de energía.			■
SRSH11	El sistema embebido debe tener alta capacidad de procesamiento.			■
SRSH12	El sistema embebido debe ser capaz de realizar el tratamiento de datos en tiempo real.	■		
SRSH13	Sensor neuronal con conexión bluetooth.	■		
SRSH14	Sensor neuronal compatible el sistema embebido.	■		
SRSH15	El sistema embebido debe tener varios puertos que permitan transmisión y recepción de datos.			■
SRSH16	El módulo bluetooth debe ser compatible con el sensor neuronal.	■		
SRSH17	El sistema embebido debe tener más de dos pines de salida para la conexión de dispositivos(alarma).	■		

REQUERIMIENTOS DE SOFTWARE

SRSH18	Lenguaje de programación de código abierto.	■		
SRSH19	Disponibilidad de librerías para la configuración de tratamiento de la información.	■		
SRSH20	Script de programación para la obtención de datos del sensor EEG.			■

SRSH21	Se requiere software compatible con el cual permita la ejecución de scripts desde un sistema embebido.	■
SRSH22	Disponibilidad de documentación para mejor comprensión del lenguaje de programación	■
REQUERIMIENTOS ELECTRICOS		
SRSH23	Se requiere un conversor eléctrico de 12 Vcc a 5 Vcc para el sistema embebido.	■
SRSH24	Se requiere una batería AAA para el funcionamiento del sensor neuronal.	■

Fuente: Elaborado por el autor

3.3 Elección de hardware y software

3.3.1 Elección de hardware

Una vez que se establece los requerimientos necesarios para el sistema se procede a la elección tanto de hardware como de software que es necesario, la elección se procede a realizarla mediante una tabla comparativa en la cual se analiza las especificaciones de cada dispositivo a elegir el cual debe cumplir con los requerimientos de arquitectura para ser elegidos; dichas especificaciones se las otorgara mediante un valor nominal el cual referenciará a detalle si cumple o no cumple con el requerimiento. En la Tabla 8 se detalla y establece la puntuación de “1” para si cumple y “0” para no cumple.

Tabla 8

Puntuación de requerimientos

Descripción	Valor
--------------------	--------------

Cumple	1
No cumple	0

Fuente: Elaborado por el autor

3.3.1.1 Bloque de captación de datos EEG:

Este bloque el cual es el encargado de la toma de los datos en la actividad cerebral para sus diferentes frecuencias en el usuario, estos datos serán tomados mediante la utilización de un sensor neuronal, por lo cual se realiza un análisis de especificaciones de distintos dispositivos que son adaptables a la realización del proyecto. Por tal motivo se observa la ponderación especificada en la *Tabla 8* para la elección del sensor dependiendo del cumplimiento de cada requerimiento de arquitectura el cual esta especificado en la sección 3.2.5, específicamente en la *Tabla 7*.

3.3.1.2 Selección de sensor neuronal.

De acuerdo con los requerimientos de arquitectura que se especificaron en la *Tabla 7* en el cual están las características específicas del sensor neuronal con el cual permita tener una correcta adquisición de datos y compatibilidad en la comunicación con el sistema embebido. Por tal motivo se muestra la *Tabla 9* en la cual se tiene distintos dispositivos que pueden ser elegibles para la ejecución del proyecto.

Tabla 9

Elección sensor neuronal

Hardware	Requerimientos				Puntuación total
	SRSH1	SRSH5	SRSH13	SRSH24	

Mindwave Neurosky	1	1	1	1	4
Mobile 2					
Muse 2	1	1	1	0	3
Brainlink lite	0	1	0	1	2
iFocusBand	1	1	0	0	2

Elección: Para la elección del sensor neuronal que se va a utilizar en la ejecución de este proyecto según la tabla de requerimientos de arquitectura planteados en la *Tabla 7*, se concluye que es ideal el uso del sensor neuronal Mindwave Neurosky Mobile 2 debido a que este cumple con los requerimientos establecidos para una correcta ejecución del proyecto.

Fuente: Elaborado por el autor

A continuación, en la *Tabla 10* se detalla las características técnicas fundamentales del sensor neuronal Mindwave Neurosky Mobile 2.

Tabla 10

Características Neurosky Mindwave Mobile2

Especificaciones	Propiedades
Modelo	MW003
Alimentación	1.5v – 95mA
Bluetooth	TGAM2.9B v4.0

Fuente: (SANDOROBOTICS, 2018)

3.3.1.3 Bloque de envío de datos.

Este bloque es el encargado de permitir la comunicación entre el sensor neuronal antes elegido y el microprocesador en donde se tratarán los datos obtenidos por el sensor. Por tal motivo para la elección de este elemento es importante que cumpla con los requerimientos de arquitectura definidos en la *Tabla 7*, teniendo en cuenta que el módulo debe ser compatible con la diadema Neurosky Mobile2.

3.3.1.4 Selección del módulo Bluetooth.

Para la selección de este módulo se toma en cuenta características fundamentales las cuales cumplen con las necesidades del proyecto. En la *Tabla 11*, se realiza la valorización de cada requerimiento para la mejor elección del módulo a utilizar.

Tabla 11

Selección módulo bluetooth

Hardware	Requerimientos		Puntuación total
	SRSH2	SRSH11	
HC-06	1	0	1
HC-05	1	1	2

Elección: Para la elección del módulo que permitirá la comunicación inalámbrica mediante bluetooth, se determina que el dispositivo más óptimo para el desarrollo del proyecto es el módulo HC-05 debido a que cumple con los requerimientos que permite una correcta compatibilidad con el sensor neuronal.

Fuente: Elaborado por el autor

En la *Tabla 12*, se detallan las características técnicas del módulo bluetooth HC-05.

Tabla 12*Características técnicas modulo bluetooth HC-05*

Características	Propiedades
Modulación	GFSK (Gaussian Frequency Shift Keying)
Frecuencia	Banda ISM 2.4 GHz
Interface	Serial TTL
Voltaje de operación	3.6VDC -6VDC
Consumo corriente	50 mA
Potencia de transmisión	4dBm
Sensibilidad	-84 dBm a 0.1% BER
Velocidad	1 Mbps
Baudios Soportados	1200,2400,4800,9600,19200,38400,57600,115200
Seguridad	Autenticación y encriptación (Contraseña 0000 o 1234)

Fuente: (Geek Factory, 2014)

3.3.1.5 Hardware para bloques de adquisición y procesamiento de datos:

Este bloque es fundamental debido a que en el mismo se procede a desarrollar tanto la adquisición de los datos mediante un microprocesador específico que permite la interconexión entre el sensor EEG y este microcontrolador obteniendo los datos necesarios, para posteriormente llevar a cabo el procesamiento de estos datos adquiridos mediante un hardware robusto, ya que este va a ser el encargado de desarrollar un análisis y proceso de los datos complejo mediante diversas técnicas de análisis, y finalmente realizar las acciones válidas para la

detección del estado de somnolencia y su correcta comunicación a los diferentes dispositivos a interactuar en la alerta electrónica. En la Tabla 7 se especifica los requerimientos que son necesarios para el buen desempeño del prototipo del sistema embebido.

3.3.1.6 Selección de dispositivo de adquisición de datos.

Para la elección de los dispositivos con el cual se realizará la ejecución del proyecto debe cumplir con las características tanto de adquisición de datos de ser compatible con el módulo hc-05 y poder recibir los datos de este sensor en tiempo real por tanto sus características de hardware deben ser las mejores posibles. La Tabla 13 se realiza la valorización de los requerimientos necesarios para realizar una correcta elección del sistema embebido para el desarrollo eficiente del proyecto.

Tabla 13

Selección de dispositivo para la adquisición de datos

Hardware	Requerimientos				Puntuación total
	SRSH3	SRSH12	SRSH14	SRSH18	
Arduino Uno	1	0	1	1	3
Arduino Mega	1	1	1	1	4
Arduino nano	1	0	1	1	3

Elección: Para la elección del dispositivo el cual será el encargado de la adquisición de los datos se debe cumplir con los parámetros específicos como también el dispositivo debe ser compatible con el módulo HC-05 mismo que es el encargado de la comunicación con el Sensor EEG, por tal motivo todos los dispositivos son eficaces para este proceso pero se realiza la elección del dispositivo de

Arduino Mega debido a que este es el que tiene características de hardware más robustas a los demás dispositivos por lo cual el funcionamiento de esta etapa será óptima.

Fuente: Elaborado por el autor

A continuación, en la Tabla 14 se muestran las características técnicas más importantes del dispositivo Arduino Mega.

Tabla 14

Características técnicas Arduino mega

Características	Propiedades
Microcontrolador	ATmega2560
Voltaje de operación	5v
Voltajes de entrada mínimo y máximo	6-20v
Pines E/S digitales	54 (15 salida PWM)
Pines entrada analógica	16
Memoria Flash	256 KB, 8 KB para el gestor de arranque (bootloader)
SRAM	8KB
EEPROM	4KB
Frecuencia de reloj	16 MHz

Fuente: (Arduino Mega 2560, 2018)

3.3.1.7 Selección de dispositivo de procesamiento de datos.

Para la elección del mejor dispositivo para la realización del procesamiento de los datos es necesario elegir uno que se adecue tanto en su software como hardware para la realización de análisis de datos de manera robusta y proceder a ejecutar acciones y conexiones de elementos electrónicos de manera externa.

Tabla 15

Dispositivo de procesamiento de datos.

Dispositivo	Requerimientos									Total
	SRSH4	SRSH7	SRSH9	SRSH11	SRSH12	SRSH15	SRSH17	SRSH18	SRSH19	
Arduino Uno	1	1	0	0	0	1	1	0	1	5
Arduino Mega	1	1	1	0	0	1	1	1	1	7
Raspberry pi 3B+	1	1	1	1	1	1	1	1	1	9
UDOO X86	1	1	0	1	0	1	0	1	0	5
Orange pi 5	1	1	0	1	1	1	1	1	0	7

Elección: Para la elección del dispositivo que será el encargado del procesamiento de los datos como también de la ejecución de acciones para activar elementos electrónicos externos se debe tener en cuenta el cumplimiento de diversos parámetros para este proceso por lo cual se determina que el Raspberry pi 3B+ es el adecuado para realizar estos procesos ya que cumple de manera satisfactoria los requerimientos necesarios para un buen funcionamiento del prototipo.

Fuente: Elaborado por el autor

A continuación, en la Tabla 16 se muestran las características técnicas más importantes del dispositivo Raspberry pi 3B+.

Tabla 16

Características Raspberry Pi 3

Características	Propiedades
Procesador	Broadcom BCM2837, Cortex-A53(ARMv8)64-bitSoC
Frecuencia de reloj	1,2 GHz
GPU	Video Core Iv 400MHz
Memoria	1GB LPDDR2SDRAM
Conectividad de Red	Fast Ethernet10/100 Gbps
Puertos	4 USB 2.0
	Micro-SD
	HDMI
Pines externos	GPIO 40 pines

Fuente: (Xataka, 2018)

3.3.2 Elección de software

El software con el cual se desarrollará el proyecto debe estar correlacionado a los requerimientos establecidos en la Tabla 7 garantizando el desempeño eficaz del sistema.

3.3.2.1 Selección de Software de Programación.

Para la elección del software que sea adecuado debe cumplir con los requerimientos establecidos en la Tabla 7 de arquitectura del sistema. A continuación, en la Tabla 17 se muestra

cada requerimiento establecido y que debe cumplir el software de programación que se va a elegir.

Tabla 17

Selección de software de programación

Software	Requerimientos				Puntuación total
	SRSH18	SRSH19	SRSH21	SRSH22	
Arduino IDE	1	1	1	1	4
MicroPython	1	1	1	0	3
Python	1	1	1	1	4

Elección: Para la elección del software de programación se determina que tanto Arduino IDE como Python son óptimos para el desarrollo del proyecto ya que estos cumplen con todos los requerimientos establecidos ya que Arduino ide será el encargado de la adquisición de los datos mientras que Python será el software con el cual se realiza el procesamiento de los datos aplicando diversas técnicas.

Fuente: Elaborado por el autor

3.4 Fase 2: Diseño

Una vez finalizado la etapa en el cual se realiza el análisis de la situación actual, la información obtenida mediante la encuesta realizada permitió determinar los requerimientos tanto técnicos como operacionales que necesita el proyecto para su ejecución de manera idónea. Por tal motivo a continuación para iniciar la fase de diseño en el siguiente apartado se realiza la descripción de forma breve la arquitectura funcional del sistema en conjunto con el diagrama de bloques del mismo, posteriormente dando a conocer los componentes tanto de hardware como de

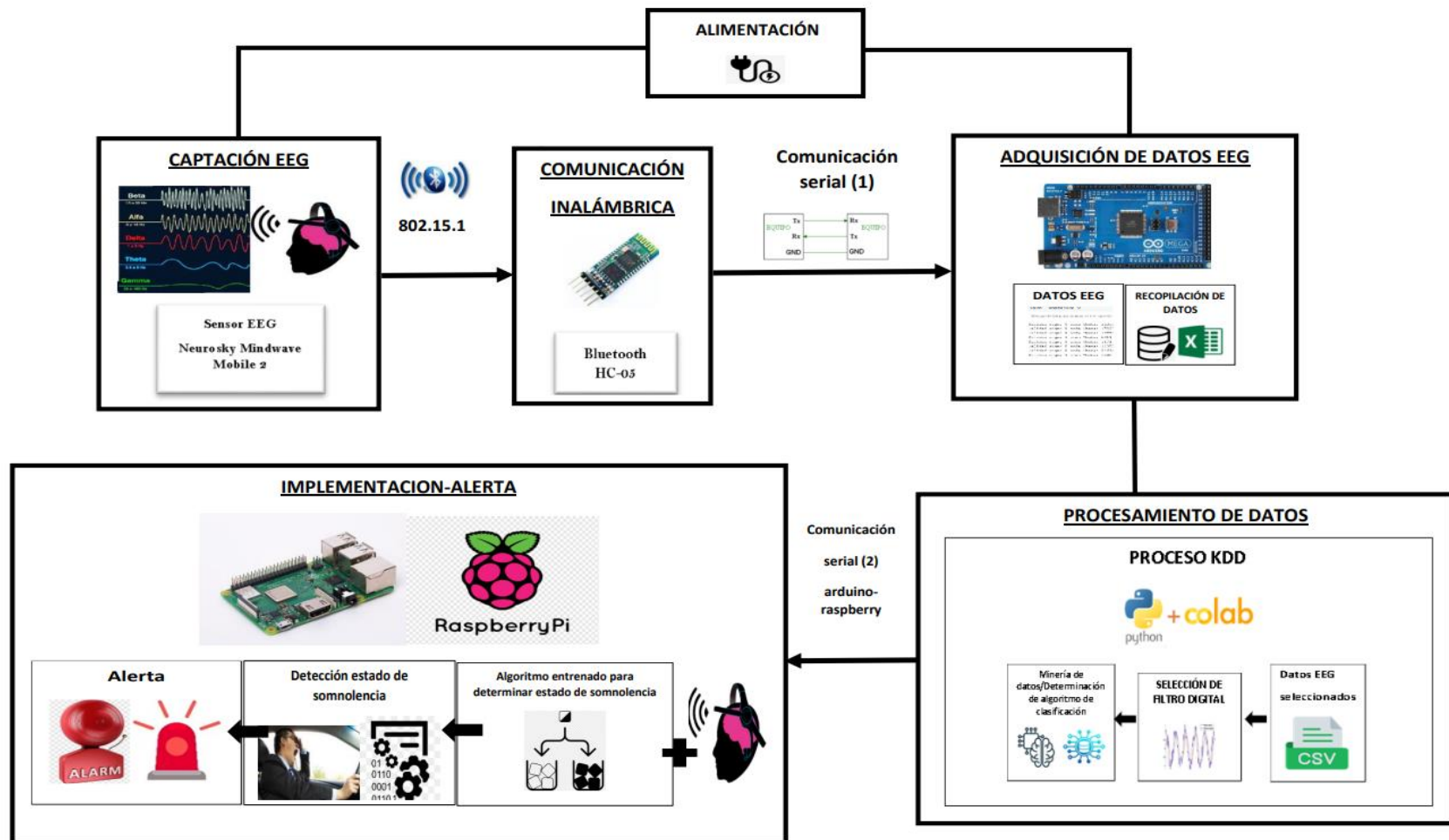
software que se utilizaran para la elaboración y ejecución del sistema mismos que serán escogidos de manera rigurosa con los requerimientos antes que deben cumplir; después se mostrara el diagrama del circuito como también el diseño del sistema y finalmente el diagrama de flujo del sistema completo.

3.4.1 Arquitectura del sistema

En la Figura 17, se observa el sistema de adquisición de las ondas cerebrales, en este caso es mediante el sensor Mindwave Neurosky Mobile 2 el mismo que es el que se encarga de la toma de los datos, dicho sensor se encuentra conectado de manera inalámbrica al microcontrolador el mismo que se encarga de recibir la información para luego procesarla y transformarla para la posterior detección del estado de somnolencia del usuario.

Figura 17

Arquitectura del sistema



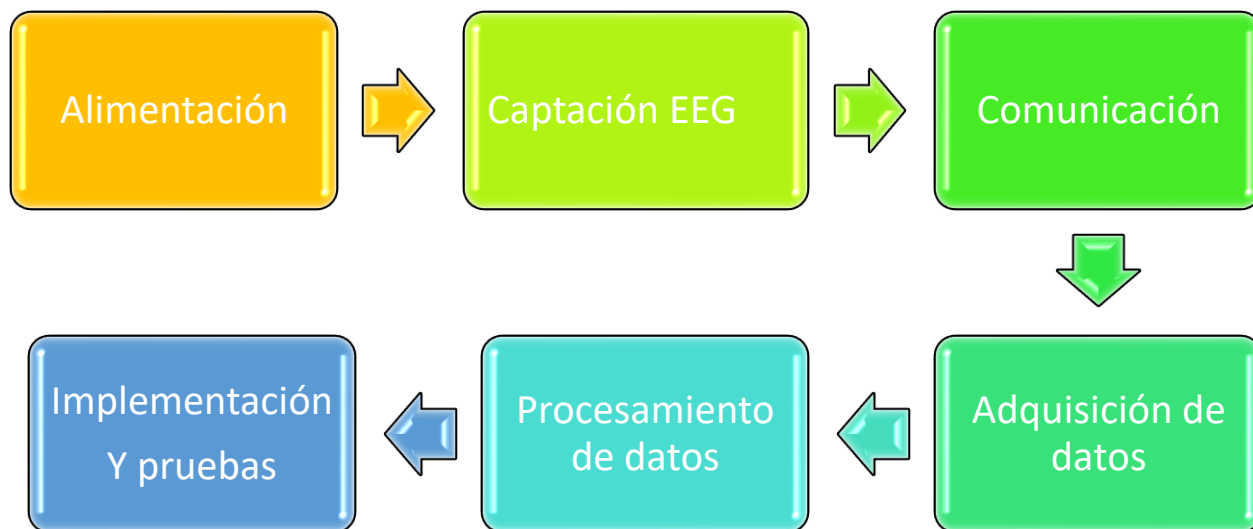
Fuente: Elaborado por el autor

3.4.2 Diagrama de bloques

La Figura 18, muestra el diagrama de bloques en donde visualiza cada etapa que se debe realizar para el desarrollo deseado del proyecto, por lo cual en cada bloque se muestra específicamente características propias que estos deben realizar para que así el funcionamiento del proyecto marche de la mejor manera.

Figura 18

Diagrama de bloques de funcionamiento del sistema



Fuente: Elaborado por el autor

Para tener un mejor entendimiento del funcionamiento del sistema embebido de detección del estado de somnolencia se describe a continuación.

- ***Bloque de alimentación.*** – Se definirá de que forma el sistema embebido o nodo central estará alimentado para su funcionamiento indicando el consumo energético de cada componente.
- ***Bloque de captación EEG.*** – Esta sección trata del uso correcto del sensor EEG indicando en donde va tanto el sensor y la tierra definiendo así su correcto funcionamiento para captar datos del usuario de manera correcta.
- ***Bloque de comunicación.*** – En esta etapa se realiza el proceso de emparejamiento del módulo bluetooth hc-05 con el neurosky indicando cada paso para que la comunicación se encuentre de manera correcta indicando como modo maestro al módulo hc-05 y modo esclavo al sensor.
- ***Bloque adquisición de datos.*** - En esta etapa se procede a realizar la codificación necesaria en Arduino ide para proceder a obtener los datos en el monitor serial del software estos datos serán: atención, ondas cerebrales (theta, Alpha, beta, gamma) mismas que se almacenan en una base de datos csv.
- ***Bloque de procesamiento de datos.*** –para proceder a realizar el respectivo tratamiento de estos datos mediante el proceso KDD implementando diferentes softwares que permitan este preproceso de datos como el uso de Google Colab en el cual se aplicara su respectivo filtraje de los datos y obtener una csv mejor para acceder a la etapa de minería de los datos en donde se realiza el respectivo entrenamiento y pruebas en los algoritmos de clasificación para la detección del estado de somnolencia en el usuario.

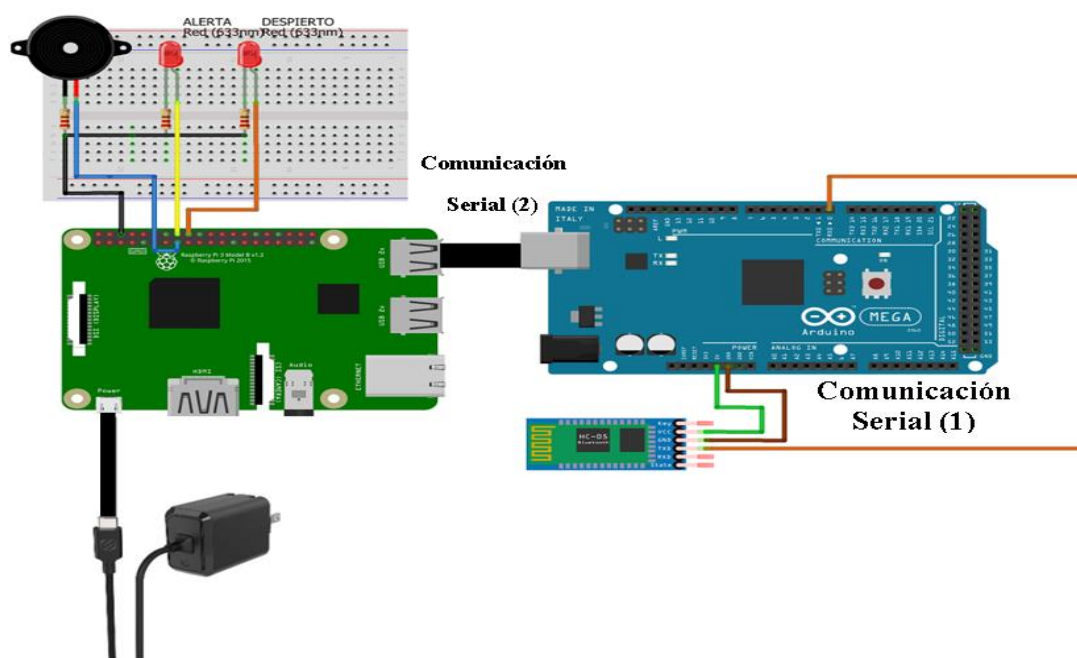
- **Bloque implementación.** – Para esta etapa final del proyecto se realiza la implementación del prototipo en el automóvil en el cual el microcontrolador actuará de manera autónoma en la detección del estado de somnolencia y si se encuentra el usuario en este estado automáticamente se procederá a ejecutar la activación de la alerta para el usuario en este caso es una alerta.

3.4.3 Diagrama esquemático del sistema

El sistema embebido es el encargado de realizar la comunicación con el sensor neuronal en donde se envían las diferentes señales a tratar como se muestra en la Figura 19.

Figura 19

Diagrama esquemático del sistema

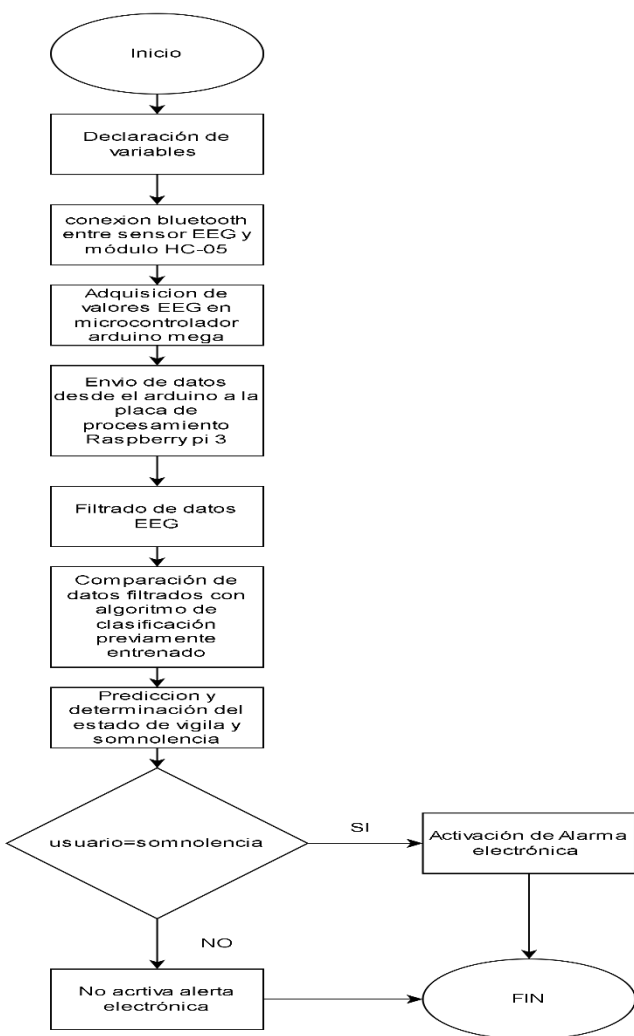


Fuente: Elaborado por el autor

El proceso que se irá desarrollando de manera secuencial en el prototipo para cada uno de los procesos a través de cada dispositivo que interactúa, se presenta en un diagrama de flujo, mismo que va describiendo el paso a paso de las acciones que realiza el sistema para la monitorización del estado de somnolencia en conductores como se muestra en la Figura 20.

Figura 20

Diagrama de flujo del sistema general



Fuente: Elaborado por el autor

3.5 Fase 3: Codificación y pruebas del sistema

3.5.1 Bloque 1: Alimentación del sistema

3.5.1.1 Cálculos para alimentación eléctrica

Para una correcta alimentación del sistema embebido se realiza una sumatoria de todos los elementos que van a intervenir en el proyecto, por tal motivo en la Tabla 18 se realiza el listado de los dispositivos con sus respectivos voltajes y corrientes necesarios para su correcto funcionamiento determinados por sus respectivos datasheet.

Tabla 18

Consumo de corriente y voltaje del sistema

Dispositivo	Corriente (mA)	Voltaje de operación	Voltaje de alimentación
Arduino Mega	Por cada pin 40 mA 5=200 mA	5V	5-12 V
Raspberry Pi3B+	360 mA	5V	5V
Modulo bluetooth			
Hc-05	50 mA	3V	5V
Buzzer	30 mA	5V	5V
2 led	Por led 30 mA	5V	5V
			Voltaje 5V

Fuente: Elaborado por el autor

A continuación, se realiza la sumatoria de corriente que ocupa cada dispositivo.

$$I_T \sum_{i=1}^n I_i$$

$$I_T = 5I_1 + I_2 + I_3$$

$$I_T = (5(40) + 50 + 90 + 360)mA$$

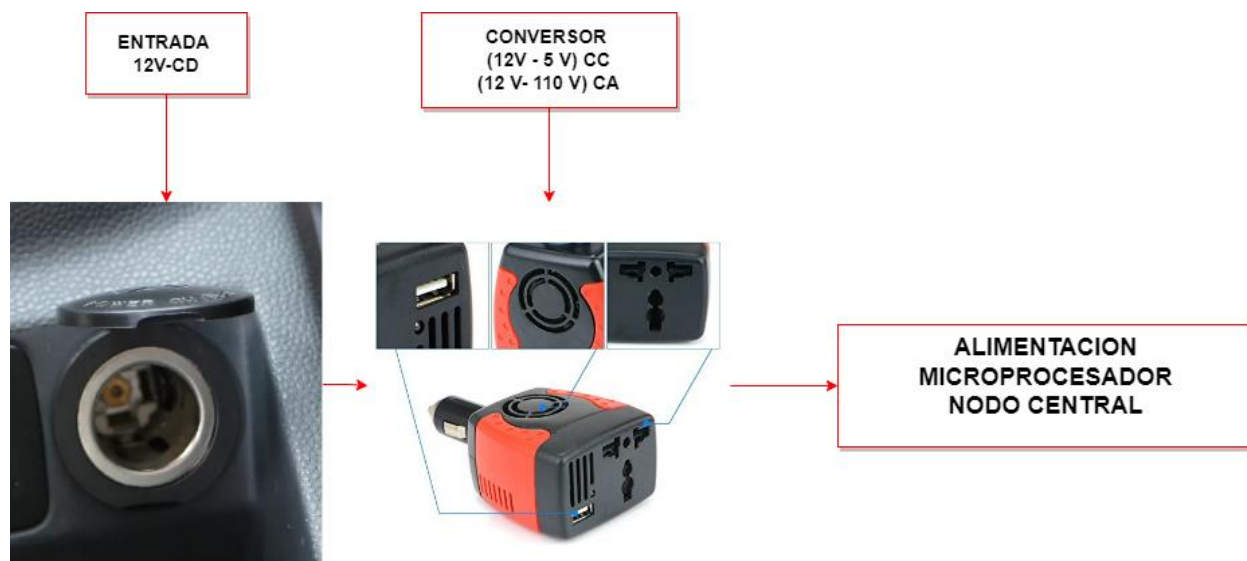
$$I_T = (200 + 50 + 90 + 360)mA$$

$$I_T = 700 mA$$

En la alimentación del sistema, es importante tener en cuenta que los automóviles por lo general cuentan con puertos de entrada para encendido mismo que emite 12 V, por tal motivo es necesario el conversor de 12 V a 5 V o a su vez de 12 v a 110 v como se muestra en la Figura 21 para así proceder a conectar el dispositivo de procesamiento del nodo central mediante el puerto USB que emite 5V de corriente continua sin problemas a quemar ningún dispositivo.

Figura 21

Alimentación del sistema

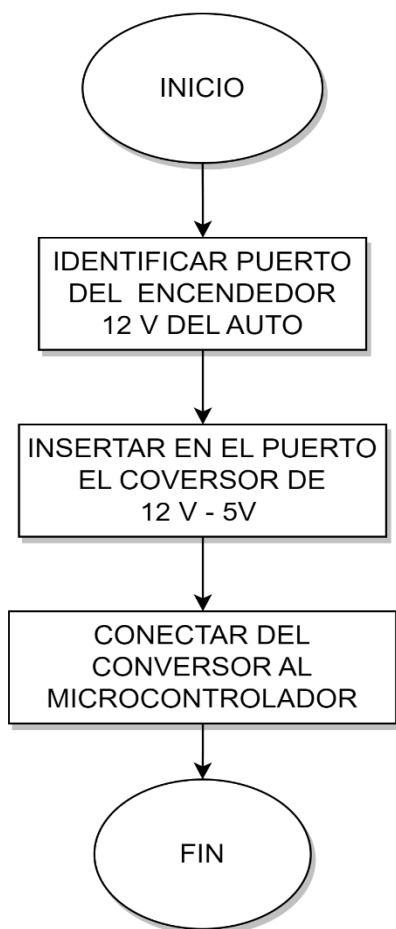


Fuente: Elaborado por el autor

A continuación, se muestra el proceso a llevar a cabo para la realizar la alimentación eléctrica para el prototipo mediante el uso de un conversor de 12Vcc a 5Vcc, como se muestra en la Figura 22.

Figura 22

Diagrama de flujo bloque de alimentación



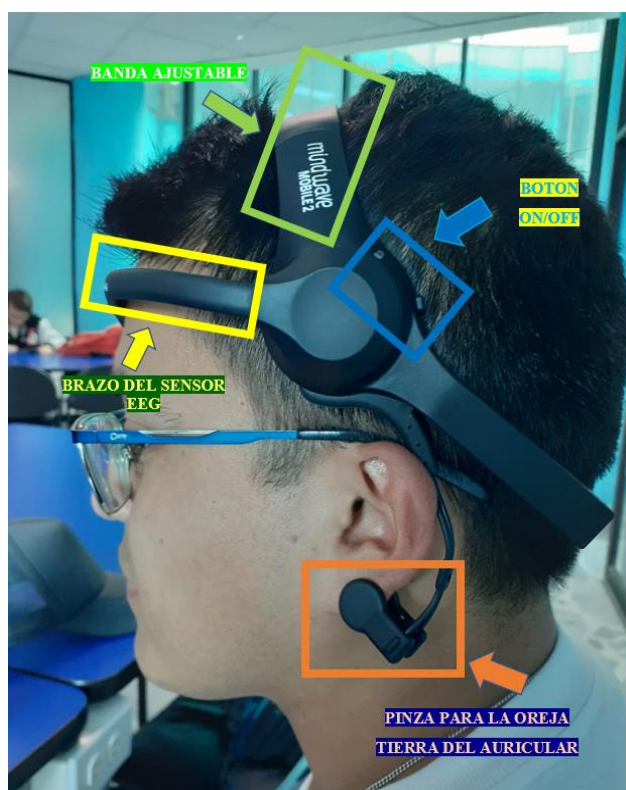
Fuente: Elaborado por el autor

3.5.2 Bloque 2: Captación EEG

Para este bloque se realiza la captación de los datos con el Neurosky Mindwave Mobile 2 de la actividad cerebral, por ende, las frecuencias neuronales que se están generando en tiempo real por el usuario y también las señales EEG que se generan en la parte de la frente del usuario por medio del electrodo seco de este dispositivo. Por lo cual es indispensable el colocar correctamente el sensor; está se debe encontrar bien asegurada en la sien derecha e izquierda del usuario ya que el sensor de electromiografía está a 3cm de la ceja del usuario o a 2 dedos de distancia como se observa en la Figura 23.

Figura 23

Componentes de Neurosky Mindwave Mobile 2

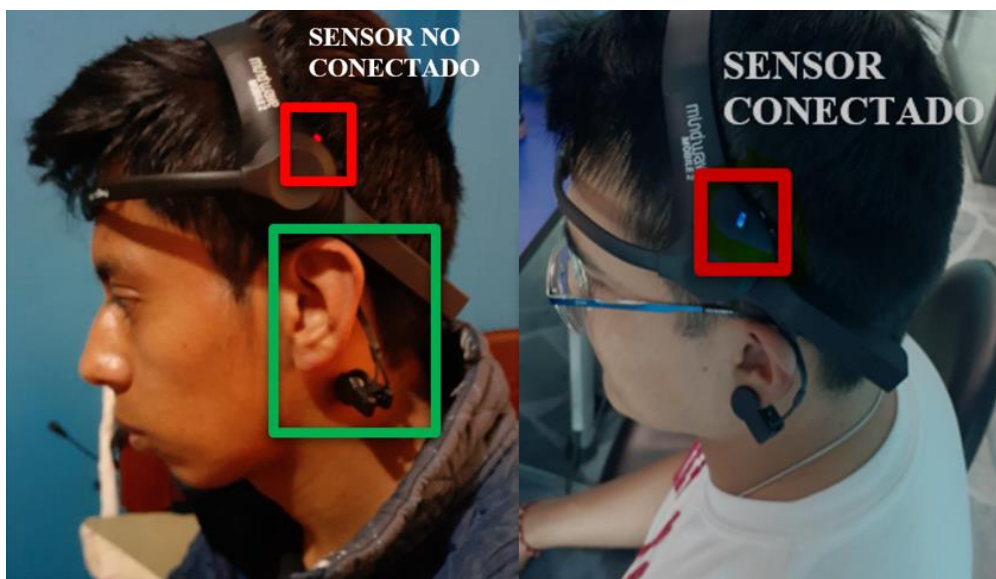


Nota. Identificación de componentes de la diadema Neurosky Mindwave Mobile 2 y su correcta colocación en el usuario.

Una vez identificado cada componente de la diadema es importante el conocer también los estados que tiene el sensor ya que si no se coloca de manera correcta cada componente como lo es la pinza de tierra misma que permite la reducción del ruido o el led indicador se encuentra de color rojo quiere decir que la batería de alimentación se encuentra agotada por tal motivo el dispositivo no captara ningún dato mientras que si el led indicador se encuentra de color azul indica que el sensor se encuentra en correcto funcionamiento y los datos son captados como se muestra en la Figura 24.

Figura 24

Identificación estados del sensor, izquierda sensor no conectado y derecha sensor conectado.

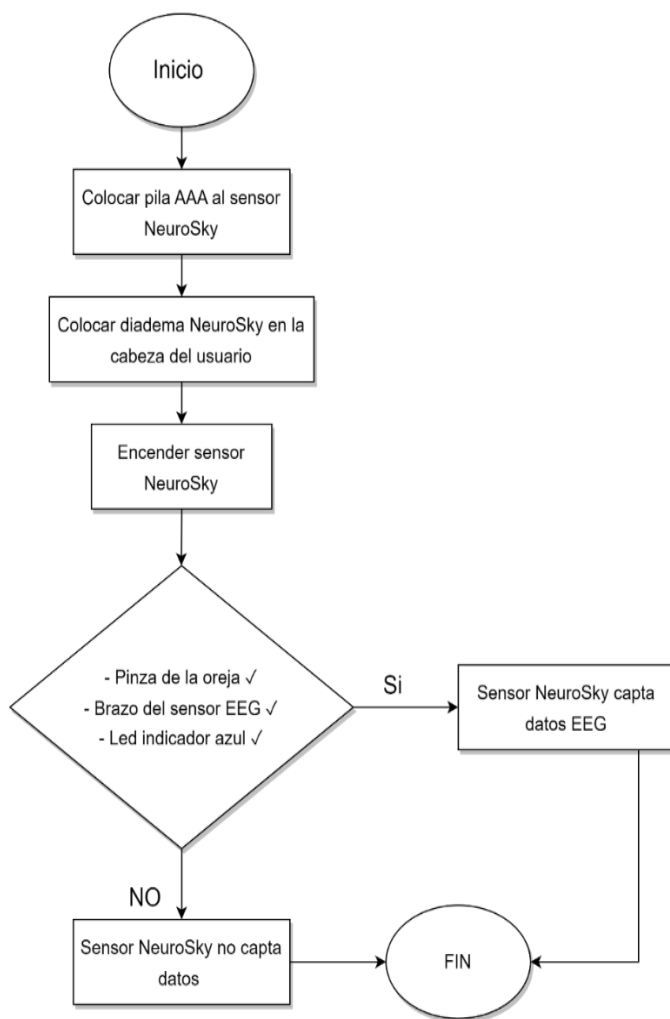


Nota. Identificación de los estados que emite el sensor de acuerdo al nivel de la batería de alimentación el led rojo indica que la batería se está agotando posibles errores de lectura; led azul indica que la batería está cargada y el sensor lee datos correctamente.

A continuación, en la Figura 25 se observa el diagrama de flujo que indica el proceso a llevar a cabo para el bloque de captación de datos en donde se debe realizar diversas acciones para el correcto funcionamiento del sensor.

Figura 25

Diagrama de bloque de captación de datos



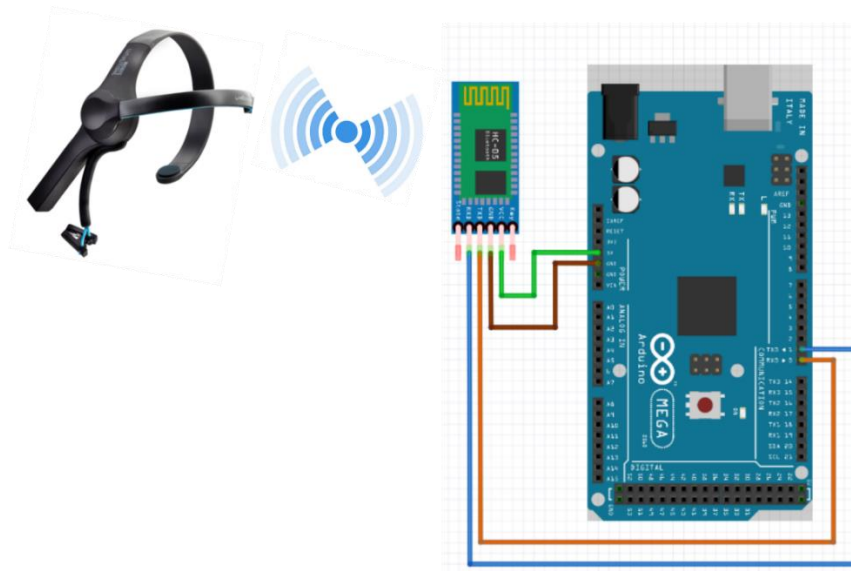
Fuente: Elaborado por el autor

3.5.3 Bloque 3: Comunicación

En este bloque se realiza la respectiva sincronización bluetooth del sensor EEG con el módulo de Arduino HC-05 para el acceder a los datos que capta el sensor como se observa en la Figura26.

Figura26

Sincronización bluetooth



Fuente: Elaborado por el autor

Por tanto, es necesario configurar el módulo HC-05 de tal manera que se encuentre en modo maestro. Este proceso se lo realiza mediante la utilización de los comandos AT, permitiendo así la comunicación entre el modem y usuario. Para ello es necesario realizar 4 configuraciones dentro del módulo las cuales son: nombre, rol (maestro, esclavo), contraseña y velocidad de comunicación (baudios) estos parámetros serán configurados dentro del software Arduino en su modo monitor serial como se muestra en la Figura 27.

Figura 27

Configuración módulo hc-05 con neurosky

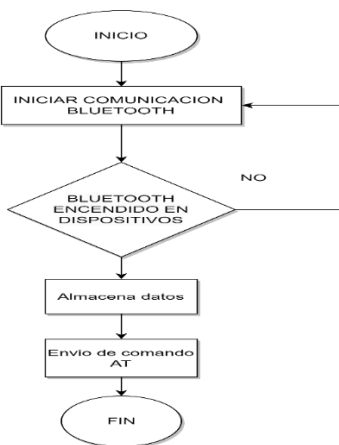


Fuente: Elaborado por el autor

En la Figura 28, se muestra el proceso que se realiza para realizar una comunicación bluetooth adecuada en donde mediante el uso de comandos AT para establecer parámetros adecuados de configuración que permite la correcta comunicación entre el módulo hc-05 con el sensor EEG.

Figura 28

Comunicación bluetooth



Fuente: Elaborado por el autor

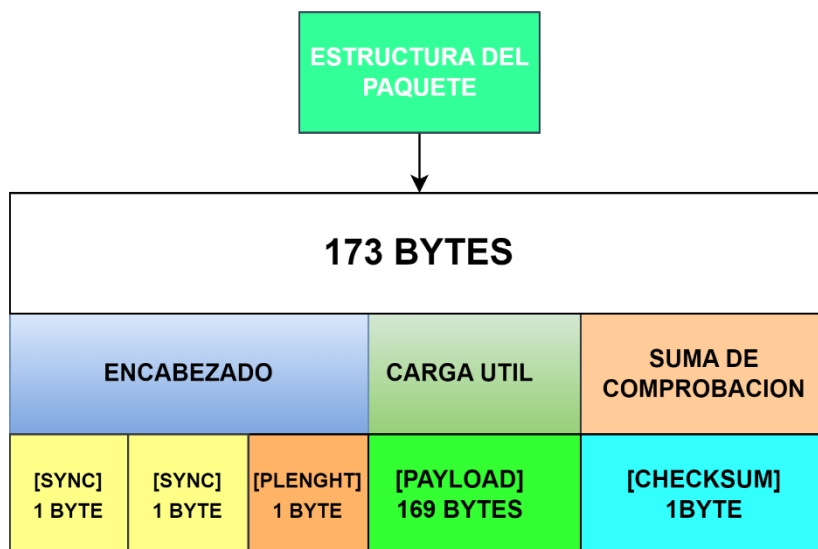
3.5.4 Bloque 4: Adquisición de datos EEG

3.5.4.1 Estructura del paquete ThinkGear

La estructura del paquete que envía el dispositivo ThinkGear el cual consta de un paquete el cual contiene 173 bytes totales en el cual esta información se divide en 3 campos que son: cabecera de 3 bytes siendo 2 bytes para sincronización y 1 byte pleno, 169 bytes los cuales son para la carga útil en donde se tendrá la información a utilizar y 1 byte del Checksum o de suma de comprobación con el fin de que la información sea integra como se muestra en la Figura 29.

Figura 29

Estructura de paquete ThinkGear



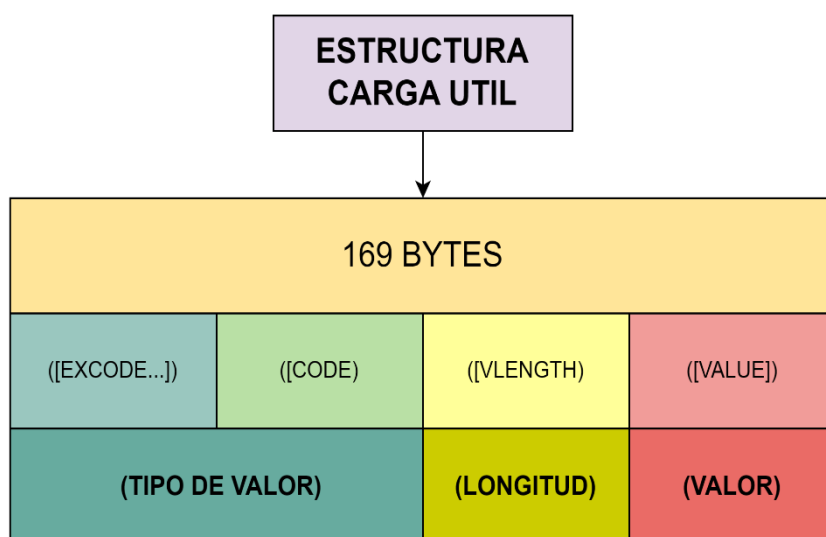
Nota: El paquete ThinkGear se encuentra estructurado en tres partes permitiendo así la representación e interpretación de información enviada por el sensor. (*Neurosky.com, 2017*)

En la Figura 30, se puede observar cómo se encuentra conformado la carga útil del paquete ThinkGear, el código extendido [EXCODE] son bytes con valor a su vez la cantidad de bytes determina el nivel del código extendido habitualmente para este apartado se encuentra

determinado por el EXCODE [0X55] es exclusivo reservado para código extendido y el EXCODE [0xAA] es reservado únicamente para valores de sincronización. Ahora el apartado [CODE] en conjunto con el código extendido determina el tipo de valor de los datos codificados a obtener y su interpretación debida como se muestra en la Tabla 19.

Figura 30

Estructura carga útil



Nota: La estructura de la carga útil es importante conocer ya que los valores tanto del EXCODE como CODE se usan de acuerdo a la versión del chip ThinkGear que se utiliza y si se encuentran habilitados para dichos dispositivos. (*Neurosky.com, 2017*)

En la Tabla 19 se puede visualizar los diferentes tipos de CODEC ya sean estos de un solo byte como también de varios bytes mismos que se encuentran actualmente para los dispositivos ThinkGear con su respectiva interpretación o significado de los datos obtenidos para su posterior uso adaptándose a la investigación necesaria del caso.

Tabla 19*Códigos ThinkGear*

CODIGOS DE UN SOLO BYTE			
NIVEL DE CODIGO EXTENDIDO	[CODE]	BYTE [LONGITUD]	SIGNIFICADO DE LOS DATOS
0	0x02	-	Calidad de la señal (0-255)
0	0x03	-	Frecuencia cardiaca
0	0x04	-	Nivel de atención eSense (0-100)
0	0x05	-	Nivel de meditación eSense (0-100)
0	0x06	-	Valor de onda (0-255)
0	0x07	-	Marcador de inicio de sesión (0)
CODIGOS DE VARIOS BYTES			
NIVEL DE CODIGO EXTENDIDO	[CODE]	BYTE [LONGITUD]	SIGNIFICADO DE LOS DATOS
0	0x80	2	Valor crudo de onda de un solo big endian. Valor con signo de 16 bits (-32768 a 32767) EEG_POWER: ocho big endian de 4 bytes.
0	0x81	32	Valores de puntos flotantes IEEE 754 que representan los valores de las ondas (delta, theta, low Alpha, high Alpha, low beta, high beta, low gamma, mid gamma) en valores de potencia.
0	0x83	24	ASIC_EEG: Información de ocho valores sin signo representada en formato big endian de 3-bytes de las ondas (delta, theta, low Alpha, high Alpha, low beta, high beta, low gamma, mid gamma).
0	0x86	2	Entero sin signo en formato big-endian de dos bytes que representan los milisegundos.
ANY	0x55	-	Nunca usado reservado para [EXCODE]
ANY	0xAA	-	Nunca usado reservado para [SYNC]

Fuente: (Neurosky.com, 2017)

3.5.4.2 Adquisición de datos

Una vez que la comunicación entre el sensor neuronal y el módulo bluetooth se encuentran correctos, esta información se la debe obtener por tramas definiendo mediante codificación en Arduino la estructura del paquete ThinkGear en el cual se debe analizar desde los 2 byte de sincronización para posteriormente verificar la longitud del payload mismo que no tiene que ser mayor a 169, ni tampoco igual a 170 ya que este valor pertenece a EXACODED [0xAA] de sincronización como se puede observar en la Figura 31.

Figura 31

Codificación de estructura del paquete ThinkGear

```

95 void loop() {
96     // mirar los bytes de sincronizacion
97     if(ReadOneByte() == 0xAA) { //Nunca utilizado re
98     if(ReadOneByte() == 0xAA) { //Nunca utilizado re
99         payloadLength = ReadOneByte();
100        if(payloadLength > 169)
101            return;
102
103        payloadChecksum = 0;
104        for(int i = 0; i < payloadLength; i++) {
105            payloadData[i] = ReadOneByte();
106            payloadChecksum += payloadData[i];
107        }
108

```

Nota: Codificación de la recepción de la estructura del paquete ThinkGear en donde se encuentran los 2 bytes de sincronización [0xAA] la carga útil de 169 y el byte del CHECKSUM de validación de los datos.

Se procede a realizar la codificación con el fin de obtener datos específicos de acuerdo a los códecs analizados en la Tabla 19 como lo es la obtención de los valores de la atención, meditación, calidad de la señal y los valores de las ondas EEG como se muestra en la *Figura 32*.

Figura 32

Activación de envío de información ThinkGear

```

brainwave=false;
for(int i = 0; i < payloadLength; i++) { // inicializa los tipos de datos que se van a obtener
  switch (payloadData[i]) {
    case 2: // valor de [CODE] es 0x02 envia el valor de calidad de la señal (0-255)
      i++;
      poorQuality = payloadData[i];
      bigPacket = true;
      break;
    case 4: // valor eSense de atencion (0-100)
      i++;
      attention = payloadData[i];
      break;
    case 5: // valor eSense de meditacion (0-100)
      i++;
      meditation = payloadData[i];
      break;
    case 0x80:
      i = i + 3;
      break;
    case 0x83: // activacion de envio de ondas EEG (delta,theta,alpha,beta,gaamma)
      i++;
      brainwave = true;
      byte vlen = payloadData[i];
      //mySerial.print(vlen, DEC);
      //mySerial.println();
      read_waves(i+1);
      i += vlen; // i = i + vlen
  }
}

```

Fuente: Elaborado por el autor

Ahora se procede a realizar la respectiva codificación de impresión el cual nos permita visualizar los datos generados en el sensor EEG mediante la comunicación serial del Arduino dicha codificación se puede observar en la *Figura 33*.

Figura 33

Impresión de datos obtenidos del sensor Neurosky

```

| if(brainwave && attention > 0 && attention < 100) { //procede a leer los datos si las variables se encuentran en los rangos establec
Serial.print("Atencion:");
  Serial.print(attention, DEC); //imprime datos de nivel de atencion eSense en formato decimal
  Serial.print("\t");
Serial.print("Calidad sign: ");
  Serial.print(poorQuality ); //imprime datos de nivel de calidad de señal en formato decimal
  Serial.print("\t");
Serial.print(" Onda Delta ");
  Serial.print(delta_wave, DEC); //imprime datos de onda delta en formato decimal
  Serial.print("\t");
Serial.print(" Theta: ");
  Serial.print(theta_wave, DEC); //imprime datos de onda theta en formato decimal
  Serial.print("\t");
Serial.print(" Low Alpha: ");
  Serial.print(low_alpha_wave, DEC); //imprime datos de onda low alpha en formato decimal
  Serial.print("\t");
Serial.print(" High Alpha: ");
  Serial.print(high_alpha_wave, DEC); //imprime datos de high alpha en formato decimal
  Serial.print("\t");
Serial.print(" low beta:"); //estado vigila1 //imprime datos de onda low beta en formato decimal
  Serial.print(low_beta_wave, DEC);
  Serial.print("\t");
Serial.print(" high beta: "); //estado vigila2 ;imprime datos de onda high beta en formato decimal
  Serial.print(high_beta_wave, DEC);
  Serial.print("\t");
Serial.print("low gamma:");
  Serial.print(low_gamma_wave, DEC); //imprime datos de onda low gamma en formato decimal
  Serial.print("\t");
Serial.print(" mid gamma:");
  Serial.print(mid_gamma_wave, DEC); //imprime datos de onda mid gamma en formato decimal
  Serial.print("\n");

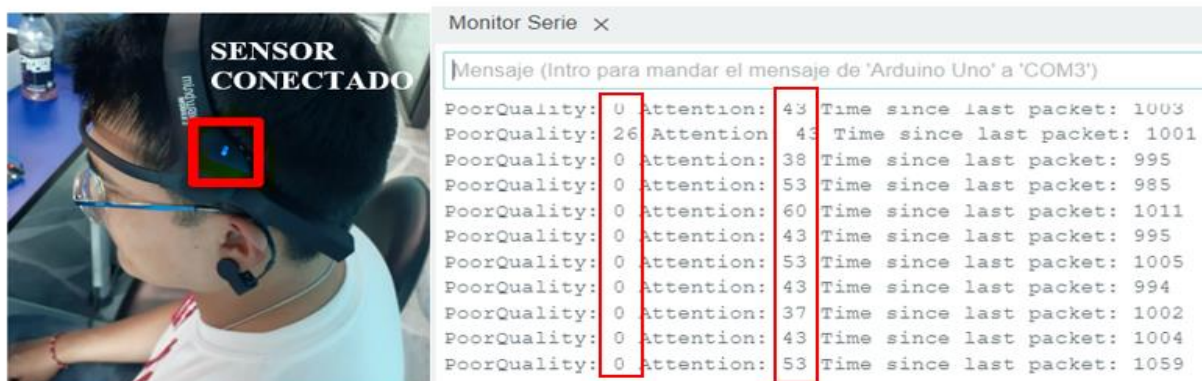
```

Fuente: Elaborado por el autor

Para la obtención de los datos ya configurada la recepción de la información la misma se tiene que verificar que el sensor emita la luz azul la cual es la indicadora que el sensor se encuentra en modo conectado y si este se encuentra de manera correcta como se muestra en la figura se puede realizar la visualización de los estados en que se encuentra el usuario estos pueden ser de concentración y meditación los cuales varían de 0 a 100 como se muestra en la Figura 34.

Figura 34

Izquierda usuario colocado sensor correctamente, derecha datos de calidad de señal y atención.

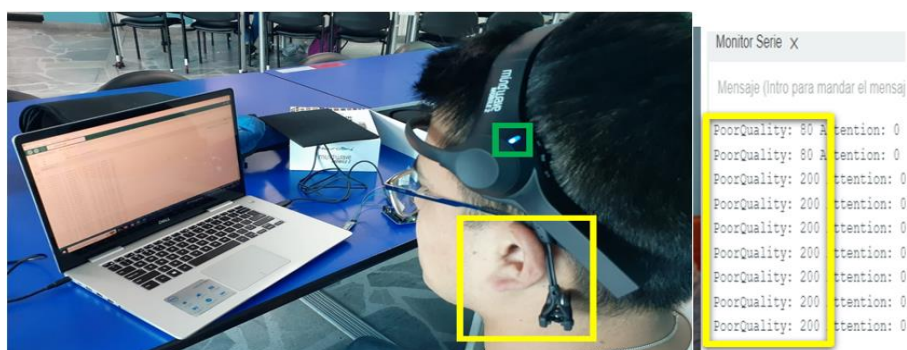


Nota: Los valores obtenidos en la calidad de la señal mientras más bajos son la señal se encuentra mejor y si los valores de atención van entre (40-60) se encuentra en un rango aceptable o neutro.

Quando la masa del sensor no se encuentra conectada al usuario, en la calidad de la señal se enviará el valor más alto teniendo la peor calidad de la señal y el sensor no detectará ningún tipo de señal como se observa en la Figura 35.

Figura 35

Verificación de calidad de señal en el peor estado, izquierda usuario colocado sensor sin pinza eliminadora de ruido, derecha datos obtenidos de calidad de señal.



Fuente: Elaborado por el autor

Se procede a obtener los datos de las ondas EEG (delta,theta,low alpha,high alpha,low beta,high beta,low gamma,mid gamma) mismas que son las cuales sirven para realizar la base de datos para el analisis del proyecto como se observa en la Figura 36.

Figura 36

Lectura de datos del sensor Neurosky mediante monitor serial de Arduino.

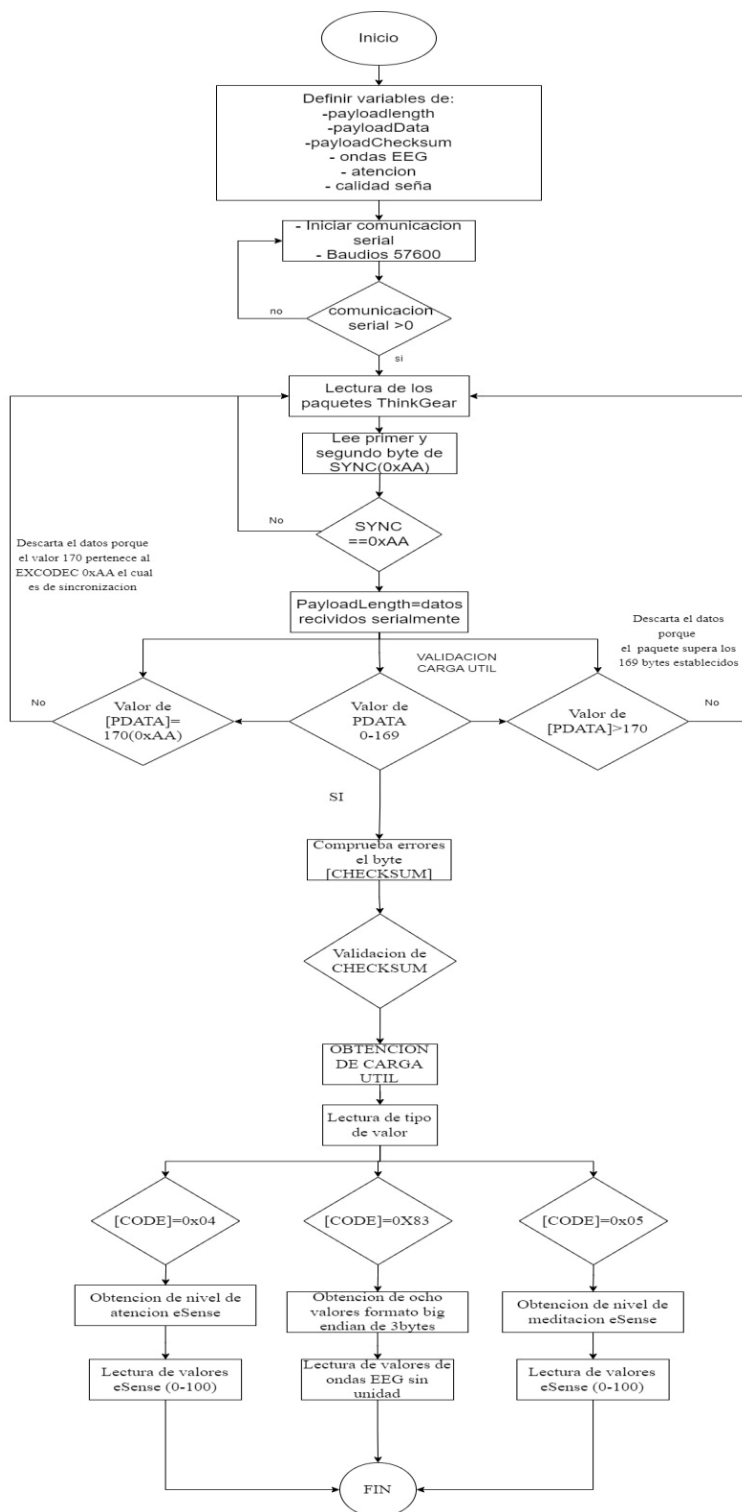
Salida Monitor Serie X								Ambos NL & CR	57
Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM3')									
Atencion:56	Calidad sign: 0	Onda Delta 35684	Theta: 36802	Low Alpha: 38756	High Alpha: 23528	low beta:33335	high beta: 20601	low gamma:7288	
Atencion:66	Calidad sign: 0	Onda Delta 65535	Theta: 41456	Low Alpha: 8865	High Alpha: 9347	low beta:8637	high beta: 5982	low gamma:5475	
Atencion:61	Calidad sign: 0	Onda Delta 15312	Theta: 41285	Low Alpha: 2739	High Alpha: 20384	low beta:6432	high beta: 4641	low gamma:2510	
Atencion:64	Calidad sign: 0	Onda Delta 6178	Theta: 14259	Low Alpha: 14653	High Alpha: 29504	low beta:2113	high beta: 9189	low gamma:25848	
Atencion:44	Calidad sign: 0	Onda Delta 62030	Theta: 15670	Low Alpha: 18246	High Alpha: 27995	low beta:15862	high beta: 14080	low gamma:23501	
Atencion:41	Calidad sign: 0	Onda Delta 26928	Theta: 21116	Low Alpha: 52303	High Alpha: 29443	low beta:19889	high beta: 11152	low gamma:20419	
Atencion:48	Calidad sign: 0	Onda Delta 3995	Theta: 28369	Low Alpha: 7567	High Alpha: 6075	low beta:8543	high beta: 6828	low gamma:34356	
Atencion:60	Calidad sign: 0	Onda Delta 29061	Theta: 28859	Low Alpha: 30872	High Alpha: 40460	low beta:6944	high beta: 23616	low gamma:11329	
Atencion:70	Calidad sign: 0	Onda Delta 28730	Theta: 27766	Low Alpha: 3003	High Alpha: 11277	low beta:31023	high beta: 17877	low gamma:17337	
Atencion:70	Calidad sign: 0	Onda Delta 19321	Theta: 56608	Low Alpha: 9262	High Alpha: 10158	low beta:9211	high beta: 5862	low gamma:9236	
Atencion:51	Calidad sign: 0	Onda Delta 48258	Theta: 56994	Low Alpha: 17931	High Alpha: 11427	low beta:27070	high beta: 8318	low gamma:13598	
Atencion:34	Calidad sign: 0	Onda Delta 64644	Theta: 40641	Low Alpha: 5204	High Alpha: 10060	low beta:3429	high beta: 3377	low gamma:6720	
Atencion:17	Calidad sign: 0	Onda Delta 60734	Theta: 23043	Low Alpha: 54799	High Alpha: 7279	low beta:7685	high beta: 2518	low gamma:2706	
Atencion:17	Calidad sign: 0	Onda Delta 30442	Theta: 52904	Low Alpha: 11081	High Alpha: 7277	low beta:9876	high beta: 5481	low gamma:16791	
Atencion:30	Calidad sign: 0	Onda Delta 19964	Theta: 37175	Low Alpha: 9041	High Alpha: 11944	low beta:6372	high beta: 5050	low gamma:15602	
Atencion:40	Calidad sign: 0	Onda Delta 13095	Theta: 58837	Low Alpha: 29866	High Alpha: 3875	low beta:12096	high beta: 5806	low gamma:32062	
Atencion:63	Calidad sign: 0	Onda Delta 27632	Theta: 32559	Low Alpha: 6904	High Alpha: 9496	low beta:12768	high beta: 9554	low gamma:16591	
Atencion:80	Calidad sign: 0	Onda Delta 10358	Theta: 13548	Low Alpha: 1308	High Alpha: 6322	low beta:10071	high beta: 9131	low gamma:16783	
Atencion:97	Calidad sign: 0	Onda Delta 3260	Theta: 3498	Low Alpha: 2397	High Alpha: 2128	low beta:6195	high beta: 4579	low gamma:43160	

Fuente: Elaborado por el autor

Para realizar el proceso de la adquisición de los valores que emite el sensor Neurosky Mindwave Mobile 2 hacia la placa de arduino mega es necesario el realizar la codificación en dicha placa de procesamiento en donde se irá ejecutando paso a paso la adquisición de valores EEG a través de los módulos involucrados en esta etapa. En la Figura 37, se puede observar el diagrama de flujo en el cual se detalla todo el proceso de la adquisición de valores EEG en la placa arduino mega.

Figura 37

Diagrama de flujo de la adquisición de datos



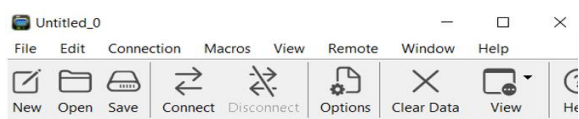
Fuente: Elaborado por el autor

3.5.4.3 Registro de los datos adquiridos

Para realizar el registro de los valores que son captados e interpretados por el microprocesador que se muestran en la Figura 36 se lo realiza mediante la utilización del software CoolTerm el cual al ser un programa con una interfaz de fácil uso como se muestra en la Figura 38.

Figura 38

Interfaz CoolTerm

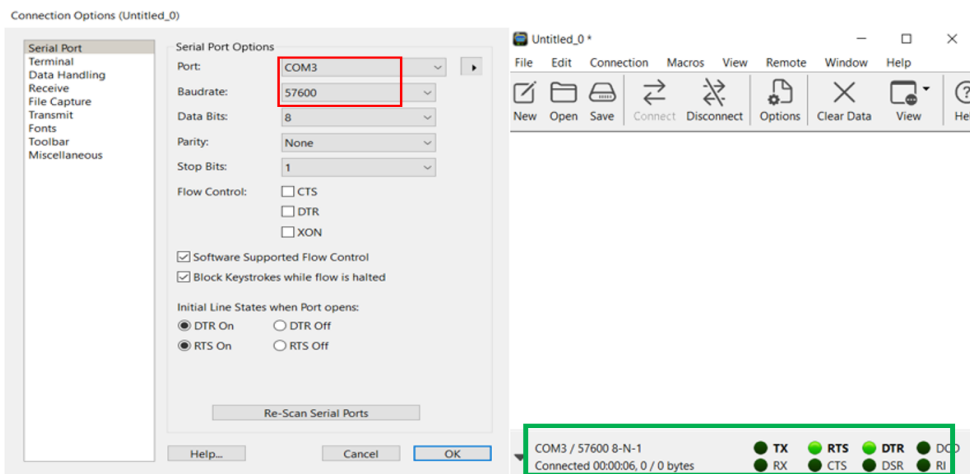


Fuente: Elaborado por el autor

En la Figura 39, se realiza el proceso de configuración del software para la conexión correcta con el microprocesador para este caso se debe configurar el puerto COM al que se encuentra conectado el microprocesador en este caso es el puerto 3 además es importante que la tasa de baudios sea la misma configurada tanto para el microprocesador como para el software CoolTerm en este caso es de 57600.

Figura 39

Izquierda configuraciones comunicación serial, derecha validación de conexión.

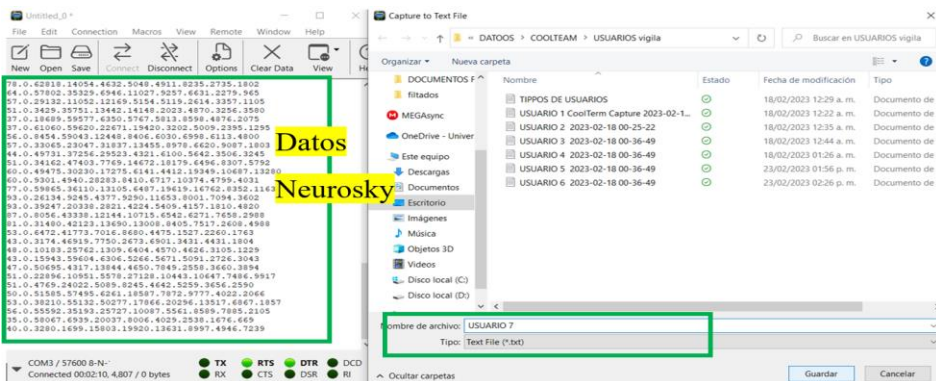


Fuente: Elaborado por el autor

Una vez realizado la configuración de conexión correcta se procede a configurar para que el software guarde los datos que se van a captar en un documento en formato txt como se muestra en la Figura 40.

Figura 40

Izquierda adquisición de datos EEG, derecha almacenamiento de BDD por usuario.



Fuente: Elaborado por el autor

3.5.5 *Bloque 5: Procesamiento de datos*

En este apartado se realiza el desarrollo del método con el cual se efectúa la clasificación de los datos adquiridos a través del sensor neurosky para el proyecto. El proceso para el procesamiento de los datos será de acuerdo con la metodología o proceso KDD representada en la Figura 13 por tal motivo es indispensable seguir en orden cada etapa de este proceso.

3.5.5.1 Selección de datos

Para el proceso de selección de los datos se tiene en cuenta los estados en los que el usuario puede estar para así poder discriminar los valores de manera correcta por tal motivo en este caso se realiza selección de los datos en los siguientes estados.

- Estado vigila del usuario
- Estado de relajación
- Estado de sueño

3.5.5.1.1 *Estudio de datos en estado de vigila*

Los datos obtenidos son cuando el usuario se encuentra en un estado normal de sus actividades como también en conducción. Debido a la gran cantidad de datos tomados a continuación en la *Tabla 20* se muestra los primeros datos almacenados para este estado.

Tabla 20

BDD estado vigila

atención	calidad señal	delta	theta	low Alfa	high alfa	low beta	high beta	low gamma	mid gamma
44	0	49871	26342	5544	7865	12138	6177	929	2354
75	0	51434	34443	5633	14688	7322	3103	4799	5185

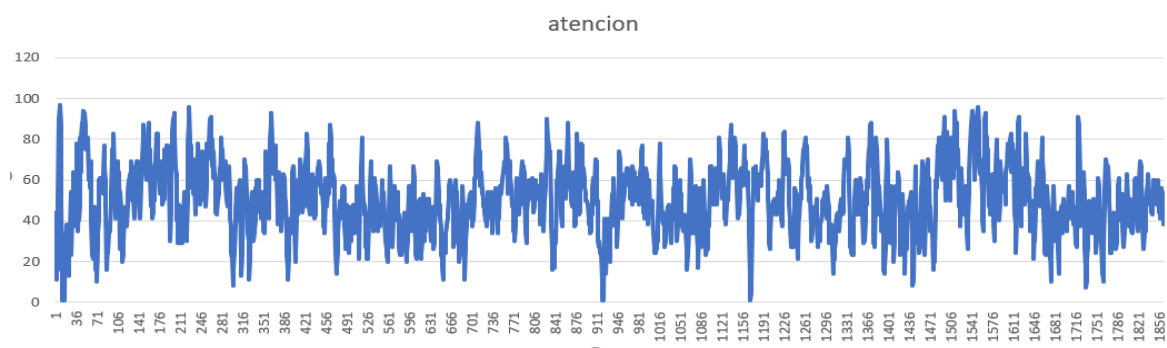
75	0	33097	57096	5704	7430	6401	3486	2830	1860
81	0	10776	17907	8584	7354	4268	7975	1871	4494
80	0	23183	37824	4501	4436	5135	4680	6951	4233
81	0	14306	19761	6952	11206	6224	12723	9090	12283
60	0	36000	10998	49364	12522	23444	26571	4160	5987
66	0	9602	20749	3033	7146	5364	10421	5821	3807
63	0	44198	3718	22917	20395	25696	19270	18876	14511
56	0	34030	5463	5319	2080	716	2149	1472	2491
69	0	15027	49585	15253	11380	10840	7963	2530	2285

Fuente: Elaborado por el autor

A continuación, se muestran los gráficos de las ondas en estado puro quiere decir sin realizar ningún proceso dichos gráficos serán representados en el estado (amplitud vs tiempo (cantidad de datos)).

Figura 41

Señal de atención escenario vigila



Fuente: Elaborado por el autor

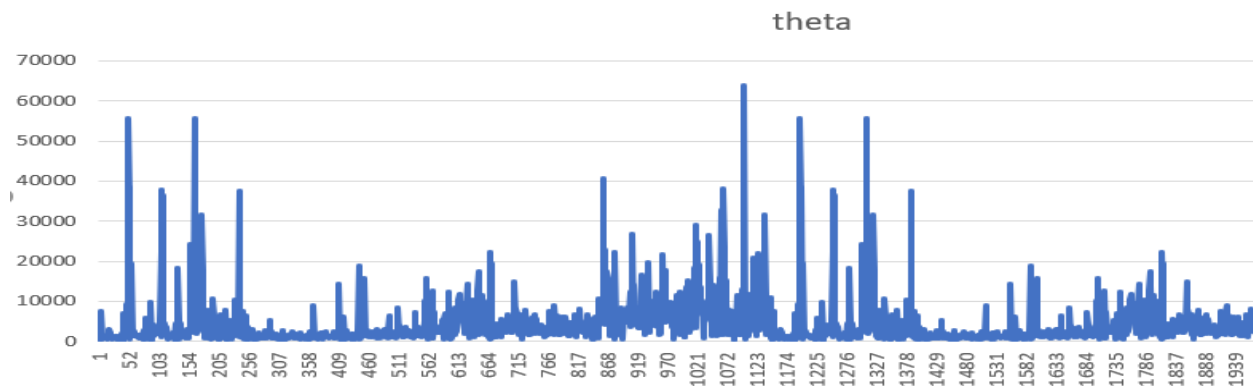
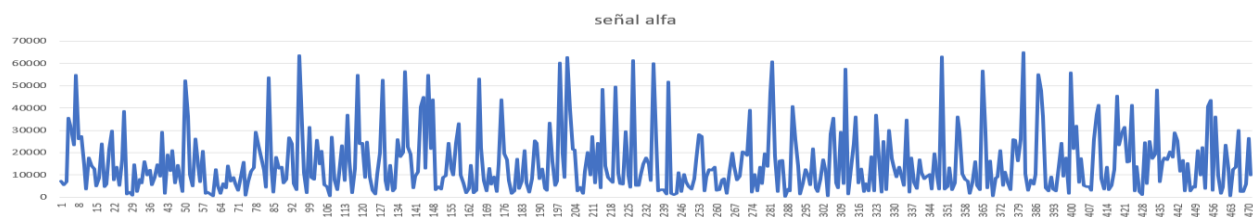
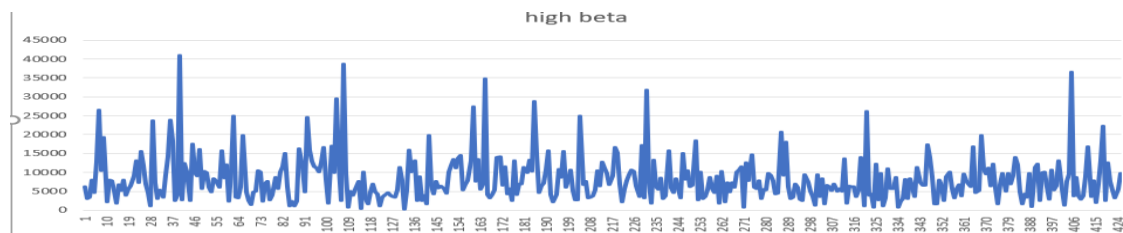
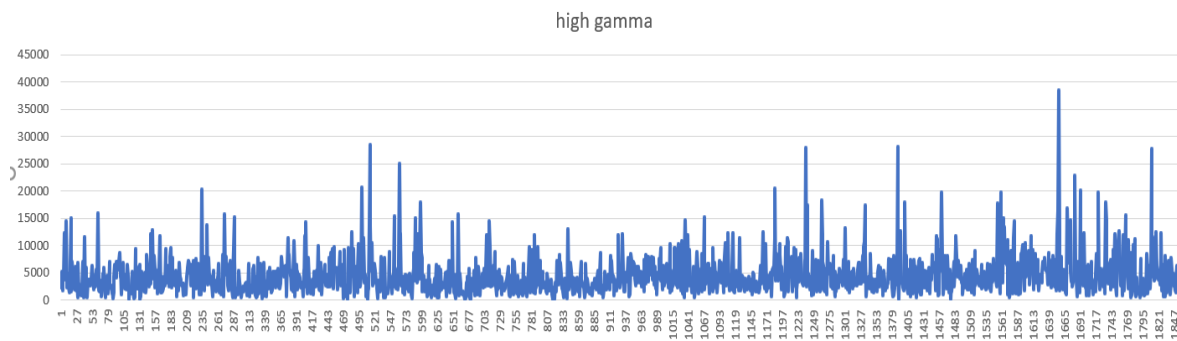
Figura 42*Señal theta escenario vigila**Fuente:* Elaborado por el autor**Figura 43***Señal Alpha escenario vigila**Fuente:* Elaborado por el autor**Figura 44***señal beta escenario vigila**Fuente:* Elaborado por el autor

Figura 45*señal gamma escenario vigila**Fuente:* Elaborado por el autor

Se puede observar en las figuras 41,42,43,44,45 que los niveles de actividad de las señales son diferentes por tal motivo en la onda tanto theta como Alpha se identifica una menor actividad debido a que estas ondas trabajan más cuando el usuario se encuentra en un estado cercano al sueño o relajación mientras que las ondas gamma, beta tienen mayor actividad porque se encuentra en su hábitat de trabajo de estas ondas. También es importante el identificar que el nivel de atención del usuario para este escenario es mayor a 20 lo que indica que este se encuentra en un estado activo.

3.5.5.1.1 Estudio de datos en estado de relajación

Para efectuar este estado en el usuario se lo realiza mediante un ambiente tranquilo en donde el usuario no realice ninguna acción por tal motivo es indispensable que este no tenga tampoco distracciones por ende este estado se lo efectúa en un cuarto en donde se trata de tener el menor ruido posible para que el usuario pueda relajarse de buena manera y obtener datos correctos en este escenario por lo cual a continuación en la tabla se muestran algunos de los datos obtenidos para este estado.

Tabla 21*BDD estado de relajación*

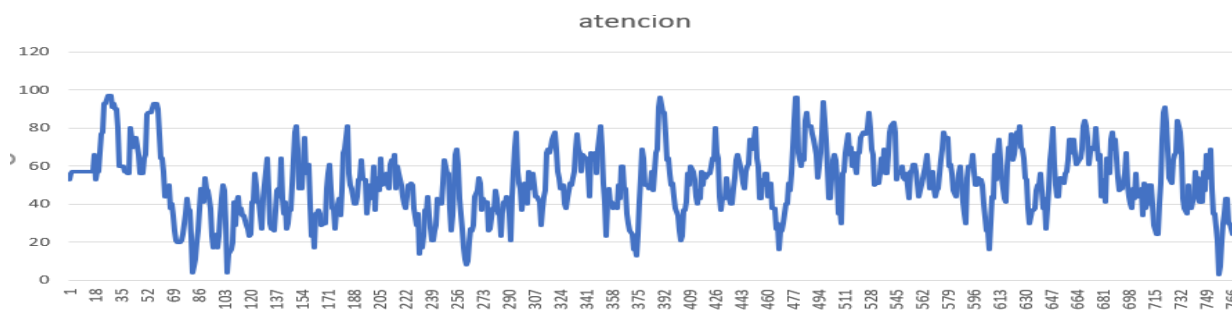
atención	calidad señal	delta	theta	low Alfa	high alfa	low beta	high beta	low gamma	mid gamma
41	0	34494	30405	26080	16059	14189	12621	11296	4832
53	0	1840	35701	47323	13359	27462	27247	39843	8500
51	0	16684	39471	25345	5037	9868	4293	8110	11607
51	25	39392	5622	992	10598	6554	5382	220	354
57	0	18100	3108	28398	8780	6619	12499	2776	593
57	26	54162	65089	20870	34052	62183	20902	10507	2312
57	26	11036	21259	6202	3878	3184	919	480	234
57	25	41020	24744	8424	39076	51986	58179	16371	15110
57	26	2624	40943	17522	9114	4075	1380	976	416
57	51	33275	13697	5238	6226	2788	3443	869	261
56	0	34743	30004	21346	5853	4376	4553	52290	4824

Fuente: Elaborado por el autor

A continuación, se muestra de manera grafica los datos obtenidos para este estado en cada una de las señales a realizar el respectivo análisis, mismos que se encuentran representados en (amplitud vs tiempo (cantidad de datos)).

Figura 46

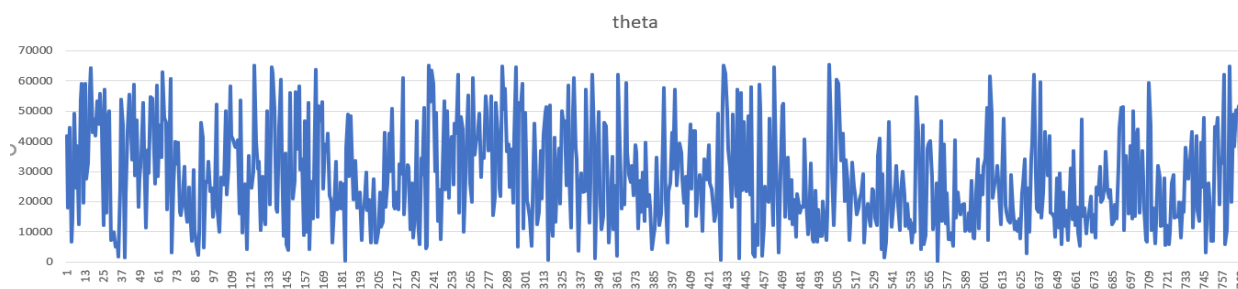
señal atención escenario relajación



Fuente: Elaborado por el autor

Figura 47

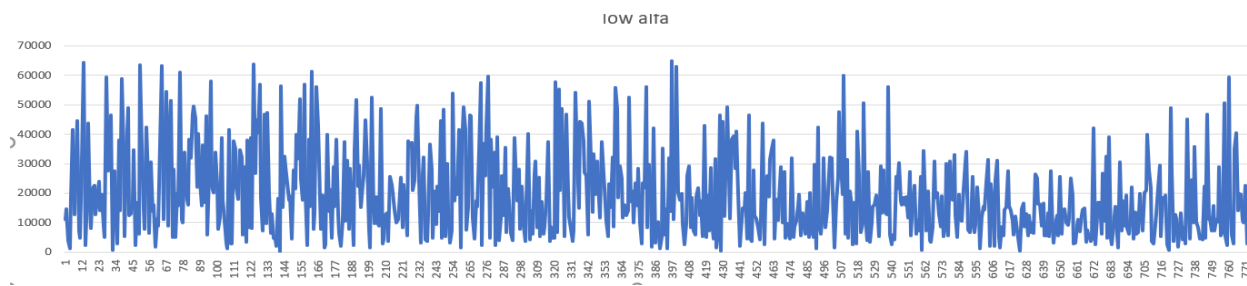
señal theta escenario relajación



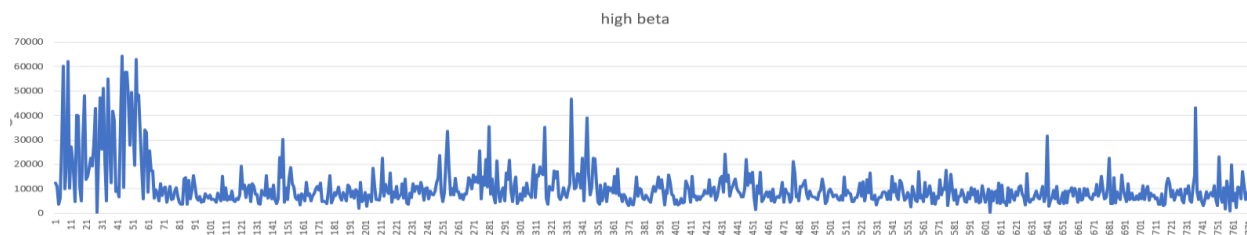
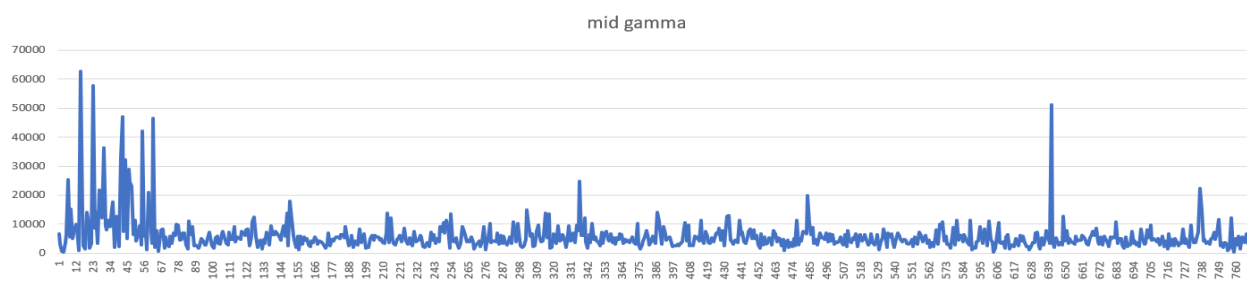
Fuente: Elaborado por el autor

Figura 48

señal alfa escenario relajación



Fuente: Elaborado por el autor

Figura 49*señal beta escenario relajación**Fuente:* Elaborado por el autor**Figura 50***señal gamma escenario relajación**Fuente:* Elaborado por el autor

Para el análisis en el escenario que el usuario se encuentra en un ambiente de relajación se puede observar en las figuras 46,47,48,49,50 que los valores son diferentes al escenario antes planteado debido a que para los niveles de atención se puede identificar que en este caso son de picos más bajos lo cual indica que el usuario no se encuentra en un buen nivel de atención; como también la actividad en las ondas theta y alfa son de mayor amplitud a diferencia de las otras ondas debido que el usuario se encuentra en un estado de relajación y posiblemente de sueño.

3.5.5.1.2 Estudio de datos en estado de sueño

Para realizar la toma de estos datos se la procede a realizar en horas específicas con lo son en la madrugada en el rango de 1am a 5am en donde el usuario presenta síntomas de Sueño como también él se realizó la toma de datos en los usuarios que se encontraban en estado de sueño profundo en este caso interrumpiendo su sueño para que la persona se encuentra en un estado entre dormido y despierto.

A continuación, en la *Tabla 22* se muestra un conjunto de datos los cuales representan al usuario en estado de somnolencia al ser la cantidad demasiado grande de los datos se presenta una cantidad reducida de estos.

Tabla 22

BDD estado de sueño

atención	calidad señal	delta	theta	low Alfa	high alfa	low beta	high beta	low gamma	mid gamma
54	0	16139	50997	13531	5708	5038	6390	5766	1617
23	0	56095	2469	19696	64745	34819	8758	14109	2742
10	0	24144	11091	2769	997	682	394	187	277
1	0	2746	25602	13551	10259	10405	2473	3931	3594
1	0	46329	14310	17159	28889	14595	8678	6002	5914
7	0	52828	10312	7762	59667	11154	6157	3059	4606
29	0	10289	37085	35177	17086	9204	12997	3135	5108
43	0	33172	26609	22087	53352	6662	8865	3474	2358
48	0	23529	26937	2656	13009	4149	3636	2452	2633
54	0	43652	22676	48821	15040	11677	11340	6625	2895

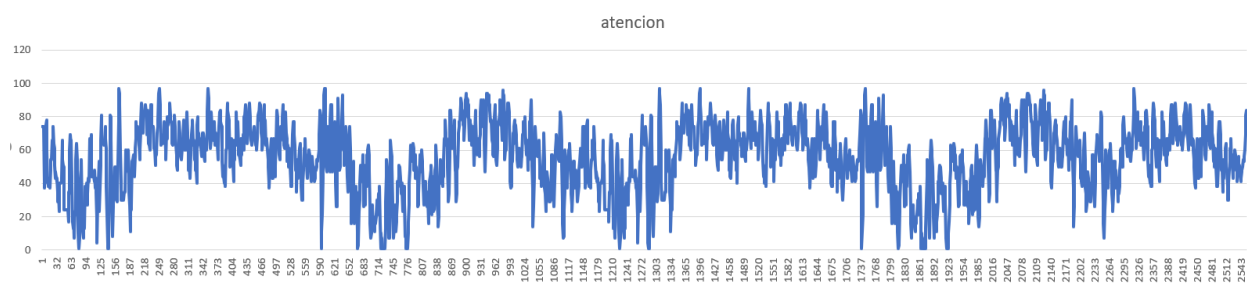
27	0	14235	28598	3197	3372	3807	3967	835	887
----	---	-------	-------	------	------	------	------	-----	-----

Fuente: Elaborado por el autor

A continuación, se muestra de manera grafica los datos obtenidos para este estado en cada una de las señales a realizar el respectivo análisis, mismos que se encuentran representados en (amplitud vs tiempo (cantidad de datos)).

Figura 51

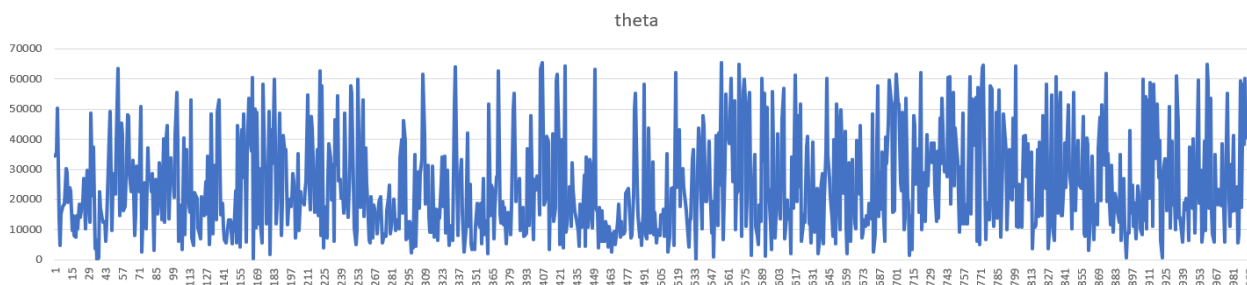
señal atención escenario sueño



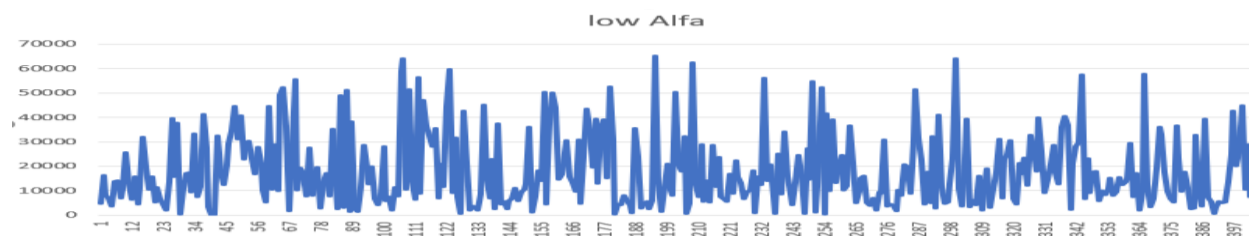
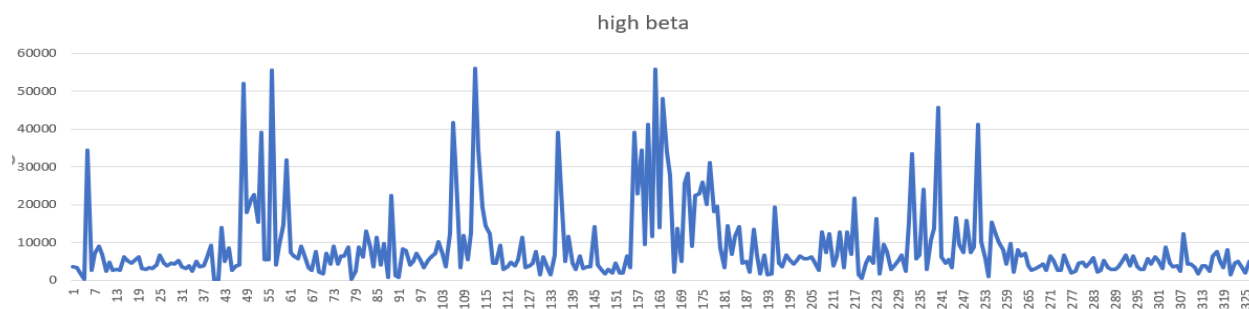
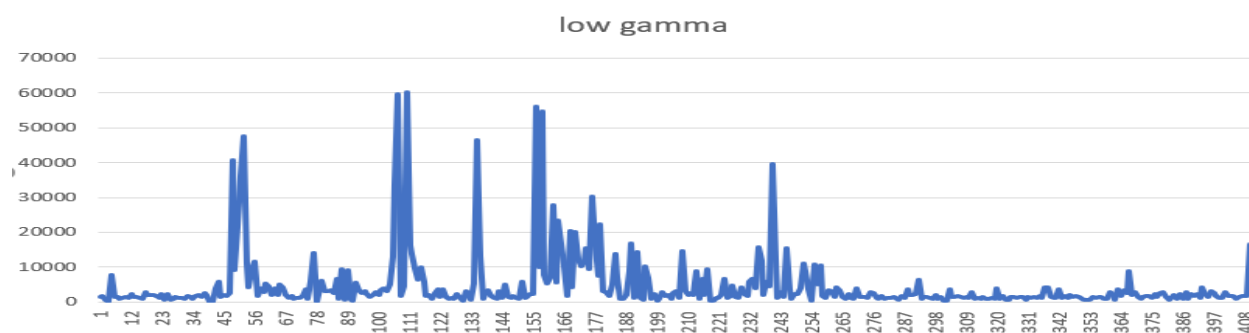
Fuente: Elaborado por el autor

Figura 52

señal theta escenario sueño



Fuente: Elaborado por el autor

Figura 53*señal alfa escenario sueño**Fuente:* Elaborado por el autor**Figura 54***señal beta escenario sueño**Fuente:* Elaborado por el autor**Figura 55***señal gamma escenario sueño**Fuente:* Elaborado por el autor

Como se puede observar en las figuras 51,52,53,54,55 la actividad para cada onda de igual manera es diferente en cada escenario para este caso las ondas tanto theta como Alpha son las que mayor actividad tienen con menos intervalos de puntos altos y bajos de oscilación siendo algo más constantes, mientras que la onda gamma es irregular ya que parte de un punto bajo de datos a un punto sumamente alto de manera brusca por cual se la denota algo irregular.

3.5.5.2 Limpieza de datos mediante uso de Filtros Digitales:

Una vez que se tiene los datos en el csv se procede a realizar el filtrado de los mismos para poder eliminar el exceso de variaciones en la señal que pueden ser producidos por diversos motivos como los son movimientos involuntarios, mal colocación del sensor entre otros

Por tal motivo se procede a realizar el filtrado de los datos obtenidos mediante la herramienta de Google Colab procediendo a ingresar tanto la base de datos como las librerías.

3.5.5.2.1 *Cálculo de valores SRN, SDT en BDD sin filtrar estado de vigila*

Ahora se procese a realizar el filtrado de la BDD en estado vigila como se mostró en la *Tabla 20* se obtuvo los datos en donde el usuario se encuentra en un estado alerta realizando diversas actividades entre ellas conduciendo.

En la *Tabla 23*, se muestran los valores tanto SRN, SDT, mean cuando los datos se encuentran sin realizar ningún proceso de limpieza para realizar la comparación necesaria a los diversos filtros a utilizar para la elección del mejor filtro que se adapte a los datos.

Tabla 23*Valores SRN, SDT de BDD estado de vigila*

	Atención	Theta	Low alfa	High alfa	Low beta	Low gamma	Medd gamma
mean	52.936	28788.448	15323.536	13436.016	11556.862	4780.302	4209.138
Desviación típica (SDT)	18.8563	18111.05473	13952.37793	10798.82919	10167.60978	3737.440754	3024.569497
SNR (mean/sdt)	2.80734162	1.589551157	1.098274149	1.244210439	1.136635085	1.279030843	1.391648631
Coefficiente de variación (sdt/mean)	0.3562	0.6291	0.9105	0.8037	0.8798	0.7818	0.7186

*Fuente: Elaborado por el autor***3.5.5.2.2 Cálculo de valores SRN, SDT en BDD sin filtrar estado de somnolencia**

Ahora se procede a realizar el filtrado de la BDD en estado de sueño como se mostró en la Tabla 22, se obtuvo del usuario en horas determinadas en donde es más pronunciado el sueño en la persona como lo es horarios de 1am a 5 am.

En la Tabla 24, se muestran los valores tanto SRN, SDT, mean cuando los datos se encuentran sin realizar ningún proceso de limpieza para realizar la comparación necesaria a los diversos filtros a utilizar para la elección del mejor filtro que se adapte a los datos.

Tabla 24*Calculo SRN, SDT de BDD sueño*

	Atención	Theta	Low alfa	High alfa	Low beta	Low gamma	Meed gamma
--	----------	-------	----------	-----------	----------	-----------	---------------

mean	57.612	22189.418	17263.02	16356.268	9722.908	4360.38	3139.66
Desviación típica (SDT)	18.8563	18111.05473	13952.37793	10798.82919	10167.60978	7743.161765	6195.590715
SNR (mean/sdt)	3.055322757	1.225186403	1.237281565	1.514633458	0.956262899	0.563126554	0.506757167
Coefficiente de variación (sdt/mean)	0.3273	0.8162	0.8082	0.6602	1.0457	1.7758	1.9733

Fuente: Elaborado por el autor

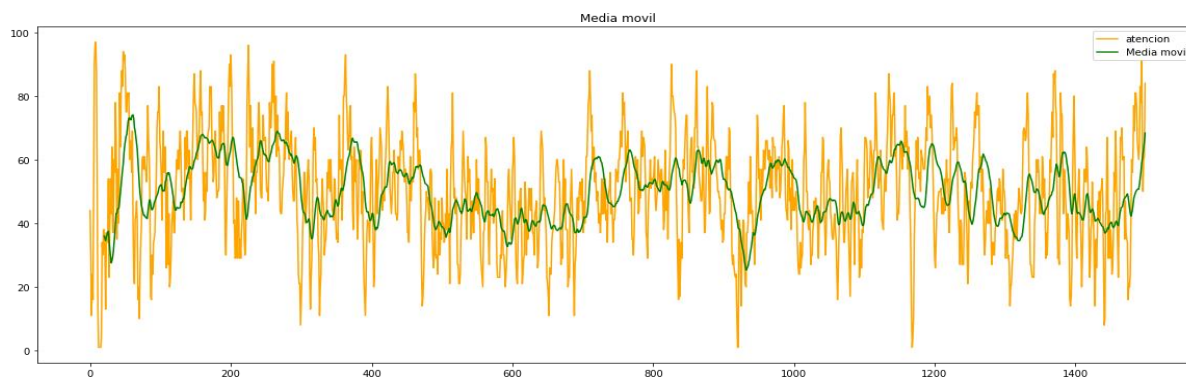
3.5.5.3 Filtro Media Movil

Se realiza el análisis de los parámetros tanto de la relación SRN que tiene que ser mayor a la obtenida de los datos originales como también el coeficiente de variación debe ser menor al de los datos sin filtrar para determinar si este filtro es adecuado para los datos obtenidos.

A continuación, se muestra en la Figura 56 las gráficas en donde se tiene la señal original versus la señal filtrada de las ondas para este caso los datos de atención y las ondas theta y alfa recalcando que el mismo análisis se lo realiza para todo el proceso en cada escenario, en este caso se realiza en el estado de vigila el análisis respectivo.

Figura 56

Datos atención filtrados por Media Móvil



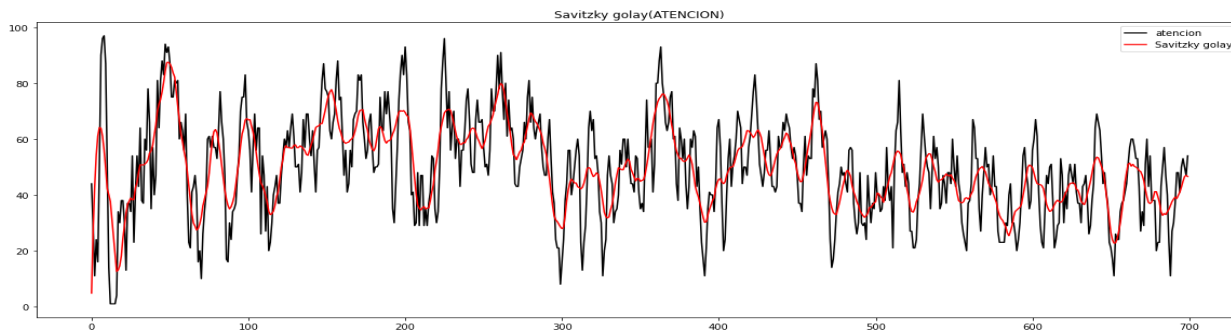
Fuente: Elaborado por el autor

3.5.5.1 Filtro Savitzky golay

El filtro Savitzky-Golay (SG) suaviza lecturas analógicas de sensores obtenidas mediante el converso análogo digital (DAC) como las que se muestra en la Figura 57 y 58. Este conjunto de datos en particular se suavizó con una longitud de ventana de 21 y con un polinomio de grado 2 ajustado por mínimos cuadrados lineales para 11000 muestras analizadas, con coeficientes enteros porque ocupan menos espacio de memoria.

Figura 57

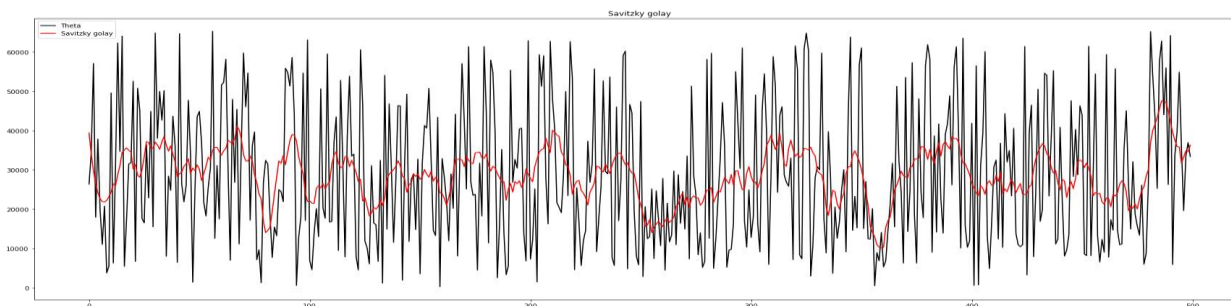
Filtro Savitzky Golay en atención



Fuente: Elaborado por el autor

Figura 58

Filtro Savitzky Golay en señal theta



Fuente: Elaborado por el autor

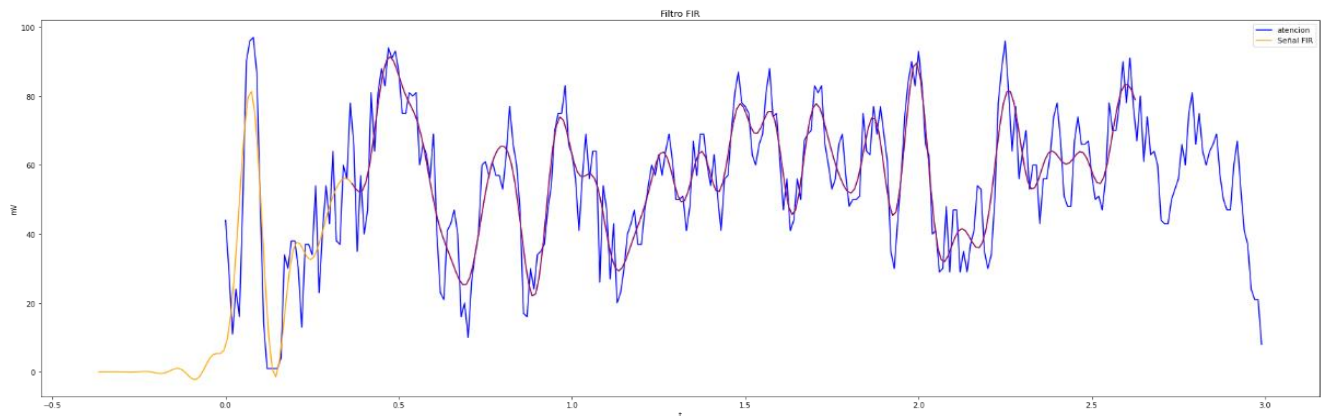
3.5.5.2 Filtro FIR

Se realiza el análisis de las señales EEG obtenidas por el sensor neurosky mindwave en el cual se aplica el FILTRO FIR ya que es un filtro digital que tiene un periodo de duración finita, estos filtros habitualmente suelen tener su fase exactamente de forma lineal además de ser filtros estables, son eficaces al implementar en hardware. Pero lo negativo de estos filtros es que a razón de que el orden de filtro sea mayor existe la posibilidad de que el retardo de esos sea mayor. (Mathworks.com, s.f.)

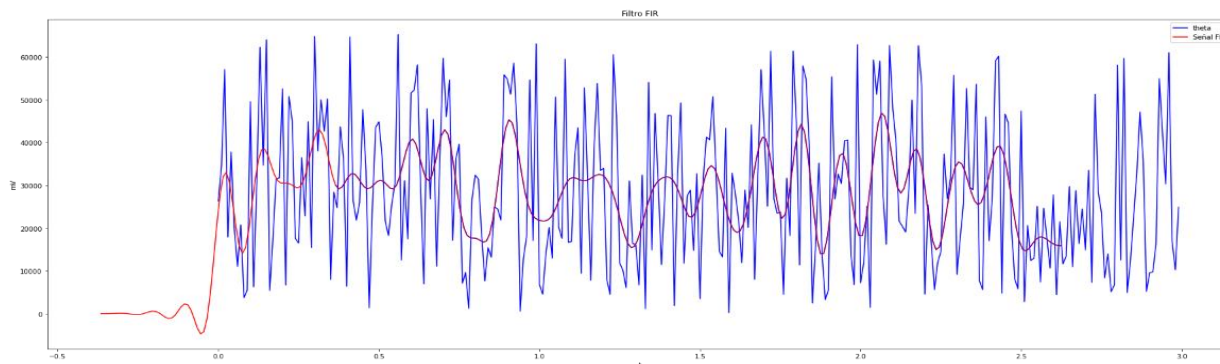
En la Figura 59 y Figura 60 se puede observar la aplicación del Filtro FIR a las señales obtenidas por el sensor.

Figura 59

Señal atención con filtro FIR



Fuente: Elaborado por el autor

Figura 60*Señal theta con filtro FIR**Fuente:* Elaborado por el autor**3.5.5.3 Selección de filtro**

En la Tabla 25, se muestra la relación que debe tener la señal filtrada con la señal original en donde si se marca de color verde indica que cumple si se marca de color rojo indica que no cumple.

Tabla 25*Relación de validación de filtro*

Relación	Cumple	No cumple
SRN señal filtrada es mayor a la SRN de la señal original.	valor	valor
Coefficiente de variación en señal filtrada es menor al Coeficiente de variación en señal original.	valor	valor

En la Tabla 26, se puede evidenciar los parámetros que se obtuvo de la señal original misma que será analizada con los parámetros que se obtendrán de cada uno de los filtros a realizar para así elegir el filtro que mejor de acople a los datos.

Tabla 26

Tabla comparativa señal original vs señal filtrada

VALORES SEÑAL ORIGINAL							
	Atención	Theta	Low alfa	High alfa	Low beta	Low gamma	Meed gamma
Media	52.936	28788.448	15323.536	13436.016	11556.862	4780.302	4209.138
Desviación típica (sdt)	18.8563	18111.05473	13952.37793	10798.82919	10167.60978	3737.440754	3024.569497
SNR (mean/sdt)	2.80734162	1.589551157	1.098274149	1.244210439	1.136635085	1.279030843	1.391648631
Coefficiente de variación (sdt/mean)	0.3562	0.6291	0.9105	0.8037	0.8798	0.7818	0.7186
VALORES SEÑAL FILTRADA MEDIA MOVIL							
	Atención	Theta	Low alfa	High alfa	Low beta	Low gamma	Meed gamma
Media	45.305303	27378.37326	14858.59955	14148.18257	11081.16685	6191.716396	4873.598245
Desviación típica (sdt)	12.800821	4476.874979	3487.58619	3007.83962	3064.185079	3995.605921	1581.963619
SNR (mean/sdt)	3.539249787	6.115509901	4.260425045	4.703768936	3.616350371	1.5496314	3.080727133
Coefficiente de	0.2825	0.1635	0.2347	0.2126	0.2765	0.6453	0.3246

variación
(sdt/mean)

Resultados: Dentro de la aplicación de este filtro se puede denotar que los valores SNR en la señal original son menores a la señal filtrada por lo cual cumple y a su vez el coeficiente de variación es menor en la señal filtrada por lo cual el uso de este filtro si es accesible, pero hay que tener en cuenta que en el uso del filtro se pierde una porción de los datos iniciales como se observa en la Figura 56 en este caso no es viable el uso de este filtro.

VALORES SEÑAL FILTRADA SAVITZKY GOLAY

	Atención	Theta	Low alfa	High alfa	Low beta	Low gamma	Meed gamma
Media	45.315624	27385.00046	14848.73296	14145.16425	11081.09224	6188.325396	4872.086038
Desviación típica (sdt)	14.903565	6141.665474	4869.786264	4353.148842	3967.968423	4323.838945	1964.13344
SNR (mean/sdt)	3.04058955	4.458888322	3.049154963	3.249409741	2.792636196	1.431210893	2.480527004
Coefficiente de variación (sdt/mean)	0.3289	0.2243	0.3280	0.3077	0.3581	0.6987	0.4031

Resultados: Los resultados obtenidos para este filtro cumplen con los parámetros que la SRN debe ser mayor con respecto a los obtenidos en la señal original, mientras que el coeficiente de variación de la señal filtrada debe ser menor a los de la señal original, además se puede visualizar en la Figura 58 que los datos no se pierden mientras se usa este filtro por lo cual el filtro es elegible para su aplicabilidad en los datos.

VALORES SEÑAL FILTRADA FILTRO FIR

	Atención	Theta	Low alfa	High alfa	Low beta	Low gamma	Meed gamma
Media	44.946669	27148.38888	14757.23366	14024.72704	10993.58021	6131.315961	4818.244903
Desviación típica (sdt)	17.009557	8244.868456	6338.359576	5564.021918	5010.468271	4637.571906	2392.215745

SNR							
(mean/sdt)	2.64243619	3.292761919	2.328241793	2.520609596	2.194122308	1.32209615	2.014134767
Coeficiente							
de							
variación	0.3784	0.3037	0.4295	0.3967	0.4558	0.7564	0.4965
(sdt/mean)							

Resultados: Los resultados obtenidos para este filtro para los datos de atención no cumple con los requisitos que la SRN sea mayor a la señal original como que el coeficiente de variación sea menor que el de la señal original. Como también se pudo observar en la Figura 59 que los datos tienen su inicio desde un valor negativo el cual para el análisis de los datos no sea factible.

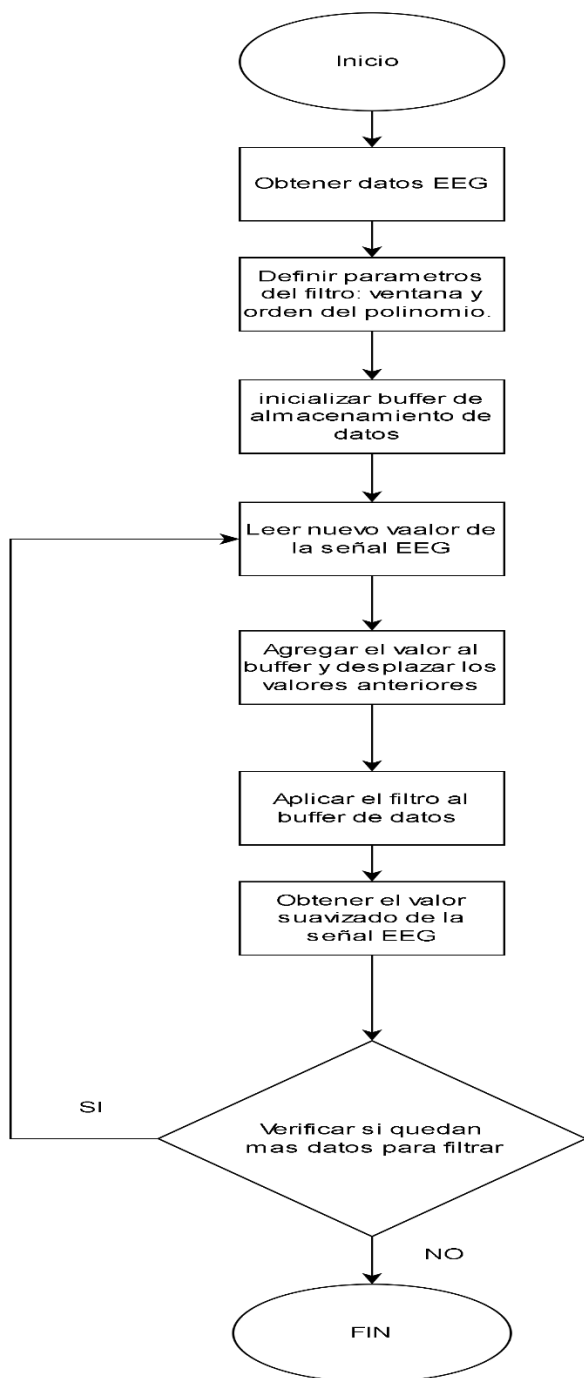
SELECCIÓN DE FILTRO A UTILIZAR

Elección: Una vez realizado el previo análisis de los resultados obtenidos de cada filtro se determina que se el filtro ideal para este proyecto será el de **Savitzky Golay** debido a que cumple con los parámetros necesarios brindando una buena SNR como también la disminución del coeficiente de variación.

El proceso del filtrado de los datos es fundamental para el desarrollo del proyecto en el cual se debe seguir una secuencia de acciones que permiten el suavizado de las señales EEG para permitir obtener mejores valores en la señal. En la Figura 61, se observa el diagrama de flujo del funcionamiento del proceso del filtrado de la señal utilizando el filtro Savitzky Golay.

Figura 61

Diagrama de bloques de filtro digital



Fuente: Elaborado por el autor

3.5.5.4 Minería de datos

Para la determinación del algoritmo de clasificación en una primera instancia se procede a realizar mediante el uso de Google Colab ya mediante este software de codificación permite el uso de diversas librerías las mismas que permiten el análisis de la base de datos ya obtenida para tener resultados preliminares para la elección del algoritmo de clasificación a implementar en el sistema embebido.

Por tal motivo se realiza el uso de la librería LazyPredict la cual es un paquete de Python el cual permite conocer de manera automática cuales son los mejores modelos de clasificación de acuerdo con los datos que nos encontremos analizando. Por lo cual como primer paso se procede con la instalación de las librerías de este paquete de Python como se muestra en la Figura 62 la instalación correcta de cada librería necesaria para el funcionamiento de LazyPredict

Figura 62

Instalación de LazyPredict

Installation of libraries

LazyPredict

Library to run many sklearn's binary classifiers with default parameters

```
[ ] !pip install lazypredict
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: lazypredict in /usr/local/lib/python3.8/dist-packages (0.2.12)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages (from lazypredict) (4.64.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.8/dist-packages (from lazypredict) (1.0.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.8/dist-packages (from lazypredict) (1.2.0)
Requirement already satisfied: lightgbm in /usr/local/lib/python3.8/dist-packages (from lazypredict) (2.2.3)
Requirement already satisfied: xgboost in /usr/local/lib/python3.8/dist-packages (from lazypredict) (0.90)
Requirement already satisfied: pandas in /usr/local/lib/python3.8/dist-packages (from lazypredict) (1.3.5)
Requirement already satisfied: click in /usr/local/lib/python3.8/dist-packages (from lazypredict) (7.1.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from lightgbm->lazypredict) (1.21.6)
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from lightgbm->lazypredict) (1.7.3)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/dist-packages (from pandas->lazypredict) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-packages (from pandas->lazypredict) (2022.7.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->lazypredict) (3.1.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.7.3->pandas->lazypredict) (1.15.0)
```

Fuente: (Readthedocs.io, 2023)

Una vez realizado la instalación de los paquetes de LazyPredict se procede con la etapa del preprocesamiento de los datos el cual se realiza todas las etapas necesarias para obtener los datos en los algoritmos de clasificación como lo es desde la importación de la base de datos y

manipulación de esta información como también el uso de los diversos algoritmos de clasificación para su respectiva respuesta y exactitud al procesar los datos como se puede observar en la Figura 63.

Figura 63

Importación de algoritmos de Machine Learning

```

# Download and unzip
from urllib.request import urlopen
from io import BytesIO
from zipfile import ZipFile

# Manipulación de datos
import pandas as pd
import numpy as np

# Plots
import matplotlib.pyplot as plt
import seaborn as sns

# Machine learning classifiers
import lazypredict
import pickle
from lazypredict.Supervised import LazyClassifier
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import GridSearchCV
from lightgbm import LGBMClassifier
from xgboost import XGBClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score, accuracy_score, precision_score, recall_score, f1_score, matthews
from sklearn.metrics import make_scorer
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix
from matplotlib_venn import venn3
from sklearn.metrics import RocCurveDisplay
from itertools import cycle
from sklearn.multiclass import OneVsRestClassifier
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import label_binarize

# Google colab
from google.colab import files
import io

```

Fuente: Elaborado por el autor

Debido que el software lee datos de manera digital se procede al etiquetamiento de los datos en el cual en la base de datos se tenía determinado como etiquetas tanto al estado de vigila como también al estado somnoliento estas etiquetas se proceden a darles un valor digital como se muestra en la Tabla 27.

Tabla 27

Valor digital de etiquetas

Etiqueta	Valor Digital
Vigila	0
Somnolencia	1

Fuente: Elaborado por el autor

A continuación, se procede a realizar el proceso de determinación de etiquetas en un modo digital dando un valor de 1 a somnolencia y un valor de 0 a vigila como se muestra en la Figura 64.

Figura 64

Determinación de etiquetas

```
# Determinacion de etiquetas
y = df['etiqueta'].copy()
y = y.map({"somnolencia": 1, "vigila": 0}).to_numpy()
```

Fuente: Elaborado por el autor

Mediante el uso de la librería LazyPredict se procede a determinar las variables tanto que servirán de entrenamiento como de prueba de la BDD en este caso se las toma de manera random como se muestra en la Figura 65.

Figura 65

Variables de entrenamiento y prueba para ML

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                random_state =42, stratify=y)
```

Fuente: Elaborado por el autor

Ahora bien, se procede a la ejecución del programa el cual dará información del nivel de exactitud de los algoritmos de machine learning que forman parte de la librería LazyPredict para elegir cual es la mejor opción para implementar como se muestra en la Figura 66, en donde el algoritmo que brinda mayor exactitud por lo cual permite a la aplicabilidad de este proyecto es con Random Forest ya que tiene una exactitud del 94%.

Figura 66

Exactitud de algoritmos de ML.

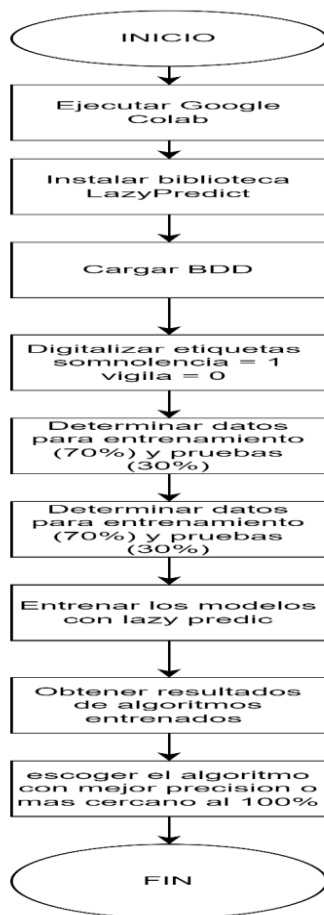
Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	matthews_corrcoef	Time Taken
RandomForestClassifier	0.94	0.93	0.93	0.94	0.87	1.62
BaggingClassifier	0.88	0.89	0.89	0.88	0.76	0.48
XGBClassifier	0.89	0.88	0.88	0.89	0.77	1.62
LabelSpreading	0.86	0.88	0.88	0.86	0.73	1.40
LabelPropagation	0.86	0.87	0.87	0.86	0.73	0.81
ExtraTreeClassifier	0.84	0.85	0.85	0.84	0.68	0.02
DecisionTreeClassifier	0.83	0.85	0.85	0.83	0.68	0.09
LGBMClassifier	0.84	0.81	0.81	0.83	0.64	0.19
NuSVC	0.77	0.73	0.73	0.76	0.49	1.79

Fuente: Elaborado por el autor

En la Figura 67, se muestra el proceso que se realiza para realizar la de minería de los datos en donde se lleva a cabo mediante el uso de la librería Lazy Predict en el software Google Colab, permitiendo el análisis de una variedad de algoritmos de clasificación y regresión para realizar la mejor elección del algoritmo que tenga mejor precisión.

Figura 67

Proceso de algoritmo de clasificación con Lazy Predict



Fuente: Elaborado por el autor

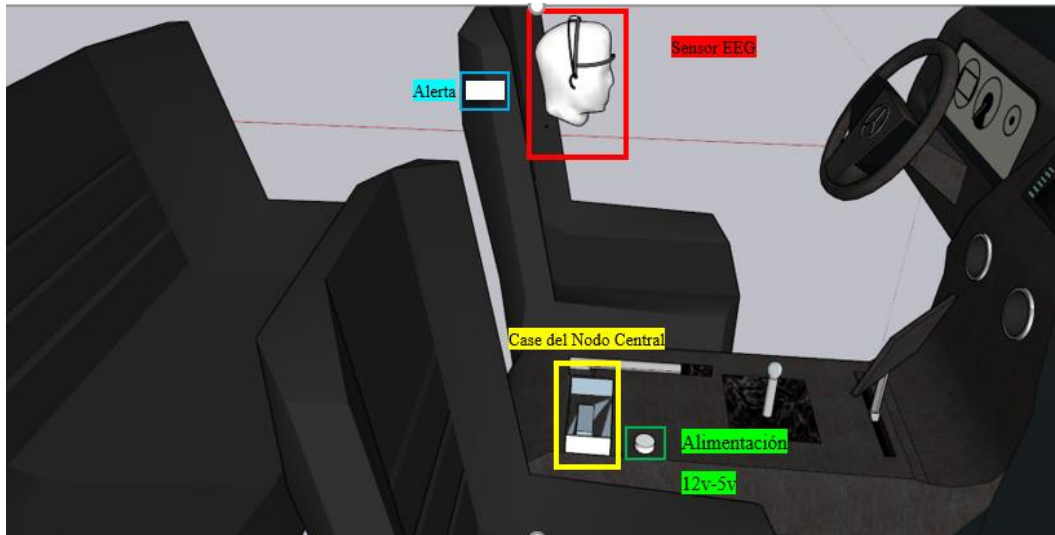
3.6 Fase 4: Integración y testeo del sistema

3.6.1 Bloque 6: Implementación

En este bloque se realiza la implementación completa del sistema el cual consta de dos partes que son implementación de hardware y software como se muestra en la Figura 68 en dónde se indica tanto el posicionamiento de cada dispositivo que integra tanto software como hardware del sistema.

Figura 68

Implementación de prototipo



Fuente: Elaborado por el autor

3.6.1.1 Implementación de hardware:

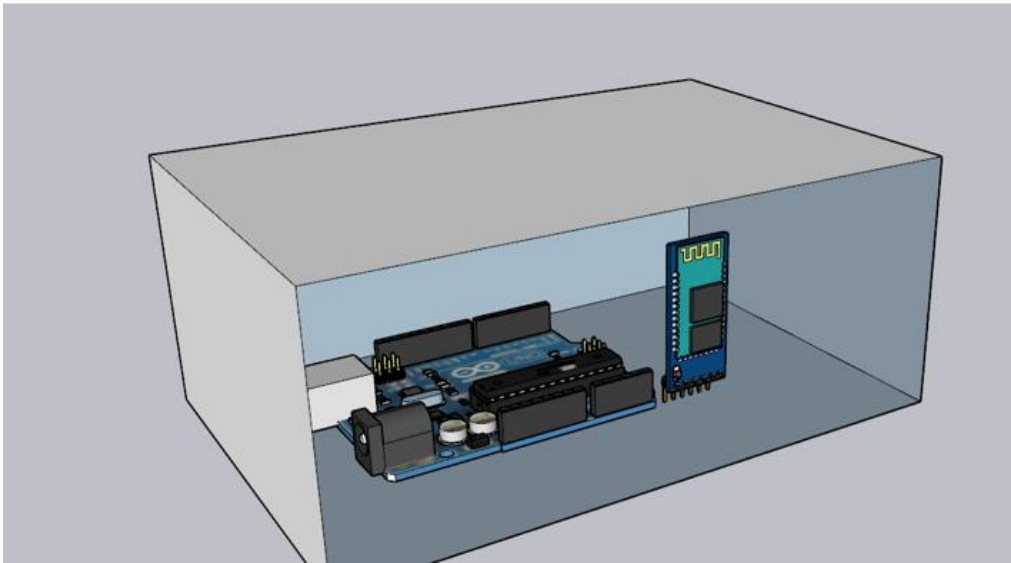
3.6.1.1.1 Case nodo central

En este apartado se realiza el diseño del case para el nodo central se posicionará tanto el microprocesador el cual es el encargado de realizar todo el proceso con los datos como también el módulo hc-05 el cual es el encargado de la comunicación inalámbrica con el sensor EEG como se muestra en la Figura 69, el case diseñado para el nodo central es de la siguiente dimensión 10

cm de ancho por 15 de largo y con una altura de 12 cm siendo un case fácil de manipular e implementar en la cabina del automóvil sin que genere algún tipo de obstrucción que entorpezca la actividad cotidiana del conductor.

Figura 69

Diseño case nodo central

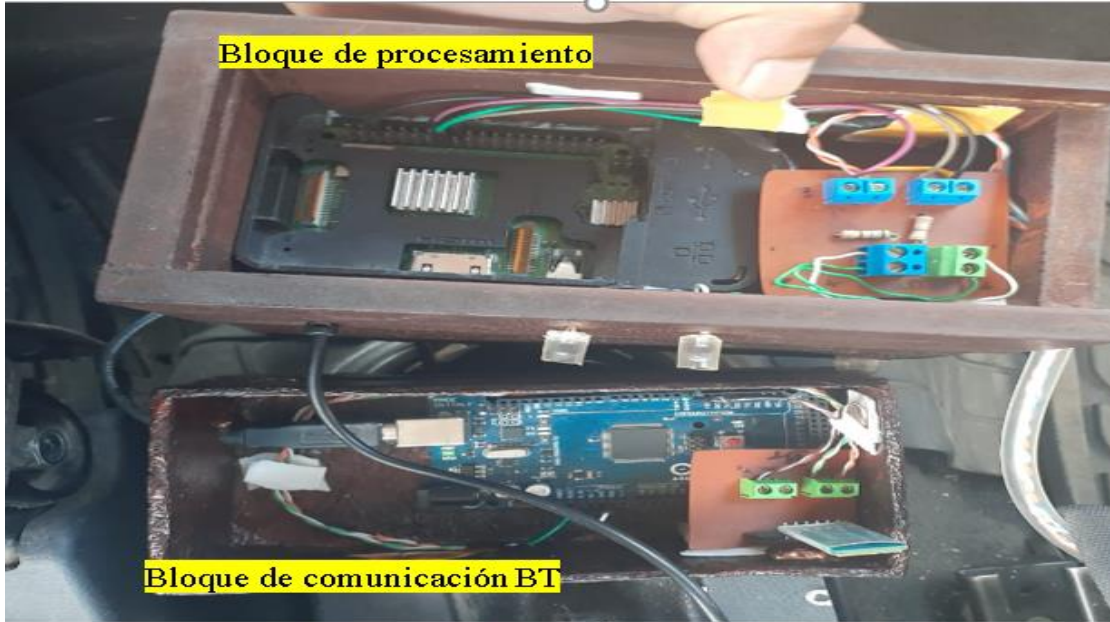


Nota: Case del prototipo de tamaño reducido en donde se implementa

En la Figura 70, se puede observar la elaboración del case del nodo central en donde se tiene acoplado los elementos tanto de comunicación inalámbrica entre el sensor EEG y también el microprocesador el cual es el encargado de realizar el tratamiento de los datos para la elección del estado del usuario.

Figura 70

Elaboración e implementación case nodo central, parte inferior bloque de comunicación, parte superior bloque de procesamiento de datos.



Nota: Case del prototipo de tamaño reducido en donde se implementa los dispositivos que realizan tanto la comunicación inalámbrica y el procesamiento de los datos.

En la Figura 71, se puede observar el case completo implementado a la cabina del automóvil en donde se tiene conectado cada bloque como los es el de alimentación, comunicación, procesamientos de datos y ejecución de acciones.

Figura 71

Case nodo principal completo, en la parte inferior se observa la fuente de alimentación y la parte superior el case del prototipo.



Fuente: Elaborado por el autor

La aleta que será emitida al conductor será de manera auditiva misma que se encuentra alimentada por un pulso eléctrico cuando el conductor presente un estado inadecuado en su actividad EEG, esta actividad es procesada por la unidad de procesamiento que es la placa Raspberry pi, al momento que los datos EEG se encuentren dentro del rango de los establecidos para la etiqueta 1 perteneciente a somnolencia que es el detonante que permite la activación de la alerta. La ubicación para la alerta se determinó que será detrás de la cabeza del conductor en donde se encuentra el apoyo cabeza del conductor, el lugar es el idóneo ya que al ser una alerta sonora se encuentra muy cerca de los oídos, como se muestra en la Figura 72 la ubicación estratégica de la alerta electrónica.

Figura 72*Ubicación de alerta*

Fuente: Elaborado por el autor

3.6.1.2 Implementación de software:

Una vez que se realiza la elección del modelo de clasificación, en este caso se lo dedujo por el nivel de exactitud que se muestra en la Figura 66 que es el de Random Forest con un porcentaje de acierto de 94%, se procede a realizar la implementación del algoritmo para detectar el estado de somnolencia en el usuario, dicho proceso será hecho mediante el software de Google Colab en donde se realiza el entrenamiento del y su posterior exportación la cual se la implementara en la placa Raspberry pi .

A continuación, se muestra el proceso realizado para la programación del algoritmo Random Forest para su implementación, entrenamiento y pruebas de predicción con los datos y posterior detección del estado de somnolencia.

En la Figura 73, se muestra las librerías y dependencias necesarias para la realización correcto del algoritmo de clasificación que es Random Forest.

Figura 73

Instalación de librerías

```
ALGORITMO RANDOM FOREST PARA SOMNOLENCIA
SOMNOLENCIA 1 VIGILA 0
atencion theta low alpha high alpha low betha high beta low gamm medd gamma etiqueta

import pandas as pd
import numpy as np
import math as ma
from matplotlib import pyplot as pyplot
from sklearn.tree import plot_tree
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

Nota. Importante instalar e importar las librerías que sean necesarias para realizar el algoritmo de clasificación, éstas facilitan la realización del programa.

En la Figura 74, se observa la realización de separación de columnas de la base de datos; para la variable (X) se determina los datos obtenidos por el sensor neurosky (atención, theta, low alfa, high alfa, low beta, high beta, low gamma, mid gamma), mientras que en la variable (y) se determina a las etiquetas para los datos (1= somnolencia, 0= vigila).

Figura 74

Determinación de variables x para columnas y variable y para etiquetas.

```
[18] X= df.iloc[:, :-1]
      y=df.iloc[:, -1]
      x.head()
```

	Attention	theta	low alfa	high alfa	low beta	high beta	low gamma	mid gamma
	0	1	2	3	4	5	6	7
0	55.048055	35476.30533	13696.12063	10444.592350	10743.448190	7860.708075	6686.600850	5257.460935
1	53.386074	34761.93364	12872.80222	8348.245505	7835.438705	6198.498856	4709.672769	4514.000000
2	52.091206	34725.81661	14417.72965	10646.404710	5575.319712	5374.288656	4889.281465	4733.869565
3	52.170644	35338.57077	14565.20497	11390.213140	3569.378882	5738.254658	5078.126185	5483.370709
4	53.060150	33558.34946	13039.00458	12004.969600	2322.362864	5670.543642	5233.664596	5501.399477


```
y
```

0	1
1	1
2	1
3	1
4	1
..	..
10634	0
10635	0
10636	0
10637	0
10638	0

ETIQUETAS
1=somnolencia
0=vigila

Name: 8, Length: 10639, dtype: int64

Nota. Determinación de variables para separar los datos EEG y la etiqueta.

Se realiza la visualización del número total de etiquetas existentes en la BDD para cada estado a realizar el análisis, por tal motivo es importante que la base de datos se encuentre balanceada con un aproximado de datos iguales para cada estado como se muestra en la Figura 75, en donde se tiene la cantidad de valores similares para cada etiqueta.

Figura 75

Verificación de cuantos valores existen para cada etiqueta

```
✓ 0s ceros= y[y==0].count()
unos= y[y==1].count()
```

```
[ ] ceros ,unos
```

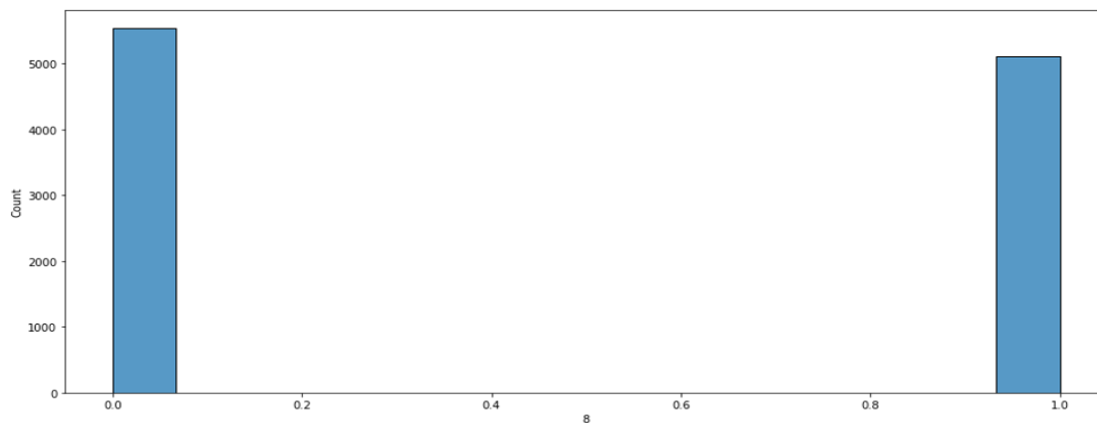
```
(5529, 5110)
```

Fuente: Elaborado por el autor

En la Figura 76, se visualiza de manera grafica la cantidad de datos para cada estado en su etiqueta, mostrando así que la BDD esta balanceada con datos para los dos estados a tratar.

Figura 76

BDD balanceada para etiqueta de somnolencia (1) y vigila (0).



Nota. Validación grafica del total de datos para cada etiqueta en la BDD para los 2 estados.

Se procede a determinar las variables las cuales seran tanto para el proceso de entrenamiento como para el proceso de prueba del algoritmo, por lo cual se lo divide en el 80% de los datos totales seran utilizados para el entrenamiento mientras que el 20% restante de los datos seran para realizar las pruebas respectivas de verificacion del algoritmo, en la Figura 77 se puede observar la separación de los datos.

Figura 77

Determinación de datos para entrenamientos como para prueba

separar conjunto de entrenamiento y datos de test

```

✓ ▶ from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test= train_test_split(X,y,test_size=0.2, random_state=1)
X_train.head()

```

Nota. Determinación de variables de entrenamiento y prueba del algoritmo.

En la Figura 78, se puede observar la cantidad de datos que son divididos para tanto el proceso de entrenamiento y el proceso de prueba.

Figura 78

Número de datos de entrenamiento y prueba

```

▶ print('# muestras en y ', np.bincount(y))
print('# muestras en y_train ', np.bincount(y_train))
print('# muestras en y_test ', np.bincount(y_test))

# muestras en y [5529 5110]
# muestras en y_train [4444 4067]
# muestras en y_test [1085 1043]

```

Nota. Datos para entrenamiento y prueba con el 80% y 20%.

Una vez realizado la separación de datos tanto para pruebas como para entrenamiento se procede a crear el algoritmo Random Forest, agregando los parámetros necesarios como lo es la cantidad estimada de bosques, la profundidad bosque para así obtener los mejores valores posibles de acierto de este algoritmo como se muestra en la Figura 79.

Figura 79

Creación de algoritmo de clasificación Random Forest

```
[ ] from sklearn.ensemble import RandomForestClassifier

[ ] rf= RandomForestClassifier(n_estimators=500,
                             n_jobs=-1,
                             criterion='gini',
                             max_features='sqrt',
                             max_depth=10,
                             random_state=1)
```

Nota. Creación de modelo Random Forest con todos sus parámetros más importantes como la cantidad de bosques, profundidad del algoritmo.

Se procede a realizar el proceso de entrenamiento del algoritmo mediante el uso de las variables creadas anteriormente para determinar los datos de entrenamiento como se muestra en la Figura 80.

Figura 80

Entrenamiento del algoritmo

```
▶ rf.fit(X_train,y_train)

↳ RandomForestClassifier
RandomForestClassifier(max_depth=10, n_estimators=500, n_jobs=-1,
                      random_state=1)
```

Nota. Proceso de entrenamiento del algoritmo Random Forest.

Una vez realizado el entrenamiento del algoritmo se procede a realizar una prueba de verificación, la cual trata de una comparación de los datos de entrenamiento que vienen a ser datos desconocidos para el algoritmo y así este envía su predicción para proceder a comparar las

etiquetas de la BDD de prueba con la clasificación del algoritmo como se puede visualizar en la Figura 81.

Figura 81

Comparación de predicciones dadas por el algoritmo entrenado

```
[ ] y_prds=rf.predict(X_test)
[ ] comp=pd.DataFrame({'real':y_test, 'prediccion':y_prds})
[ ] comp.head(10)
```

	real	prediccion
8976	0	0
9485	0	0
10169	0	0
4774	1	1
4908	1	1
4850	1	1
9798	0	0
7777	0	0
9743	0	0
2028	1	1

Nota. Prueba de predicción del sistema con datos de testeo.

Se calcula el nivel de exactitud que tiene el algoritmo para la clasificación de estados este proceso se lo calcula utilizando el parámetro **accuracy_store**, si el valor de exactitud mientras más cercano a 1 este es sistema es confiable, como se puede observar en la ;Error! No se encuentra el origen de la referencia. el sistema tiene una exactitud del 0.94.

Figura 82

Verificar nivel de exactitud del algoritmo

```
[88] acc=accuracy_score(y_test,y_prds)
      acc
      0.9421992481203008
```

```
score=rf.score(X_test,y_test)
print(score)
0.9421992481203008
```

Nota. Valor de precisión del algoritmo.

En la Figura 83, se puede observar las métricas del algoritmo en donde indica valores de Recall el cual el indicador del porcentaje de falsos positivos que emite el algoritmo y también el valor de f1-score el cual permite medir la precisión y exhaustividad que tiene el modelo, en este caso si los valores de estas métricas son más altas o más cercanas al 1 quiere decir que el modelo es preciso.

Figura 83

Identificación de métricas del algoritmo

METRICAS

```

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

```

	precision	recall	f1-score	support
0	0.98	0.91	0.95	1164
1	0.90	0.98	0.94	964
accuracy			0.94	2128
macro avg	0.94	0.95	0.94	2128
weighted avg	0.95	0.94	0.94	2128

Nota. Métricas calculadas para el algoritmo de clasificación.

En la Figura 84, se puede observar la matriz de confusión en donde se puede realizar el análisis del modelo de clasificación en su desempeño, la matriz muestra el número de verdaderos positivos, verdaderos negativos marcados de verde, falsos positivos y falsos negativos marcados de rojo.

Figura 84

Matriz de confusión

```

[90] cm = confusion_matrix(y_test, y_prds)
cm

```

array([[1063, 22], [101, 942]])

Fuente: Elaborado por el autor

En la Figura 85, se realiza la codificación necesaria para la programación de la activación de la alarma en este caso si la etiqueta se encuentra con un valor de 1 la alarma se encenderá.

Figura 85

Programación de acciones

```
# Ejecutar accion dependiendo su estado

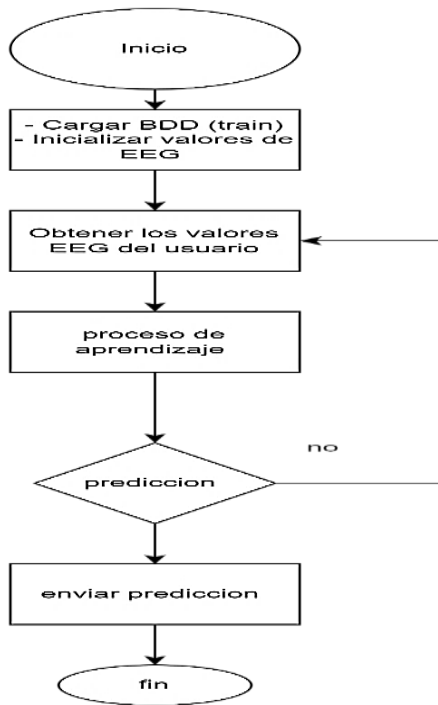
if (
  prediction == 0
): # si el usuario esta en estado despierto se encendera el pin 18 mientras que los pines 15 y 16 estaran off
  print("DESPIERTO")
  GPIO.output(18, GPIO.HIGH)
  GPIO.output(15, GPIO.LOW)
  GPIO.output(16, GPIO.LOW)
else:
  GPIO.output(18, GPIO.LOW)
  if (
    prediction == 1
  ): # si el usuario esta en estado de somnolencia se encenderan los pines 15 (Alerta electronica) y 16 de alarma mientras que el pin 18 modo off
    print("ALERTA... ")
    GPIO.output(15, GPIO.HIGH)
    GPIO.output(16, GPIO.HIGH)
    GPIO.output(18, GPIO.LOW)
  else:
    GPIO.output(15, GPIO.LOW)
    GPIO.output(16, GPIO.LOW)
```

Nota. Programación para activación de alarma si se encuentra en estado de somnolencia el usuario.

El funcionamiento del software general para el prototipo el cual usa el algoritmo de clasificación en específico Random Forest lleva a cabo un proceso de entrenamiento con bases de datos previamente adquiridas en los estados de somnolencia y vigila digitalizando estos datos con una etiqueta de 1 para somnolencia y una etiqueta de 0 para vigila para que así se realice la predicción debido del estado del usuario dependiendo de los datos que emita el algoritmo determinará si este está en somnolencia o despierto sí el usuario se encuentra en estado de somnolencia se procede a realizar la activación de las acciones en este caso mediante una alerta sonora que permitirá establecer una advertencia al conductor, en la Figura 86 se puede evidenciar mediante el diagrama de flujo el funcionamiento del software del prototipo de detección del estado de somnolencia mediante el análisis de ondas EEG.

Figura 86

Diagrama de flujo software de Random Forest



Fuente: Elaborado por el autor

3.6.2 Testeo del prototipo

En la Figura 87, se visualiza tanto la implementación como el primer testeo del prototipo para llevar a cabo el proceso de prueba, en donde se encuentra implementado tanto software como hardware de manera simultánea.

Figura 87

Implementación prototipo en ambiente real



Fuente: Elaborado por el autor

CAPITULO IV: PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS

En este apartado se presenta las pruebas de funcionamientos del prototipo de detección del estado de somnolencia en conductores. Dichas pruebas son la parte final de la metodología iterativa, por lo cual se realiza la validación tanto de requerimientos, requisitos y especificaciones que son establecidas en el diseño del sistema para su funcionamiento implementado. Por tal motivo con esta etapa se da por concluido cada una de las fases del proceso iterativo para así presentar la información de una manera ordenada.

4.1 Delimitación del proceso de pruebas a ejecutar

La ejecución de las distintas pruebas a realizar, se las desarrolla para cada etapa del modelo iterativo, a continuación, en la Tabla 28 se puede observar el proceso en cada fase y los resultados que se estima obtener como también el documentar los mismos en este capítulo.

Tabla 28

Pruebas de funcionamiento

PRUEBAS DE FUNCIONAMIENTO		
Tipo de prueba	Descripción	Resultado estimado
Fase 1. Cumplimiento de requerimientos	En esta fase corresponde a la verificación del cumplimiento de requerimientos necesarios para la elaboración del prototipo electrónico mediante una lista de verificación.	Verificar mediante una lista de verificación del cumplimiento de los requerimientos de Stakeholders más importantes para el desarrollo del proyecto.
Fase 2. Funcionalidad en cada bloque	La finalidad de esta fase es la verificación de la funcionalidad en cada bloque propuesto en el desarrollo del sistema.	Comprobar los requerimientos del sistema sean cumplidos en mayoría.

Fase 3. Implementación	Las pruebas realizadas en el ambiente de campo del usuario.	Se estima comprobar la detección del estado de somnolencia y alerta respectiva.
------------------------	---	---

Fuente: Elaborado por el autor

4.2 Pruebas de cumplimiento de requerimientos: stakeholders, arquitectura, sistema

Para la realización de la verificación de los requerimientos en cada una de las instancias se la procederá a evaluar mediante un análisis del proyecto realizado con cada requerimiento y validando en la tabla si el requerimiento es cumplido y en caso de que no se cumpla el requerimiento se debe detallar el motivo técnico por el cual no se cumple dicho parámetro.

Tabla 29

validación de requerimientos

REQUERIMIENTO	CUMPLE	NO CUMPLE
# REQUERIMIENTO	✓	
# REQUERIMIENTO		X

Fuente: Elaborado por el autor

4.2.1 Cumplimiento de Stakeholders

Para el desarrollo de estas pruebas se recopila la información con respecto a los requerimientos de Stakeholders para verificar su existencia en el desarrollo del proyecto estableciendo una lista de los más importantes que se cumplen para validar las especificaciones tanto operacionales como de usuario que se muestra en la Tabla 30.

Tabla 30*Validación de Stakeholders*

REQUERIMIENTOS DE STAKEHOLDERS													
	STSR 1	STSR 2	STSR 3	STSR 5	STSR 6	STSR 7	STSR 8	STSR 9	STSR 10	STSR 13	STSR 14	STSR 15	STSR 17
CUMPLE	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
NO CUMPLE									X				

Fuente: Elaborado por el autor

En los requisitos de Stakeholders se analiza tanto requerimientos operacionales como de usuario, por tal motivo existe un requisito que en su totalidad no se cumple, debido a que el dispositivo que capta los datos EEG tiene un valor considerablemente alto en el mercado haciendo más costosa la implementación del prototipo.

En la Tabla 31, se enlistan los requerimientos del sistema, en donde se toma en cuenta los requerimientos que aportan de manera significativa al prototipo.

Tabla 31*Validación de requerimientos del sistema*

REQUERIMIENTOS DEL SISTEMA									
	SYSR 2	SYSR 6	SYSR 7	SYSR 8	SYSR 10	SYSR 11	SYSR 14	SYSR 15	SYSR 16
CUMPLE	✓	✓	✓	✓	✓	✓	✓	✓	✓
NO CUMPLE									

Fuente: Elaborado por el autor

Para los requerimientos del sistema, hay que tener en cuenta que abarca tanto los requerimientos de uso, rendimiento, interfaces, modos & estados y por último los físicos del prototipo. Por tal motivo el análisis de validación de estos requerimientos se encuentra de manera correcta en la Tabla 31 donde se valida el cumplimiento de los más importantes para desarrollo del prototipo.

En la Tabla 32, se muestra los requerimientos de arquitectura, en donde se tomó los que son más importantes ya que brindan un aporte de manera significativo al prototipo.

Tabla 32

Validación de requerimientos de arquitectura

REQUERIMIENTOS DE ARQUITECTURA													
	SRSH 1	SRSH 2	SRSH 5	SRSH 7	SRSH 9	SRSH 11	SRSH 12	SRSH 13	SRSH 18	SRSH 19	SRSH 21	SRSH 22	SRSH 23
CUMPLE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NO CUMPLE													

Fuente: Elaborado por el autor

En los requerimientos de arquitectura los cuales son establecidos y cumplidos por el desarrollador del proyecto ya que son requerimientos de Software, Hardware, Lógicos, Diseño y Eléctricos, por lo cual se logra cumplir de manera correcta dichos parámetros. Por lo cual el desarrollo del prototipo se lo llevo de manera correcta, cumpliendo cada requerimiento que aporta de manera más significativa en el correcto funcionamiento del prototipo.

4.3 Pruebas de funcionalidad por cada bloque

Para la realización de las pruebas de funcionalidad que componen al sistema se tiene que validar el correcto funcionamiento en cada bloque con la implementación tanto de hardware y software.

4.3.1 Bloque 1: Alimentación

Debido a que los automóviles tienen su batería la cual alimenta a todo el sistema eléctrico con un voltaje de 12 Vcc dicho voltaje es muy alto para el funcionamiento del prototipo por lo cual es necesario un conversor a 5Vcc para que funcione el prototipo tanto en el microcontrolador como también el módulo hc-05, este parámetro se encuentra documentado en la parte de requerimientos. Por tal motivo la ubicación del conversor puede ser directamente a la batería del automóvil o se lo puede conectar al tablero ya que se encuentra conectado al sistema eléctrico del automóvil.

Se procede a realizar un test eléctrico el cual tiene la finalidad de comprobación del correcto funcionamiento del conversor de 5Vcc para el módulo Arduino mega y el módulo de comunicación bluetooth hc-05, en donde se realiza la verificación mediante la medición de voltaje, creando una línea de voltaje de la entrada de alimentación del prototipo en un periodo de 3 horas, y verificar si existen posibles caídas de tensión, en la Tabla 33 se muestra el test a realizar.

Tabla 33

Test del bloque de alimentación

TEST ELÉCTRICO	
BLOQUE DE PRUEBA	Bloque de alimentación

DESCRIPCIÓN

Validación del funcionamiento del bloque de alimentación.

PRERREQUISITOS:

- ✓ Adquisición de dispositivo conversor de 12 Vcc a 5 Vcc
- ✓ Adquisición de módulo Arduino mega
- ✓ Adquisición de módulo bluetooth hc-05
- ✓ Adquisición de módulo Buzzer

PASOS:

- ✓ Verificar el voltaje de entrada al Arduino mega
- ✓ Verificar continuidad en la placa con el multímetro.
- ✓ Verificar entrada al módulo BT Hc-05
- ✓ Crear una línea de tiempo del voltaje suministrado

RESULTADOS ESTIMADOS:

Bloque de alimentación realiza la administración de voltaje al Arduino mega y el módulo hc-05 con un voltaje de 3.3Vcc a 5Vcc.

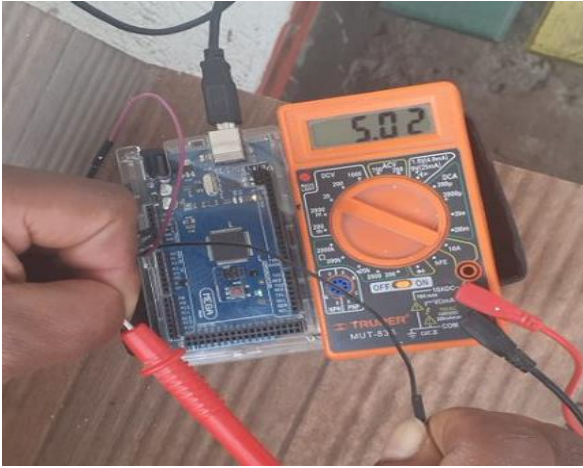
Fuente: Elaborado por el autor

Resultado obtenido: Después de realizar las diferentes mediciones de alimentación de los componentes del prototipo, se obtiene que el conversor entrega un voltaje de 5 Vcc es idóneo para el funcionamiento del microcontrolador Arduino mega el cual es el nodo principal en el proyecto, como también se verifica el voltaje para el módulo de comunicación bluetooth que va en rangos desde 3.3 v hasta un valor menor a 5v los cuales entra en el rango de trabajo para el correcto funcionamiento de este dispositivo.

A continuación, en la Figura 88, se muestra la prueba de medición del voltaje de entrada suministrado para el módulo de Arduino Mega el cual es el encargado de la adquisición de los datos.

Figura 88

Medición de voltaje en Arduino mega

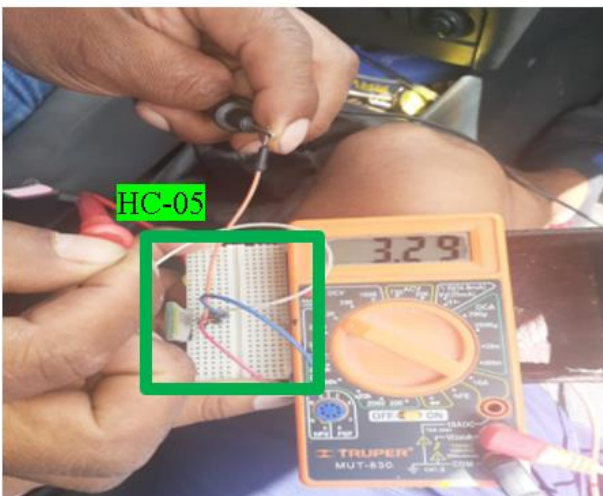


Fuente: Elaborado por el autor

En la Figura 89, se observa el voltaje de alimentación del módulo hc-05 el cual es de la comunicación inalámbrica entre el sensor EEG y el Arduino.

Figura 89

Alimentación de modulo hc-05

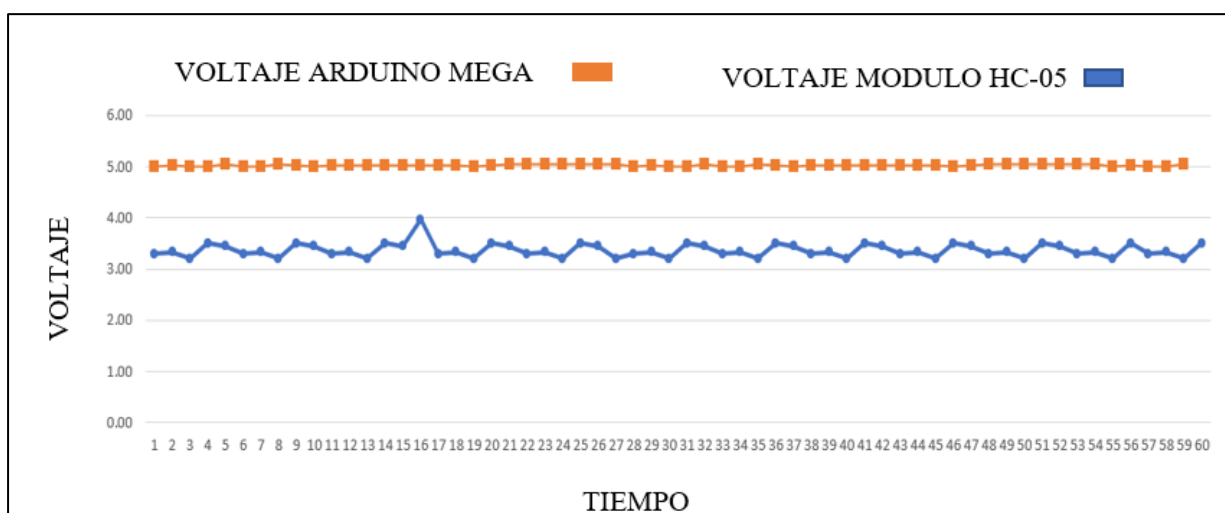


Fuente: Elaborado por el autor

En la Figura 90, se muestra el histograma del voltaje que fue suministrado a cada dispositivo del prototipo, tanto el dispositivo de procesamiento que funciona con un valor de 5Vcc y el de comunicación inalámbrica el cual funciona con un voltaje de 3.3 Vcc a 5Vcc.

Figura 90

Línea de tiempo de voltaje



Fuente: Elaborado por el autor

4.3.1.1 Tiempo de vida del prototipo

Para determinar el tiempo de vida útil del prototipo es importante verificar las hojas técnicas de cada dispositivo que forma parte del prototipo.

En el caso del prototipo el tiempo de vida estimado es de 7 a 10 años con un uso normal y buena manipulación de dispositivos, pero hay que tener en consideración cuales son las condiciones adecuadas en las que deben trabajar cada uno estos dispositivos, como lo es la temperatura que no afecte a su funcionamiento, otro de los aspectos a tener en cuenta es también

el polvo generado externamente que puede llegar a los dispositivos electrónicos ya que merman en su correcto desempeño.

También hay que tener en consideración al ser un prototipo implementado en un automóvil es indispensable que la batería de este se encuentre en buen estado, ya que si la batería no se encuentra en un estado óptimo los niveles de voltaje pueden variar y así alimentar a los dispositivos del prototipo con voltajes diferentes los cuales pueden perjudicar o dañar el funcionamiento de estos.

4.3.2 Bloque 2: captación EEG

En el proceso de captación es muy importante ya que en este bloque se familiariza al usuario con el dispositivo y también se da a conocer cada uno de los elementos del sensor y como es su correcta colocación, para lo cual se realiza un test de uso del sensor neurosky mindwave mobile 2 de cómo funciona como se muestra en la Tabla 34.

Tabla 34

Test bloque captación EEG

TEST DE CAPTACION EEG	
BLOQUE DE PRUEBA	Bloque de captación EEG
DESCRIPCIÓN	Validación del uso para el correcto funcionamiento del dispositivo neurosky mindwave mobile 2.
PRERREQUISITOS:	
<ul style="list-style-type: none"> ✓ Adquisición de dispositivo neurosky mindwave mobile 2. ✓ Adquisición de pilas AAA 	

PASOS:

- ✓ Insertar pilas AAA en el neurosky mindwave.
- ✓ Verificar estados que emite el sensor mediante el led indicador según la carga de la pila.
- ✓ Verificar correcta colocación del sensor como de la pinza de tierra

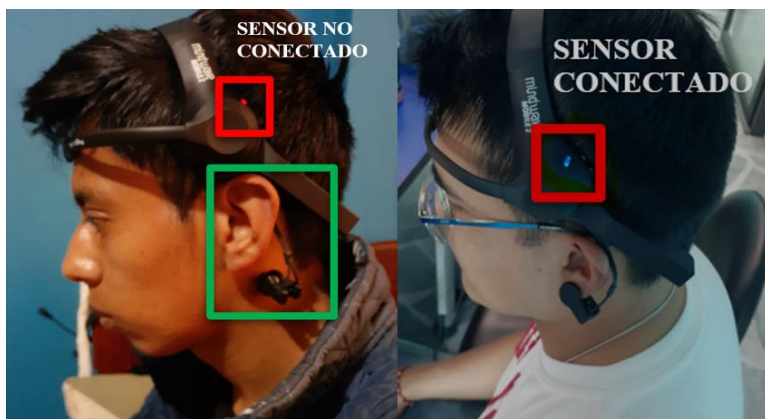
RESULTADOS ESTIMADOS:

La familiarización del dispositivo con el usuario en donde, se verifica el estado del sensor según el color del led muestra el funcionamiento del sensor, si el led indicador es de color rojo indica que la batería se encuentra agotándose por tal motivo la captación de los datos puede ser nula o errónea, cuando el led se encuentra de color azul indica un correcto funcionamiento del sensor mindwave mobile 2.

RESULTADOS: Cuando el led indicador se encuentra de color rojo quiere decir que la batería se encuentra agotándose por tal motivo la lectura de los datos puede ser imprecisa en este caso se realiza el cambio de la pila. Si el led indicador se encuentra de color azul indica que el sensor se encuentra en un correcto funcionamiento para la captación de los datos acatando que se debe estar colocado la pinza de la oreja la cual ayuda a la reducción del ruido externo, como se muestra en la Figura 91.

Figura 91

Estados del sensor EEG, izquierda sensor no conectado, derecha sensor conectado.



Fuente: Elaborado por el autor

4.3.3 Bloque 3: Comunicación

Este bloque es uno de los más importantes en el desarrollo del prototipo ya que permite la interconexión entre el sensor neurosky por bluetooth hacia el microprocesador Arduino mega, por tal motivo este proceso consta de dos etapas la primera etapa es el emparejamiento de el sensor neurosky mindwave 2 con el módulo hc-05 mediante una conexión bluetooth; para la segunda etapa se realiza la conexión del módulo hc-05 con el Arduino mega mediante la conexión de sus pines de TX y RX respectivamente de forma serial. Se realiza un test para la verificación de la interconexión de estos dispositivos de manera correcta como se muestra en la Tabla 35.

Tabla 35

Test de comunicación

TEST DE COMUNICACIÓN

BLOQUE DE PRUEBABloque de comunicación

Validación de la comunicación bluetooth entre el

DESCRIPCIÓN

Sensor EEG y el módulo hc-05 para el
microcontrolador Arduino mega.

PRERREQUISITOS:

- ✓ Adquisición de dispositivo neurosky mindwave mobile 2.
- ✓ Adquisición de modulo bluetooth hc-05.
- ✓ Adquisición de Arduino mega.

PASOS:

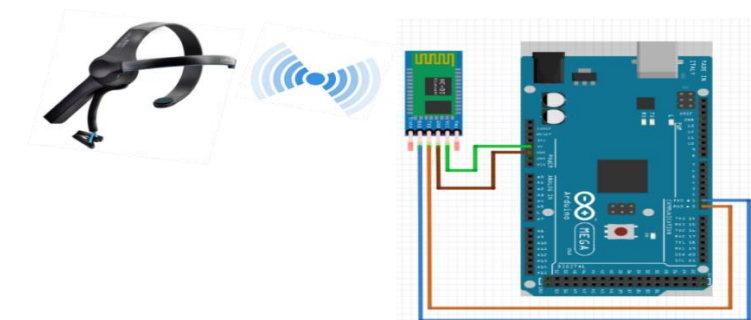
- ✓ Verificar la correcta comunicación entre el sensor EEG con el módulo HC-05.
 - ✓ Verificar la conexión del módulo HC-05 con el Arduino mega mediante los pines de TX y RX.
-

RESULTADOS ESTIMADOS: La verificación de la correcta comunicación entre el sensor neurosky mindwave mobile 2 con el módulo bluetooth hc-05 para este caso el módulo bluetooth deja de titilar si estos dispositivos se encuentran emparejados correctamente para el envío de datos.

RESULTADOS: Para la validación de la correcta configuración de emparejamiento entre el sensor mindwave mobile 2 con el módulo hc-05 es necesario realizar algunas configuraciones previas que se detallan en el apartado 3.5.3. Por tal motivo es importante comprender que la comunicación bluetooth se la realiza entre el dispositivo neurosky mindwave mobile y el módulo hc-05, mientras que para que esta información sea recibida por el microcontrolador Arduino mega se realiza la conexión desde el pin de TX del módulo hc-05 al ping de RX del microcontrolador como se muestra en la Figura 92.

Figura 92

Conexión de bloque de comunicación.



Fuente: Elaborado por el autor

Se valida el proceso de emparejamiento de la comunicación bluetooth, cuando no se encuentra encendido el sensor EEG el módulo hc-05 con su led que titila de manera constante indica que no se encuentra emparejado con ningún otro dispositivo inalámbrico como se muestra en la Figura 93.

Figura 93

Etapas de emparejamiento bluetooth

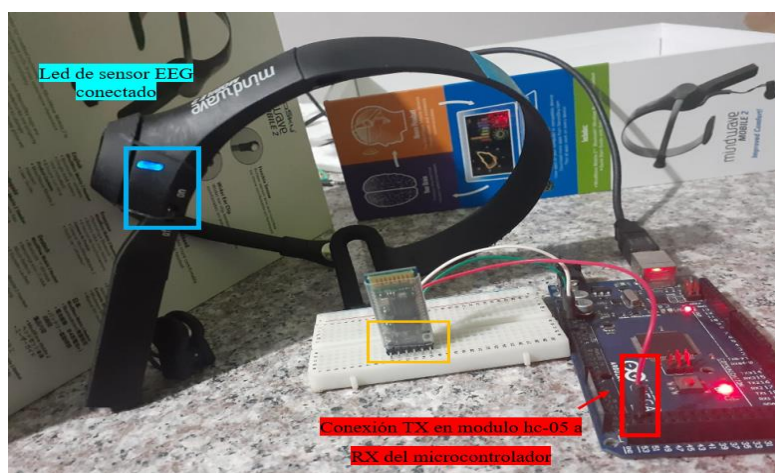


Fuente: Elaborado por el autor

Una vez que se enciende el sensor EEG, el módulo hc-05 comienza a emparejarse con el sensor cuando este proceso de emparejamiento se encuentra correcto el led indicador del módulo hc-05 cambia de estar constantemente titilando a ya no titilar y eso indica que los dispositivos se encuentran correctamente conectados como se muestra en la Figura 94.

Figura 94

Comunicación entre el sensor y modulo hc-05 correcta



Fuente: Elaborado por el autor

4.3.4 Bloque 4: adquisición y almacenamiento de los datos

Para el bloque de adquisición y almacenamientos de los datos se realiza un test de validación en donde se procede a verificar este bloque, en la Tabla 36 se muestra el test a realizar.

Tabla 36

Test de adquisición y almacenamiento de datos

TEST DE ADQUISICION Y ALMACENAMIENTO DE DATOS

BLOQUE DE PRUEBA	Bloque de adquisición y almacenamiento
DESCRIPCIÓN	Prueba de funcionamiento para el bloque de adquisición y almacenamiento de datos.
PRERREQUISITOS:	
<ul style="list-style-type: none"> ✓ Adquisición de dispositivo neurosky mindwave mobile 2. ✓ Adquisición de modulo bluetooth hc-05. ✓ Adquisición de Arduino mega. ✓ Adquisición de CoolTerm. 	
PASOS:	
<ul style="list-style-type: none"> ✓ Verificar la adquisición de datos en tiempo real y en formato ThinkGear tanto de señales EEG y valores de atención. ✓ Guardar los datos en un .txt para generar las bases de datos. 	
RESULTADOS ESTIMADOS:	
Toma de los datos en tiempo real de las señales emitidas por el sensor que son delta, theta, alfa, beta, gamma. Mediante la interconexión de hardware y software.	

Fuente: Elaborado por el autor

Resultados obtenidos: Una vez que el bloque de comunicación se encuentra funcionando de manera correcta es importante el generar el script para la adquisición de los datos, proceso que se encuentra detallado en el apartado 3.5.4.2.

En la Figura 95, se muestra los datos emitidos por el sensor neurosky mindwave mobile 2 los cuales son en tiempo real de las señales EEG, como también valores de la calidad de la señal y el nivel de atención que tiene el usuario validando el correcto funcionamiento de los dispositivos que interactúan.

Figura 95

Datos emitidos por el sensor Neurosky Mindwave Mobile 2 en el Arduino

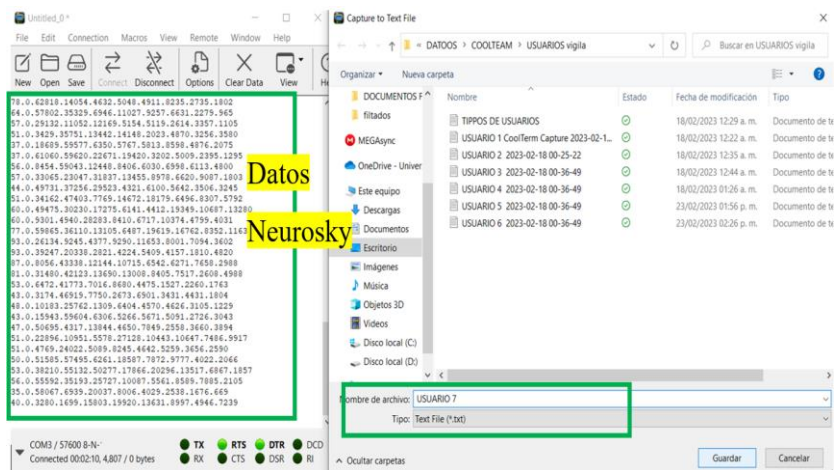
Attention	theta	low Alpha	high alpha	low beta	high beta	low gamma	medd gamma
44	17710	13275	11737	29825	7376	2863	2313
56	22507	25880	26445	14621	16317	10408	7824
69	38627	4935	12515	6403	6724	816	947
80	31089	6808	16159	8423	21974	8385	3884
88	10285	15676	6440	5391	4986	1665	2609
88	44859	1677	4516	15750	9228	8964	3513
93	23743	21834	5917	10420	10035	7909	5262
78	37231	26597	27099	45508	16741	3396	6353
64	44041	11968	2403	3726	2601	979	381
51	34699	4593	17054	4744	3313	3925	2550
40	34216	7501	6643	9495	13010	3356	5265

Fuente: Elaborado por el autor

Una vez realizada la validación de la adquisición de los datos EEG, se procede al almacenamiento de estos datos para generar las BDD respectivas a cada estado para los usuarios, mediante la utilización del software COOLTERM mismo que permite la lectura del puerto COM en donde se encuentra conectado el microprocesador para proceder a guardar los datos en un documento en formato .txt facilitando la manipulación y almacenamiento de esta gran cantidad de datos, como se muestra en la Figura 96.

Figura 96

Izquierda datos EEG, derecha almacenamiento en .txt con COOLTERM



Fuente: Elaborado por el autor

4.3.5 Bloque 5: Procesamiento de datos

Para la validación del bloque de procesamiento de datos, se realiza un test en donde se debe verificar el cumplimiento de cada fase del proceso KDD proceso que se encuentra detallado en el apartado 3.5.5 y obtener el mejor resultado de software con hardware trabajando en conjunto. Por tal motivo se lleva a cabo un test para la validación del funcionamiento del bloque de procesamiento como se observa en la Tabla 37.

Tabla 37

Test de bloque de procesamiento

TEST DE PROCESAMIENTO	
BLOQUE DE PRUEBA	Bloque de procesamiento de datos mediante el proceso KDD.

DESCRIPCIÓN

Prueba de funcionamiento para el bloque de procesamiento de datos.

PRERREQUISITOS:

- ✓ Adquisición de neurosky mindwave mobile 2.
 - ✓ Bloque de alimentación.
 - ✓ Bloque de captación EEG.
 - ✓ Bloque de comunicación.
 - ✓ Bloque de adquisición y almacenamiento de datos.
 - ✓ Algoritmo de clasificación supervisado con mayor precisión.
 - ✓ Datos de entrenamiento y prueba.
-

PASOS:

- ✓ Montaje de bloques de alimentación, captación, comunicación, adquisición y procesamiento.
 - ✓ Realizar la verificación de patrones que permitan identificar diferencias entre cada estado del usuario.
 - ✓ Cargar script con el código que permite adquirir los datos EEG que envía el Arduino hacia el Raspberry pi 3.
 - ✓ Obtener los valores obtenidos por el bloque de adquisición para ser sometidos a la etapa del filtrado de estos.
 - ✓ Cargar script en Raspberry con el código del algoritmo de clasificación ya entrenado Random Forest.
 - ✓ Cargar los valores filtrados a un buffer para proceder a la comparación con el algoritmo de clasificación.
 - ✓ Aplicar clasificación supervisada a los valores adquiridos y filtrados.
 - ✓ Crear la matriz de confusión
 - ✓ Calcular la precisión del algoritmo
-

RESULTADOS ESTIMADOS:

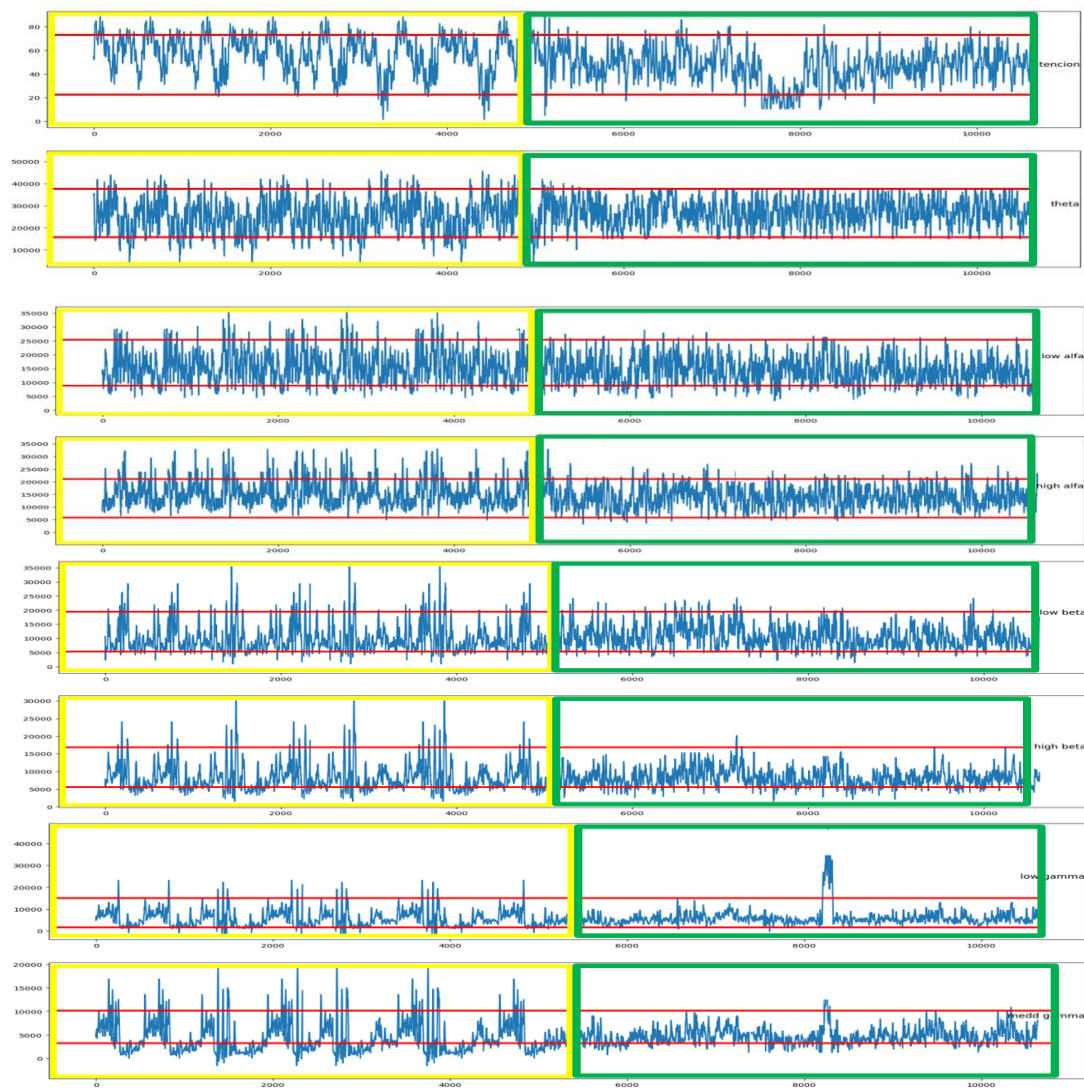
El acoplamiento correcto de todos los elementos que conforman el prototipo embebido y verificar el correcto rendimiento del algoritmo de clasificación.

Fuente: Elaborado por el autor

Resultados obtenidos: Una vez que se tiene elaboradas las respectivas BDD que corresponden tanto al estado de somnolencia como el estado de vigila o despierto , se procede a identificar patrones de los datos que permitan establecer las diferencias entre estos estados como lo son la diferencia de amplitudes en estos estados proceso que se muestra en la Figura 97, en donde la primera porción de los datos marcados de amarillo corresponden a los datos del usuarios en un estado de somnolencia o sueño y la otra porción de los datos pertenecen a un estado de vigila o despierto; en donde se puede identificar claramente como el usuario si se encuentra en un estado u otro las señales son distintas en cada caso.

Figura 97

Señales en estado de somnolencia(amarrillo) y vigila (verde) de la BDD total



Fuente: Elaborado por el autor

En la Figura 98, se verifica el correcto acoplamiento entre el sensor EEG con el módulo HC-05 el cual envía los datos al Arduino mismo que realiza el reenvío de datos EEG en tiempo real hacia el Raspberry pi.

Figura 98

Datos recibidos por el Raspberry desde el Arduino

```
(myenv) raspberry@raspberrypi:~/Desktop $ python3 2lecturadatos.py
51 16477 1854 14579 5830 12317 5613 3518
54 26554 11970 5451 5580 6303 5966 3066
44 52040 8921 20012 6501 3889 6101 3349
50 6964 3927 1968 3182 2159 1389 454
30 20274 6463 2046 2832 2473 861 2002
29 10582 3335 3642 3303 2077 1796 1066
30 19503 36221 17920 28107 19047 9419 6381
37 17204 7251 4370 2465 7655 4062 3064
34 65001 44208 14719 12600 7975 6852 2897
30 14616 7040 4033 4779 2589 7714 4566
47 33797 5058 39393 39622 5184 21604 16369
27 21308 29715 10374 27477 14246 8449 4340
50 7222 5565 4799 7361 6729 3508 1361
57 11790 6352 17090 2639 7525 2496 2638
35 33595 20475 6509 3448 4945 2211 2278
51 20369 1050 2003 4441 4461 2258 2280
29 54103 17864 49560 25609 7266 3247 4837
13 19504 27702 25794 5113 7309 2696 3436
30 8594 4703 9092 6995 3813 4844 2065
34 10302 3526 5644 3290 4015 3880 2362
```

Fuente: Elaborado por el autor

Con el correcto funcionamiento del bloque tanto de adquisición y comunicación de los datos el Raspberry procede a realizar la identificación de la etiqueta respectiva como se muestra en la Figura 99.

Figura 99

Identificación de etiqueta de predicción del algoritmo Random Forest.

```

rasberry@raspberrypi: ~/Desktop
rasberry@raspberrypi:~ $ source myenv/bin/activate
(myenv) rasberry@raspberrypi:~ $ cd Desktop/
(myenv) rasberry@raspberrypi:~/Desktop $ python3 clasificacionsomnolencia.py
Prediccion: [0] ← Etiqueta
DESPIERTO
Prediccion: [0]
DESPIERTO ← Estado Usuario
Prediccion: [0]
DESPIERTO
Prediccion: [0]
DESPIERTO
Prediccion: [0]
DESPIERTO
Prediccion: [0]
DESPIERTO
Prediccion: [0]
DESPIERTO

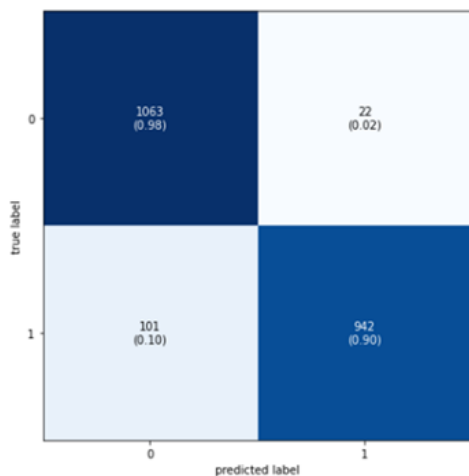
```

Fuente: Elaborado por el autor

En la Figura 100, se muestra la matriz de confusión la cual permite el cálculo del desempeño del algoritmo de supervisión clasificado Random Forest.

Figura 100

Matriz de confusión



Fuente: Elaborado por el autor

Con los valores que otorga la matriz de confusión que fue evaluada en el software de Google Colab se procede a verificar la exactitud, donde se puede determinar que el algoritmo implementado tiene una exactitud de clasificación de

$$Exactitud = \frac{1063 + 942}{1063 + 942 + 101 + 22}$$

$$Exactitud = \frac{2005}{2128}$$

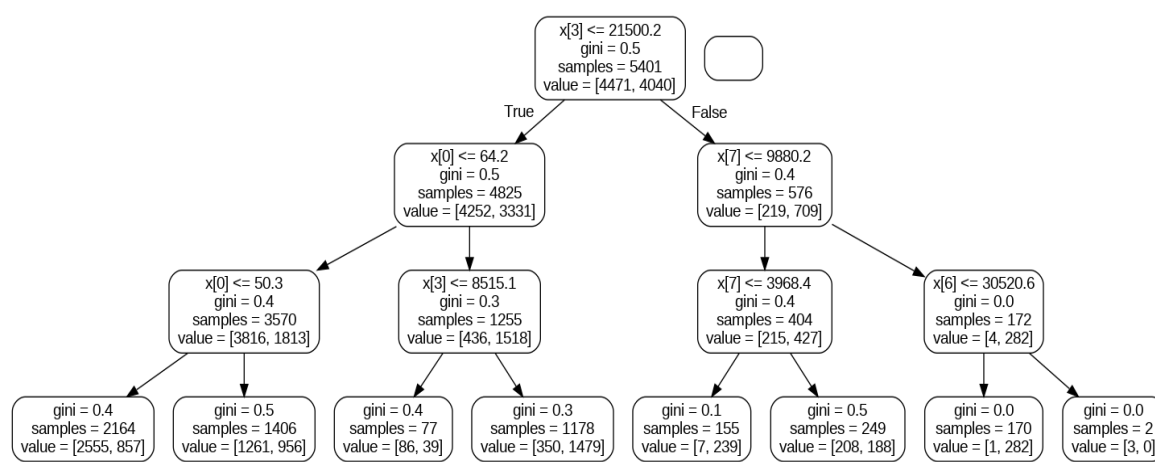
$$Exactitud = 0.942 * 100$$

$$Exactitud = 94.2\%$$

En la Figura 101, se muestra cómo se lleva a cabo el proceso de la determinación de la respuesta dada por el algoritmo Random Forest mismo que es realizada por número mayoritario de votos.

Figura 101

Toma de decisión de Random Forest



Fuente: Elaborado por el autor

4.4 Pruebas finales de implementación del prototipo completo

Se determino distintas pruebas de testeo para la validación de funcionalidad de cada bloque realizado, de igual forma se realizaron otro tipo de pruebas que se centran para la verificación del algoritmo el cual detecta la somnolencia mediante el análisis de ondas EEG. Dichas pruebas se las realizo en base al siguiente plan de actividades misma que se encuentra detallada en la *Tabla 38*, que contiene usuarios de prueba, sitio en donde se desarrolla las pruebas, resultados y también el tiempo que duraran dichas pruebas.

Tabla 38*Cronograma de pruebas finales*

Actividades para las pruebas del sistema			
	Etapa 1		Etapa 2
Tipo de prueba	Verificar que el envío de datos EEG del sensor al Raspberry y evaluar que el sensor no sea invasivo.		Detección de estado de somnolencia y vigila
Horario	Dia, tarde y noche		Dia, tarde y noche
Sujeto de prueba	Conductores de automóviles		Conductores de automóviles
Lugar	Cabina del automóvil		Cabina del automóvil
Resultados Estimados	Se busca que el usuario se familiarice con el dispositivo en funcionamiento y validar la obtención de señales EEG.		Se espera alertar al usuario cuando se encuentre en estado de somnolencia.
Duración de pruebas	15 min		30 min
Tiempo de duración de pruebas	3 días		2 semanas

Fuente: Elaborado por el autor

En la primera etapa del cronograma, se presente un 8% de error esto se debe a diversos factores que alteran una correcta adquisición de datos EEG como lo son una mala colocación del sensor en el usuario produce errores de adquisición de datos, además hay que tener en consideración también que las pilas de alimentación al sensor se encuentren en buen estado debido a que estos son los principales errores para una correcta adquisición de datos y de paso permite la familiarización del usuario con el sensor. En la *Tabla 39* se presentan los resultados obtenidos en dicha prueba.

Tabla 39

Pruebas etapa 1

	Prueba 1	Prueba 2	Prueba 3	Total
Horario	Dia	Tarde	Noche	
# pruebas	25	25	25	75
Pruebas exitosas	23	22	24	67
Pruebas Fallidas	2	3	1	6
% aciertos	92%	88%	96%	92 %
% errores	8%	12%	4%	8%

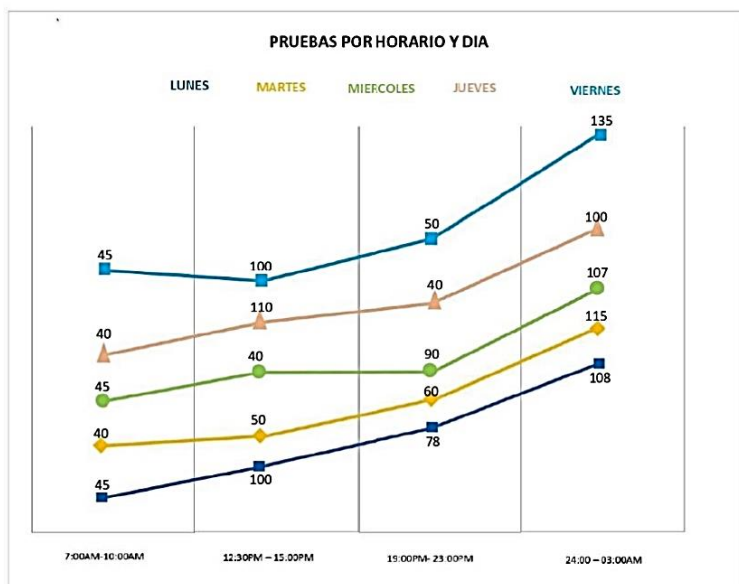
Fuente: Elaborado por el autor

En la segunda etapa, se verifica la detección de las etiquetas que validan el estado del conductor estos estados son tanto de vigila o despierto o somnolencia el cual se realiza durante una semana con un conductor en los horarios establecidos, pero con distintos usuarios de prueba, debido que el sistema debe ser validado con usuarios de distintas características para un mejor rendimiento del sistema. Los resultados se muestran en la *Tabla 40* donde se detallan las pruebas realizadas en la primera semana.

En la Figura 103, se puede observar donde los conductores presentan el cambio de estado a somnolencia durante la conducción. El estado de detección se determina por la aplicación del algoritmo Random Forest mismo que se encuentra previamente entrenado donde si da la etiqueta 1 indica que el usuario se encuentra en esta de sueño por tal motivo se activa la alerta del prototipo, si la etiqueta es 0 permite determinar el buen estado del conductor denominado vigila o despierto, para sustentar los horarios en donde son realizados las pruebas se encuentra descrito en el apartado 2.5.2 en se detalla los horarios y causantes en donde es más propenso que el conductor este en estado de somnolencia. En la gráfica se evidencia en el eje de las x los escenarios en donde los conductores pueden ser más propensos de caer en estado de sueño o somnolencia. En el eje y se tiene los valores en donde el sistema detecta con más frecuencia la etiqueta 1 denominada para el estado de somnolencia de manera constante.

Figura 103

Grafica de pruebas etapa 2



Fuente: Elaborado por el autor

En la Tabla 41, se puede observar el porcentaje respectivo para cada cantidad de datos obtenido para las pruebas en los días y horarios respectivos, partiendo que el total de datos adquiridos es 1500 durante el tiempo de prueba, por tanto, se indica el porcentaje de datos en donde se detecta si el usuario entra en estado de somnolencia con la etiqueta 1.

Tabla 41

Porcentaje de datos de prueba

Total, de datos obtenidos 1500	Porcentaje 100%
40	2% (98% etiqueta = 0)
60	4% (96% etiqueta = 0)
75	5% (95% etiqueta = 0)
105	7% (93% etiqueta = 0)
135	9% (91% etiqueta = 0)

Fuente: Elaborado por el autor

4.5 Análisis de resultados

Realizado las pruebas respectivas del prototipo en los diferentes escenarios permitiendo la identificación de una constante en los usuarios, cuando el usuario realiza sus actividades diarias y al finalizar su día, cuando realiza la conducción es posible que el usuario entre en un estado de somnolencia teniendo en cuenta que el escenario será desde las 19:00 a 23:00 horas siendo más evidente el estado de somnolencia en horas más altas de la noche. Pero también existe un gran cambio de estado en usuarios en las horas después de la comida (almuerzo) dispuesto entre 12pm a 15:00 pm.

En el escenario más frecuente del cambio de estado de vigila a somnolencia se presenta desde las altas horas de la noche y primeras horas de la madrugada mismas que van entre las 23:30 pm y 3:00 am horas en donde se pudo evidenciar de manera más evidente que el usuario entra en estado de somnolencia en donde se puede evidenciar que el usuario reduce en gran medida la concentración y control tras el volante. El análisis de estos resultados se encuentra detallados en la Figura 103; debido a la realización de estas pruebas se pudo determinar las fallas que se pueden suscitar en el prototipo para poder solventarlas, por otra parte se pudo evidenciar que existe una mayor tendencia en quedarse dormido mientras conduce en los horarios de media noche a 4:00 am, por tal motivo si este se encuentra en un estado de sueño el prototipo procede a realizar la activación de la alarma sonora siendo un indicador que el conductor debe tener un momento de descanso para prevención de un posible accidente.

Este tipo de asistentes de conducción no podrá eliminar los accidentes en las vías, pero puede ayudar a reducir de gran manera el riesgo de tener los accidentes viales, si el conductor se acata a tomar los descansos debidos cuando el prototipo se lo recomiende con la alerta sonora.

4.6 Costo del Prototipo

En el presente apartado se realiza un análisis del costo de cada uno de los elementos que forman parte del prototipo tomando en cuenta el precio por unidad y el precio incluido valor de envío si es necesario, para la producción y construcción de varios o más prototipos. En la Tabla 42 se muestra detallado el costo final del prototipo con todos los componentes que forman parte de este proyecto.

Tabla 42*Costo prototipo*

Costo final del prototipo			
Cantidad	Descripción	Costo Unitario (USD)	Costo total (USD)
1	Sensor Neurosky Mindwave Mobile 2	190	190
1	Arduino Mega	25	25
1	Modulo hc-05	6	6
1	Raspberry pi3 B+	55	55
2	Diodos led	0.50	1
1	Buzzer	3	3
1	Case nodo principal	10	10
1	Conversor 12 Vcc a 5 Vcc	10	10
		Total	300

Fuente: Elaborado por el autor

CONCLUSIONES

El prototipo para la detección del estado de somnolencia en conductores cumplió de manera satisfactoria los requerimientos establecidos de Stakeholders, sistema y arquitectura, en donde, la implementación del diseño físico cumple con las expectativas de uso para el usuario final, teniendo un producto de un tamaño el cual es adecuado para que no obstruya ni entorpezca las actividades del conductor.

El uso del sensor adecuado es importante para llevar a cabo este tipo de investigación como lo es el dispositivo Neurosky Mindwave Mobile 2 mismo que es fácil de adquirir en el mercado y cuenta con gran flujo de información que permite la obtención de datos EEG, además es compatible con diversos dispositivos externos como lo es el módulo hc-05 y Arduino para realizar análisis de datos de una mejor manera.

El uso de placas de desarrollo Open Source como lo son tanto el Arduino como la Raspberry pi, presentan grandes posibilidades para la creación y desarrollo de proyectos de innovación e investigación, ya que permite al desarrollador lograr adaptar a estas placas con el objetivo que se desee, de igual manera permite el añadir dispositivos electrónicos externos que mejora el desarrollo de proyectos.

El uso de software que permite el análisis y tratamiento de datos como Google Colab es fundamental ya que cuenta con diversas librerías que permite realizar un análisis exhaustivo de los datos permitiendo así el obtener los resultados deseados para la investigación del prototipo implementando algoritmos de machine learning para obtener el resultado final con los datos a analizar.

Se demostró que existe una variedad de patrones entre los datos obtenidos cuando el usuario se encuentra en un estado de adormecimiento o somnolencia las ondas EEG tienen amplitudes más altas y poco estables a diferencia de cuando el usuario se encuentra en un estado de vigila o despierto las ondas que emite el usuario son de menor amplitud como también más uniformes.

Se pudo concluir que al establecer un número considerable de muestras con varias características para la generación de la matriz de entrenamiento permite que el algoritmo de clasificación supervisada obtenga una mayor precisión superior al 90%, lo cual permite al usuario el poder tener un prototipo electrónico confiable y óptimo para realizar las predicciones del conductor.

Las pruebas de funcionamiento permitieron demostrar la efectividad del prototipo permitiendo validar su exactitud en diversos escenarios planteados dando la validez necesaria para el uso del prototipo que es adaptable a futuras actualizaciones y mejoras en todos sus ambientes.

Se pudo concluir que mediante las pruebas realizadas en los diferentes escenarios se identificó patrones en donde se evidencia que el usuario entra en un estado de alerta o somnolencia más frecuentemente, mismo que se encuentra en los horarios de 13:00 pm a 15:00 pm como también en el horario de la 23:00 pm a 3:00 am debido a diversas situaciones tanto fisiológicas como ambientales. Por tanto, cuando el conductor se encuentra en un estado de alerta o somnolencia el prototipo activara la alarma sonora permitiendo alertar al usuario con el fin de que este tome medidas precautelaría para evitar algún evento desfavorable en la carretera.

RECOMENDACIONES

Se recomienda para el análisis de ondas EEG el obtener la mayor cantidad de datos en diversos escenarios para que así sea más fácil el poder observar de manera más exacta los diversos patrones que emite el usuario en cada escenario propuesto.

Es recomendable usar un conversor de voltaje adecuado ya el automóvil emite 12 Vcc y los dispositivos que forman parte del hardware trabajan de manera adecuada a 5Vcc o 3.3 Vcc.

Se recomienda adecuar de manera correcta las configuraciones de comunicación entre el sensor EEG con el módulo HC-05 ya que si estas configuraciones como lo es la tasa de baudios son diferentes de 57600 que son los recomendados para este dispositivo el microcontrolador no será capaz de adquirir los datos que son emitidos por el sensor.

Para la elaboración de la matriz la cual será de entrenamiento del algoritmo de machine learning se recomienda tomar una cantidad amplia de datos en diferentes escenarios para así lograr identificar de mejor manera los patrones de comportamiento del usuario en dichos escenarios planteados y poder determinar de manera correcta las etiquetas del mismo.

El prototipo de detección de somnolencia tiene que estar ubicado de manera correcta y estratégica en el habitáculo interior del automóvil, para que no sea invasivo ni tampoco llegue a entorpecer las actividades del conductor.

Para el diseño del prototipo en la etapa de procesamiento de datos es recomendable usar placas de desarrollo que cuenten con una alta capacidad de procesamiento como Raspberry pi ya que este es capaz de realizar análisis de una gran cantidad de datos en tiempo real sin problemas y limitaciones por tanto optimizan la ejecución del prototipo.

REFERENCIAS

- {Altran}. (s.f.). *Conducir con sueño o cansancio*. Dgt.es: <https://www.dgt.es/muevete-conseguridad/evita-conductas-de-riesgo/Conducir-con-sueno-o-cansancio>
- {canalMOTOR}. (25 de Febrero de 2022). *canalMOTOR*. Detector de fatiga en el coche Volkswagen: <https://www.motor.mapfre.es/consejos-practicos/seguridad-vial/volkswagen-detector-de-fatiga-bosch-de-serie/>
- Afifi, A., & Bergman, R. (2020). *Neuroanatomía funcional: texto y atlas* (tercera ed.). McGrawHill.
- Aguayo, P. (19 de Enero de 2019). *Arduino.cl*. Arduino.cl - Compra tu Arduino en Línea: Arduino UNO
- ALTER TECHNOLOGY. (29 de Abril de 2021). ALTER TECHNOLOGY: <https://www.laboratoriodecertificacion.es/breve-explicacion-del-proceso-kdd/>
- ANT. (2021). *Agencia Nacional de Tránsito*. Agencia Nacional de Tránsito: <https://www.ant.gob.ec/>
- Arduino Mega 2560*. (23 de Agosto de 2018). Retrieved 17 de Enero de 2023, from Proyecto Arduino: <https://proyectoarduino.com/arduino-mega-2560/>
- Automotive - optalert. (s.f.). *Opalert*. <https://www.optalert.com/solutions/automotive/>
- AutosMagazine. (22 de Diciembre de 2020). *AutosMagazine*. AutosMagazine : <https://automagazine.ec/los-autos-mas-buscados-online-en-ecuador-en-2020/>
- B105 LAB. (25 de Octubre de 2017). *b105 lab*. b105 lab: <http://elb105.com/obtencion-de-indicadores-de-fatiga-mediante-el-electroencefalograma/>

BuscaEU. (27 de Febrero de 2019). Electrodo húmedo, seco, activo y pasivo. ¿Qué son, y cuál elegir?: <https://www.brainlatam.com/blog/electrodos-humedos-secos-activos-y-pasivos.-%C2%BFque-son-y-cual-elegir-412>

Drowsiness. (14 de Julio de 2021). Medlineplus.gov:
<https://medlineplus.gov/ency/article/003208.htm>

EMOTIV. (03 de Febrero de 2016). Insight Brainwear® 5 channel wireless EEG headset.

Eyecomtec.com. (2019). Retrieved 1 de Diciembre de 2022, from NeuroSky MindWave:
<https://eyecomtec.com/es/3401-NeuroSky-MindWave>

Formation Data Science / DataScientest.com. (12 de Diciembre de 2021). Descubra el algoritmo KNN : un algoritmo de aprendizaje supervisado: <https://datascientest.com/es/que-es-el-algoritmo-knn>

Geek Factory. (01 de Julio de 2014). Retrieved 17 de Enero de 2023, from HC-05 Módulo Bluetooth maestro esclavo:
<https://www.geekfactory.mx/tienda/modulos/radiofrecuencia/hc-05-modulo-bluetooth-maestro-esclavo/>

Instituto Nacional del Cáncer. (02 de Febrero de 2011). *Instituto Nacional del Cáncer.*
Diccionario de cáncer del NCI:
<https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/sueno-mor>

Java. (30 de Octubre de 2011). Techopedia.com: What is Java? - Definition from Techopedia

Jobit, J. (19 de Marzo de 2022). *Types of Arduino boards – quick comparison on specification and features*. Circuitdigest.com: <https://circuitdigest.com/article/different-types-of-arduino-boards>

Johns Hopkins University. (14 de Agosto de 2021). *Johns Hopkins Medicine*. Johns Hopkins Medicine: <https://www.hopkinsmedicine.org/health/conditions-and-diseases/anatomy-of-the-brain#:~:text=The%20largest%20part%20of%20the,hearing%2C%20touch%20and%20other%20senses>

Laguna , M. (24 de Agosto de 2022). *Sistema nervioso central (SNC)*. KEN HUB: www.kenhub.com

Martins, J. (20 de Abril de 2021). *asana . anasa* : <https://asana.com/es/resources/iterative-process>
Mathworks.com. (s.f.). Diseño de filtros FIR - MATLAB & Simulink - MathWorks América Latina: <https://la.mathworks.com/help/signal/ug/fir-filter-design.html>

Mayo Clinic. (20 de Noviembre de 2020). Desgaste laboral: cómo detectarlo y tomar medidas: <https://www.mayoclinic.org/es-es/healthy-lifestyle/adult-health/in-depth/burnout/art-20046642>

MedlinePlus. (14 de Julio de 2021). *Drowsiness*. Medlineplus.gov: <https://medlineplus.gov/ency/article/003208.htm>

MERCEDES-BENZ. (s.f.). *mercedes-benz.com*. mercedes-benz.com: <https://www.la.mercedes-benz.com/es/passengercars/mercedes-benz-cars/models/gla/gla->

suv/explore.pi.html/mercedes-benz-cars/models/gla/gla-suv/explore/intelligent-technologies/attention-assist

Muse. (08 de Agosto de 2018). *Muse 2 - Tu Asistente Personal de Meditacion*:

<https://choosemuse.com/es/muse-2/>

NeuroHealth Associates. (29 de Mayo de 2013). *The science of brainwaves - the language of the*

brain: <https://nhahealth.com/brainwaves-the-language/>

Neurosky.com. (06 de Mayo de 2014). *esenses_tm* [NeuroSky Developer - Docs]:

https://developer.neurosky.com/docs/doku.php?id=esenses_tm

Neurosky.com. (17 de 01 de 2017). *thinkgear_communications_protocol* [NeuroSky Developer - Docs]:

https://developer.neurosky.com/docs/doku.php?id=thinkgear_communications_protocol

Olsen, E., & Hagen, J. (02 de Abril de 2014). *Nine more digital health projects on Indiegogo*.

MobiHealthNews: <https://www.mobihealthnews.com/31643/eight-more-digital-health-projects-on-indiegogo>

Opensourceimaging. (s.f.). Retrieved 12 de 02 de 2023, from OpenBCI – open-source EEG:

<https://www.opensourceimaging.org/project/openbci/>

Orbe, A. (16 de Julio de 2012). *Electroencefalograma EEG para detectar imágenes del enemigo*

[Fotografía]. hipertextual: www.hipertextual.com

Primicias. (14 de Marzo de 2022). ¡Alerta!, no se quede dormido en el volante:

https://www.primicias.ec/nota_comercial/autos/garage/talleres/alerta-no-se-queda-dormido-en-el-volante/#gsc.tab=0

Python. (15 de Junio de 2017). Lenguajes de programación:

<https://lenguajesdeprogramacion.net/python/>

Python. (06 de Junio de 2017). Lenguajes de programación:

<https://lenguajesdeprogramacion.net/python/>

Ramos, F. (2009). Basic techniques of electroencephalography: principles and clinical applications. *Scielo*, 2-3.

Readthedocs.io. (15 de Febrero de 2023). Readthedocs.io:

<https://lazypredict.readthedocs.io/en/latest/>

SANDOROBOTICS. (24 de Agosto de 2018). Diadema NeuroSky MindWave Mobile 2:

<https://sandorobotics.com/producto/sen-14758/>

SAYAN, K. (29 de Septiembre de 2011). *GeeksforGeeks*. GeeksforGeeks:

<https://www.geeksforgeeks.org/software-engineering-iterative-waterfall-model/>

Shiklo, B. (08 de Agosto de 2019). *8 software development models: Sliced, diced and organized in charts*. ScienceSoft: <https://www.scnsoft.com/blog/software-development-models>

Sitrack, E. (14 de Diciembre de 2021). *El rol vital de los sistemas de detección de fatiga y somnolencia para vehículos*. Sitrack.com: <https://blog.sitrack.com/el-rol-vital-de-los-sistemas-de-deteccion-de-fatiga-y-somnolencia-para-vehiculos>

Soares, S., Monteiro, T., Lobo, A., Couto, A., Cunha, L., & Ferreira, S. (2020). Analyzing driver drowsiness: From causes to effects. *Sustainability*, 12(5), 1971.

<https://doi.org/10.3390/su12051971>

- Sruthi, E. R. (17 de Junio de 2021). *Understand Random Forest algorithms with examples(updated 2023)*. Analytics Vidhya:
<https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>
- Subramaniyam, N. (15 de Junio de 2021). *Sapienlabs.org*. Retrieved 20 de Julio de 2022, from
 Wet and dry electrodes for EEG – sapien labs: <https://sapienlabs.org/lab-talk/wet-and-dry-electrodes-for-eeg/>
- TK. (03 de Marzo de 2014). *TEKD*. IFocusBand teaches a calm mind for better performance:
<https://tekd.com/ifocusband-teaches-a-calm-mind-for-better-performance/>
- Tu-mejor-viaje.es*. (09 de Febrero de 2022). ¿Que puede causar somnolencia en un conductor?:
<https://tu-mejor-viaje.es/que-puede-causar-somnolencia-en-un-conductor/>
- UDOO. (17 de Noviembre de 2021). *UDOO: Android Linux Arduino in a tiny single-board computer*. Kickstarter: <https://www.kickstarter.com/projects/udoo/udoo-android-linux-arduino-in-a-tiny-single-board>
- Unidad Nacional de Seguridad Vial*. (12 de Diciembre de 2020). Somnolencia y conducción:
<https://www.gub.uy/unidad-nacional-seguridad-vial/comunicacion/publicaciones/somnolencia-conduccion>
- Uribe, C., & Arana, C. (2010). Electrodiagnóstico:electroencefalografía. En C. Uribe, & C. Arana, *Neurología* (séptima ed., págs. 70-72). Medellin, Colombia: CIB.
- Veron, O. (2018). Modelos fisiológicos que explican, el estrés y la fatiga. *VIDA UCALP*.
- Vigo*. (s.f.). vigo: <https://www.wearvigo.com/fleets>

Vivas , J. (8 de Enero de 2019). *Sistema Nervioso Central: Funciones y partes*. Psicología-online: www.psicologia-online.com

What is a raspberry pi? (2019). En J. Barkstrom, *Introduction to the Raspberry Pi*. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-5571-1_1

Xataka. (16 de Abril de 2018). *Raspberry Pi 3 Model B+, análisis: más potencia y mejor WiFi para un miniPC que sigue asombrando*. Xataka.com: <https://www.xataka.com/ordenadores/raspberry-pi-3-model-b-analisis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando>

ANEXOS

Anexo A

Encuesta

Encuesta realizada a conductores

Formato de la encuesta realizada a conductores mismos que realizan diferentes actividades como: comerciantes, constructores, mecánicos, taxistas, estudiantes, etc. Debido a sus diferentes actividades en sus labores conllevan a tener distintos criterios sobre el efecto del estado de somnolencia al volante.

1. ¿Conoce sobre el estado de somnolencia?

Si () No ()

2. ¿Cuáles cree usted que son las causas de sueño o somnolencia durante la conducción?

- Cansancio
- Dormir poco
- Estrés
- Demasiado calor

3. ¿Alguna vez ha tenido la sensación de quedarse dormido por corto de tiempo mientras conduce?

Si () No ()

4. ¿Cree usted que es importante implementar un sistema de detección de somnolencia mediante frecuencias neuronales en su automóvil?

Si () No ()

5. ¿Considera que el sistema puede ayudar a los conductores para evitar accidentes?

Si () No ()

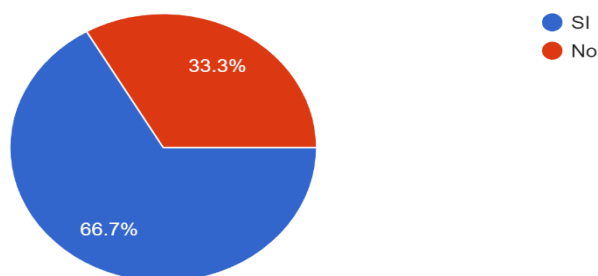
6. ¿Cree que el sistema debe coexistir con el automóvil?
- Si () No ()
7. ¿Como cree usted que es la mejor manera para dar a conocer los datos?
- Interfaz grafico
 - Alerta
 - Mensaje de texto
 - Aplicación Móvil
8. ¿Considera que es importante el indicar los niveles de las frecuencias neuronales?
- Si () No ()
9. ¿Cree que el usuario debe tener acceso a la manipulación total del sistema?
- Si () No ()
10. ¿Considera difícil la manipulación del sensor neuronal para su encendido/apagado?
- Si () No ()
11. ¿Considera incomodo el uso del sensor neuronal al conducir?
- Si () No ()
12. ¿Cuál de los siguientes considera que es el mejor método de alimentación eléctrica para el sistema de detección del estado de somnolencia?
- Batería
 - Al sistema eléctrico del automóvil
 - Mediante alimentación autónoma (panel solar)

Tabulación de encuesta

A continuación, se muestra el resultado de la encuesta realizada a un total de 12 posibles usuarios del sistema teniendo como resultado los siguientes datos que se muestran a continuación por cada pregunta planteada.

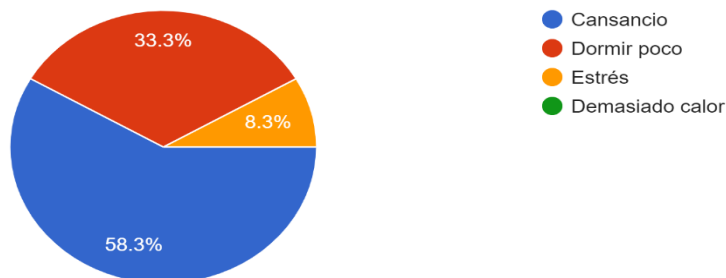
1. ¿Conoce sobre el estado de somnolencia?

12 respuestas



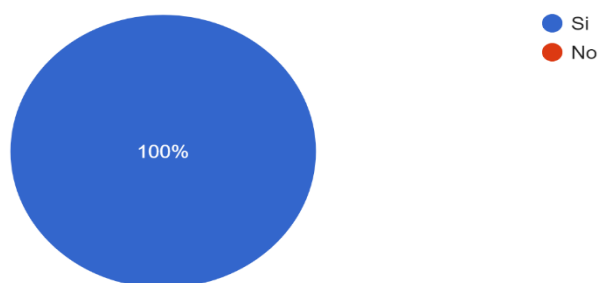
2. ¿Cuál cree usted que son las causas de sueño o somnolencia durante la conducción?

12 respuestas



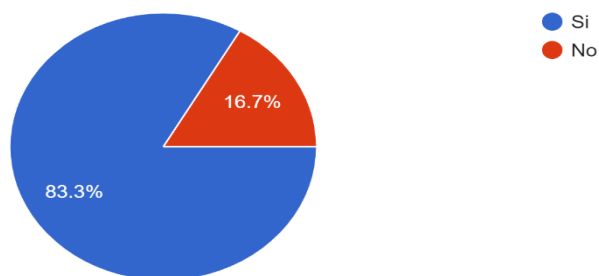
3. ¿Alguna vez ha tenido la sensación de quedarse dormido por corto de tiempo mientras conduce?

12 respuestas



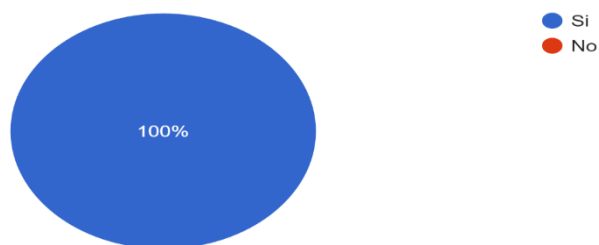
4. ¿Cree usted que es importante implementar un sistema de detección de somnolencia mediante frecuencias neuronales en su automóvil?

12 respuestas



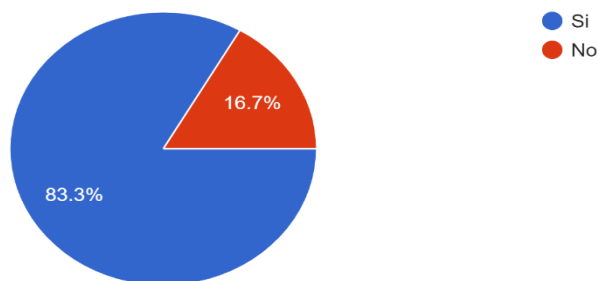
5. ¿Considera que el sistema puede ayudar a los conductores para evitar accidentes?

12 respuestas



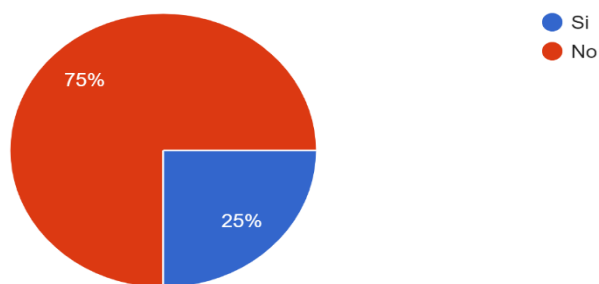
6. ¿Cree que el sistema debe coexistir con el automóvil?

12 respuestas



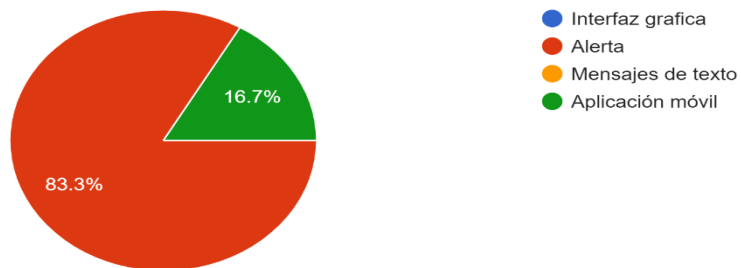
7. ¿Considera que es importante el indicar los niveles de las frecuencias neuronales?

12 respuestas



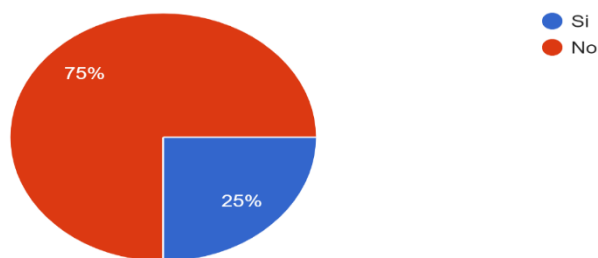
8. ¿Como cree usted que es la mejor manera para dar a conocer los datos?

12 respuestas



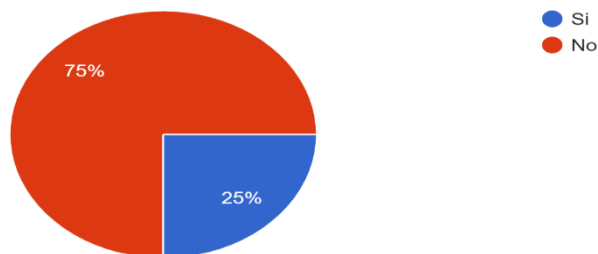
9. ¿Cree que el usuario debe tener acceso a la manipulación total del sistema de detección de somnolencia?

12 respuestas



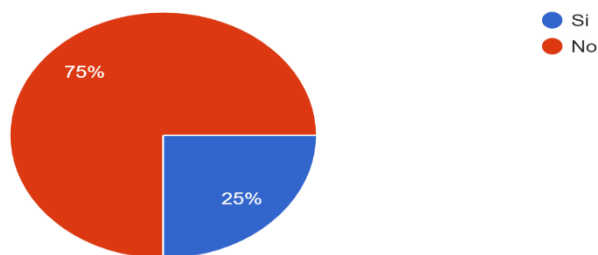
10. ¿Considera difícil la manipulación del sensor neuronal para su encendido/apagado?

12 respuestas



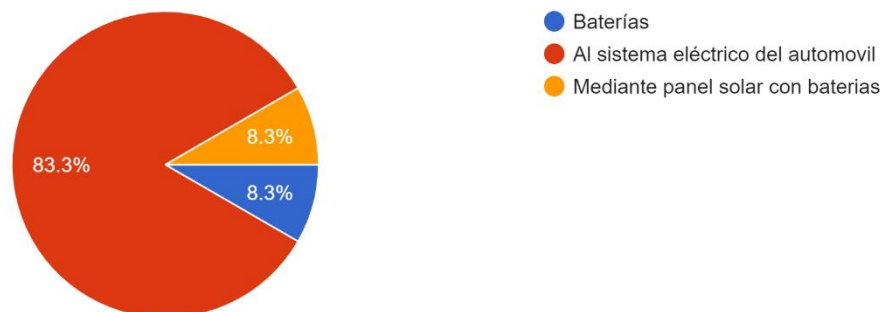
11. ¿Considera incomodo el uso del sensor neuronal para conducir?

12 respuestas



12. ¿Cual de los siguientes considera que es le mejor método de alimentación eléctrica para el sistema de detección del estado de somnolencia ?

12 respuestas



ESCALA DE SOMNOLENCIA DE EPWORTH PARA CONDUCTORES:

Instrucciones: Lee cada situación descrita a continuación y evalúa tu probabilidad de quedarte dormido mientras conduces en cada una de ellas en una escala del 0 al 3, según las siguientes opciones:

0 = Nunca me quedo dormido mientras conduzco

1 = Baja probabilidad de quedarme dormido mientras conduzco

2 = Moderada probabilidad de quedarme dormido mientras conduzco

3 = Alta probabilidad de quedarme dormido mientras conduzco

Por favor asigna un valor del 0 al 3 a cada una de las siguientes situaciones:

- 1) Conduciendo en una autopista recta y sin mucho tráfico. ____
- 2) Conduciendo en una carretera con curvas pronunciadas. ____
- 3) Conduciendo en un tramo monótono y aburrido de la carretera. ____
- 4) Conduciendo después de haber dormido menos de 5 horas la noche anterior. ____
- 5) Conduciendo durante un largo viaje sin hacer paradas frecuentes. ____
- 6) Conduciendo en condiciones de tráfico intenso. ____
- 7) Conduciendo después de consumir alcohol. ____
- 8) Conduciendo en un día caluroso y con el aire acondicionado apagado. ____
- 9) Conduciendo durante la noche o en horarios en los que normalmente estás dormido/a. ____

Anexo B.***Constancias de autorización de uso de datos***

Autorización de la utilización de datos recolectados

Yo portador de la cedula de identidad....., autorizo la utilización de las señales encefalográficas recolectadas para fines científicos y desarrollo del proyecto de titulación: *DISEÑO DE UN PROTOTIPO DE MONITOREO DEL ESTADO DE SOMNOLENCIA EN CONDUCTORES MEDIANTE EL ESTUDIO DE LA ACTIVIDAD CEREBRAL EN LA ONDAS ALFA Y THETA.*

Anexo C: CODIFICACIÓN

Adquisición de datos EEG en Arduino

```

UNIVERSIDAD TECNICA DEL NORTE
ALEXANDER ICHAU
LECTURA E IMPRESION DE DATOS DE : atencion,delta,theta,low alfa,high
alfa,low beta,high beta, low gamma, midd gamma
*/
//CONEXION BT ENTRE HC-05 CON NEUROSKY Y CONEXION SERIAL CON MODULO
ARDUINO
//DEFINIR LA TASA DE BAUDIOS
#define BAUDRATE 57600
#define DEBUGOUTPUT 0
//DEFINIR VARIABLES PARA EL PAQUETE THINKGEAR
byte payloadChecksum = 0, generatedChecksum = 0, checksum = 0;
int payloadLength = 0;
byte payloadData[64] = {0};
byte poorQuality = 0, attention = 0, meditation = 0, ondas = 0;
long lastReceivedPacket = 0;
boolean bigPacket = false, brainwave = false;
unsigned int delta_wave = 0, theta_wave = 0, low_alpha_wave = 0,
high_alpha_wave = 0,
        low_beta_wave = 0, high_beta_wave = 0, low_gamma_wave = 0,
mid_gamma_wave = 0;
void setup() {
    Serial.begin(BAUDRATE);
    pinMode(LED, OUTPUT);
}

byte ReadOneByte() {
    while(!Serial.available());
    byte ByteRead = Serial.read();
    #if DEBUGOUTPUT
        Serial.print((char)ByteRead);
    #endif
    return ByteRead;
}

void read_waves(int i) {
    delta_wave = ((payloadData[i] << 16) + (payloadData[i+1] << 8) +
payloadData[i+2]);
    i+=3;
    theta_wave = ((payloadData[i] << 16) + (payloadData[i+1] << 8) +
payloadData[i+2]);
    i+=3;
    low_alpha_wave = ((payloadData[i] << 16) + (payloadData[i+1] << 8) +
payloadData[i+2]);
    i+=3;
    high_alpha_wave = ((payloadData[i] << 16) + (payloadData[i+1] << 8) +
payloadData[i+2]);
    i+=3;
    low_beta_wave = ((payloadData[i] << 16) + (payloadData[i+1] << 8) +
payloadData[i+2]);
    i+=3;
    high_beta_wave = ((payloadData[i] << 16) + (payloadData[i+1] << 8) +
payloadData[i+2]);
    i+=3;
    low_gamma_wave = ((payloadData[i] << 16) + (payloadData[i+1] << 8) +
payloadData[i+2]);
    i+=3;
    mid_gamma_wave = ((payloadData[i] << 16) + (payloadData[i+1] << 8) +
payloadData[i+2]);
}

int read_3byte_int(int i) {
    return ((payloadData[i] << 16) + (payloadData[i+1] << 8) +
payloadData[i+2]);
}

void loop() {
    ad_datos();
    imp_datos();
}

/*
Metodo en donde se realiza la lectura de los datos en formato THINKGEAR
con su estructura
y se utiliza los codigos para activar el envio de ondas eeg con el
0x83
*/
void ad_datos(){
    if(ReadOneByte() == 170 && ReadOneByte() == 170) {
        payloadLength = ReadOneByte();
        if(payloadLength > 169) return;
        payloadChecksum = 0;
        for(int i = 0; i < payloadLength; i++) {
            payloadData[i] = ReadOneByte();
            payloadChecksum += payloadData[i];
        }
        checksum = ReadOneByte();
        payloadChecksum = 255 - payloadChecksum;
        if(checksum == payloadChecksum) {
            poorQuality = 200;
            attention = 0;
            meditation = 0;
            ondas = 0;
        }
        brainwave=false;
        for(int i = 0; i < payloadLength; i++) { // inicializa los tipos de
datos que se van a obtener
            switch (payloadData[i]) {

```

```

        case 2: // valor de [CODE] es 0x02 envia el valor de calidad de
la señal (0-255)
            i++;
            poorQuality = payloadData[i];
            bigPacket = true;
            break;
        case 4: // valor eSense de atencion (0-100)
            i++;
            attention = payloadData[i];
            break;
        case 5: // valor eSense de meditacion (0-100)
            i++;
            meditation = payloadData[i];
            break;
        case 0x80:
            i = i + 3;
            break;
        case 0x83: // activacion de envio de ondas EEG
(delta,theta,alpha,beta,gaamma)
            i++;
            brainwave = true;
            byte vlen = payloadData[i];
            //mySerial.print(vlen, DEC);
            //mySerial.println();
            read_waves(i+1);
            i += vlen; // i = i + vlen
            break;
            } // switch
    } // for loop
} }
/*
Se crea el metodo en el cual se imprime los valores de los datos
obtenidos por el sensor EEG
*/
void imp_datos(){
if(brainwave && attention > 0 && attention < 100) { //procede a leer
los datos si las variables se encuentran en los rangos establecidos
// Datos enviados por serial.print
Serial.print("Atencion:");
Serial.print(attention, DEC);//imprime datos de nivel de atencion
eSense en formato decimal
Serial.print("\t");
Serial.print("Calidad sign: ");
Serial.print(poorQuality );//imprime datos de nivel de calidad de
señal en formato decimal
Serial.print("\t");
Serial.print(" Onda Delta ");
Serial.print(delta_wave, DEC);//imprime datos de onda delta en
formato decimal

```

```

Serial.print("\t");
Serial.print(" Theta: ");
Serial.print(theta_wave, DEC);//imprime datos de onda theta en
formato decimal
Serial.print("\t");
Serial.print(" Low Alpha: ");
Serial.print(low_alpha_wave, DEC);//imprime datos de onda low alpha
en formato decimal
Serial.print("\t");
Serial.print(" High Alpha: ");
Serial.print(high_alpha_wave, DEC); //imprime datos de high alpha en
formato decimal
Serial.print("\t");
Serial.print(" low beta:"); //estado vigila1 //imprime datos de onda
low beta en formato decimal
Serial.print(low_beta_wave, DEC);
Serial.print("\t");
Serial.print(" high beta: "); //estado vigila2 ;imprime datos de onda
high beta en formato decimal
Serial.print(high_beta_wave, DEC);
Serial.print("\t");
Serial.print("low gamma:");
Serial.print(low_gamma_wave, DEC); //imprime datos de onda low gamma
en formato decimal
Serial.print("\t");
Serial.print(" mid gamma:");
Serial.println(mid_gamma_wave, DEC); //imprime datos de onda mid
gamma en formato decimal
Serial.print("\n");
}
}

```

Código de Random Forest en Google Colab (Python)

```

#ALGORITMO RANDOM FOREST PARA SOMNOLENCIA DONDE SOMNOLENCIA 1 VIGILA 0
#SEÑALES TRATADES DE ATTENTION, THETA, LOW ALPHA, HIGH ALPHA, LOW BETHA, HIGH BETA,
#LOW GAMM MEDD GAMMA ETIQUETA
#LIBRERIAS COMPLEMENTARIAS PARA FUNCIONAMIENTO DE RANDOM FOREST ENSABLET MODEL
import pandas as pd
import numpy as np
import math as ma
from matplotlib import pyplot as pyplot
from sklearn.tree import plot_tree
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
#IMPRIMIR CANTIDAD DE MUESTRAS
print('# muestras en y ', np.bincount(y))
print('# muestras en y_train ', np.bincount(y_train))
print('# muestras en y_test ', np.bincount(y_test))
#Llamar dependencia para algoritmo de clasificacion
from sklearn.ensemble import RandomForestClassifier
#CONFIGURAR PARAMETROS PARA GENERAR EL ALGORITMO
rf= RandomForestClassifier(n_estimators=481,
                           n_jobs=-1,
                           criterion='gini',
                           max_features='sqrt',
                           max_depth=18,
                           random_state=1)

rf.fit(X_train,y_train)
#realizar pruebas de prediccion
y_prds=rf.predict(X_test)
comp=pd.DataFrame({'real':y_test, 'prediccion':y_prds})
y_prds
print(y_prds)
#CALCULOS DE PRECISION DE MODELO
from sklearn.metrics import accuracy_score
acc=accuracy_score(y_test,y_prds)
acc
score=rf.score(X_test,y_test)
print(score)
#GENERAR MATRIZ DE CONFUSION
cm= confusion_matrix(y_test, y_prds)
cm
#IMPRESION DE METRICAS DE ALGORITMO
from sklearn.metrics import classification_report
print(classification_report(y_prds, y_test))
#GRAFICAR ARBOL DE TOMA DE DESICIONES DEL ALGORITMO
estimator=rf.estimators_[1]
from sklearn.tree import export_graphviz
export_graphviz(estimator, out_file='tree.dot',
                 rounded = True, proportion = False,
                 precision = 2, filled = True)
from subprocess import call
call(['dot', '-Tpng', 'tree.dot', '-o', 'tree.png', '-Gdpi=600'])
from IPython.display import Image
Image(filename = 'tree.png')

```


Código final del prototipo para implementar en el Raspberry Pi

```

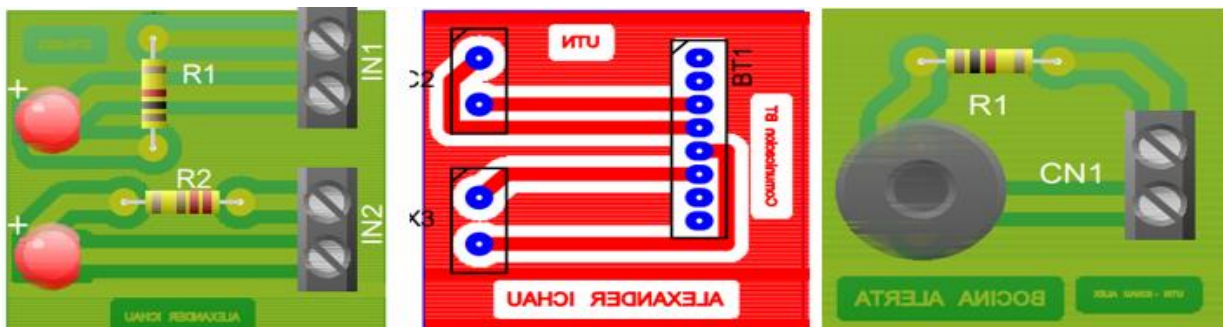
# ALEXANDER ICHAU-UTN
# Código final en donde se tiene la adquisición de datos EEG, filtrados de estos
# datos, implementación de algoritmo de clasificación random forest ya entrenado y
# predicción final del estado del conductor.
import serial
import csv
import numpy as np
import matplotlib.pyplot as plt
import pickle # Importar la librería pickle para cargar el modelo
import RPi.GPIO as GPIO # importar librería RPi.GPIO

# Configurar la conexión serial con los mismos parámetros que en Arduino
ser = serial.Serial(
    "/dev/ttyACM0", 57600
) # Cambia '/dev/ttyACM0' por el puerto serial correcto en tu sistema
# Configurar los parámetros del filtro Savitzky-Golay
window_length = 3 # Longitud de la ventana del filtro
polyorder = 2 # Orden del polinomio para ajusta
# Cargar el modelo de Random Forest previamente entrenado desde un archivo .sav
model = pickle.load(
    open("modelofinalizado.sav", "rb")
) # Cambia 'random_forest_model.sav' por el nombre de tu archivo .sav
# configurar el pin 18,16,15 de la raspberry como salida
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(15, GPIO.OUT) # alarma sonora
GPIO.setup(16, GPIO.OUT) # led rojo de alarma visual
GPIO.setup(18, GPIO.OUT) # led indicador estado vigila del usuario
# Crear un buffer para almacenar los valores leídos
buffer = np.zeros((window_length, 8))
# Leer los datos enviados desde Arduino y hacer predicciones en tiempo real?
while True:
    # Leer una línea de datos desde la conexión serial
    data = ser.readline().decode("utf-8").strip()
    # Separar los datos por los caracteres de tabulación ('\t')
    values = data.split("\t")
    # Si se recibieron todos los valores esperados
    if len(values) == 8:
        attention = int(values[0])
        theta_wave = int(values[1])
        low_alpha_wave = int(values[2])
        high_alpha_wave = int(values[3])
        low_beta_wave = int(values[4])
        high_beta_wave = int(values[5])
        low_gamma_wave = int(values[6])
        mid_gamma_wave = int(values[7])
        # Agregar los valores al buffer
        buffer[-1, :] = buffer[1, :]
        buffer[-1, :] = [
            attention,
            theta_wave,
            low_alpha_wave,
            high_alpha_wave,
            low_beta_wave,
            high_beta_wave,
            low_gamma_wave,
            mid_gamma_wave,
        ]
        # Aplicar el filtro Savitzky-Golay a los valores del buffer
        smoothed_values = np.zeros(8)
        for i in range(8):
            smoothed_values[i] = np.round(
                np.median(
                    np.polynomial.polynomial.polyval(
                        np.linspace(0, window_length - 1, window_length),
                        np.polynomial.polynomial.polyfit(
                            np.linspace(0, window_length - 1, window_length),
                            buffer[:, i],
                            polyorder,
                        ),
                    ),
                    1,
                )
            )
        # Hacer predicción con el modelo de Random Forest
        prediction = model.predict(
            [smoothed_values]
        ) # Hacer la predicción con los valores suavizados
        print("Predicción: ", prediction) # Imprimir la predicción en consola
        # Ejecutar acción dependiendo su estado
        if (
            prediction == 0
        ): # si el usuario esta en estado despierto se encendera el pin 18
            # mientras que los pines 15 y 16 estaran off
            print("DESPIERTO")
            GPIO.output(18, GPIO.HIGH)
            GPIO.output(15, GPIO.LOW)
            GPIO.output(16, GPIO.LOW)
        else:
            GPIO.output(18, GPIO.LOW)
            if (
                prediction == 1
            ): # si el usuario esta en estado de somnolencia se encenderan los pines
                # 15 y 16 de alarma mientras que el pin 18 modo off
                print("ALERTA...")
                GPIO.output(15, GPIO.HIGH)
                GPIO.output(16, GPIO.HIGH)
                GPIO.output(18, GPIO.LOW)

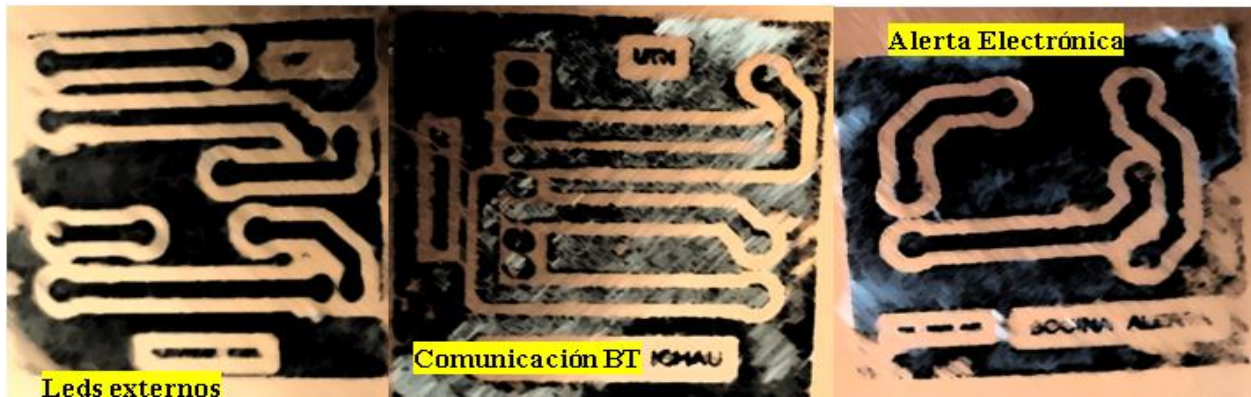
```

Anexo D. Elaboración de placas

Diagramas



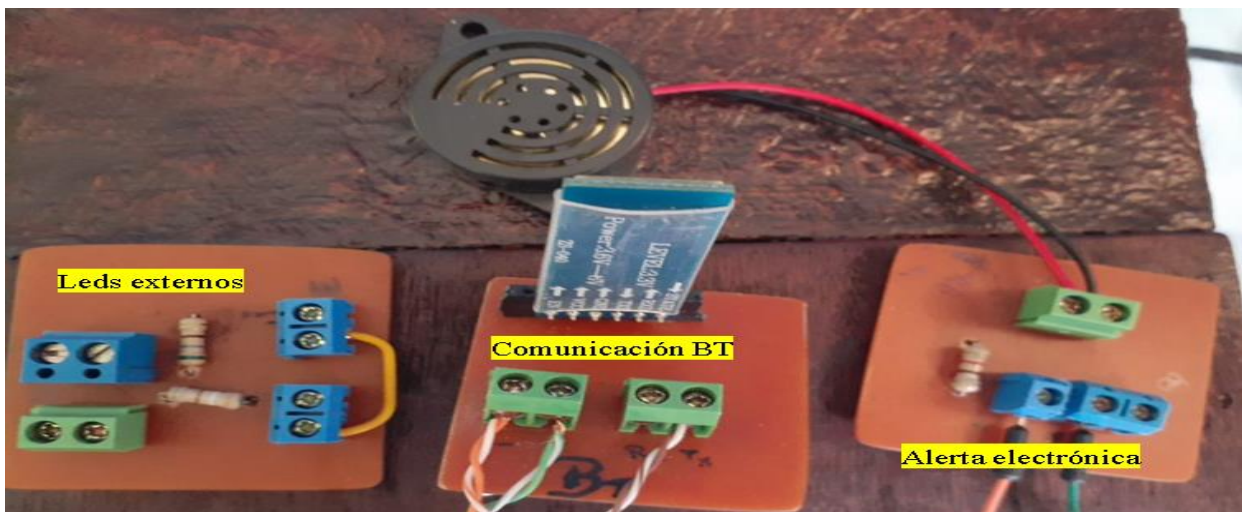
Planchado de placas



Suelda de placas



Placas finales



Anexo E. Fotografías de lo realizado

- Elemento 1. Sensor principal captador de ondas EEG neurosky mindwave mobile2



- Elemento 2. Módulo de comunicación Bluetooth hc-05 el cual conecta de manera inalámbrica con el elemento 1 y a su vez está conectada al elemento 3 por su pin de Tx al pin de Rx del elemento 3.



- Elemento 3. Placa de desarrollo de adquisición de datos EEG Arduino mega.

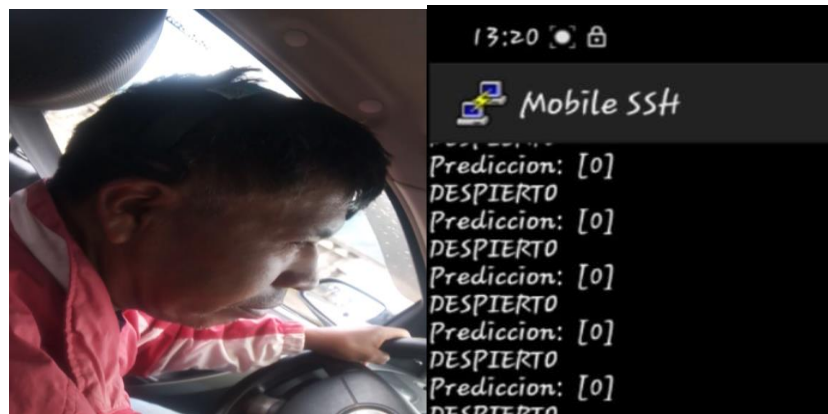


- Elemento 4. Unidad de procesamiento de los datos placa Raspberry pi 3 B+

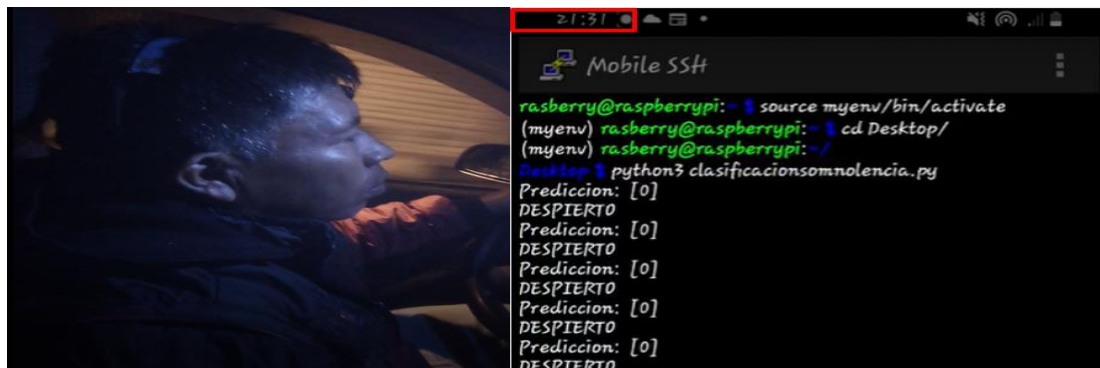
- Automóvil utilizado para las respectivas pruebas del prototipo

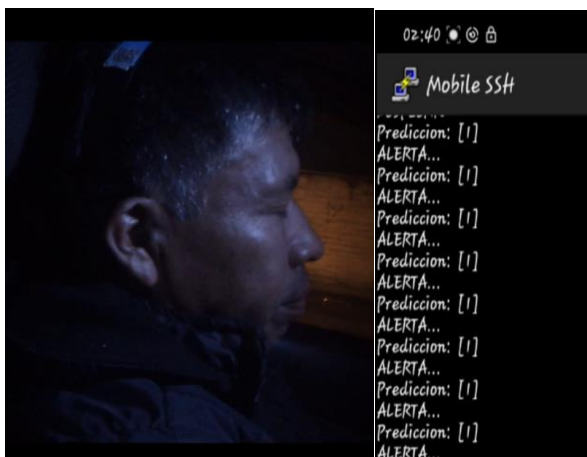


- Resultados de las pruebas realizadas en los diferentes escenarios planteados
 - o Escenario horario día de 13:00 pm a 15:00 pm

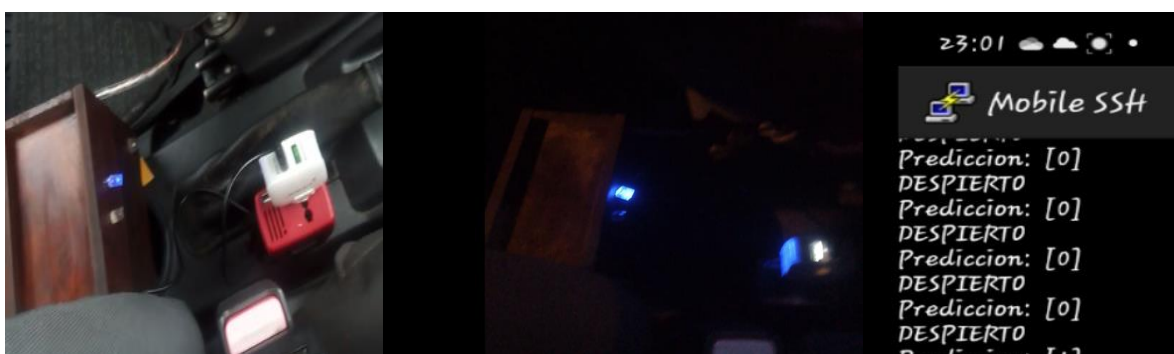


- o Escenarios en los horarios noche de 19:00 pm a 23:00 y de 23:30pm a 3:00 am

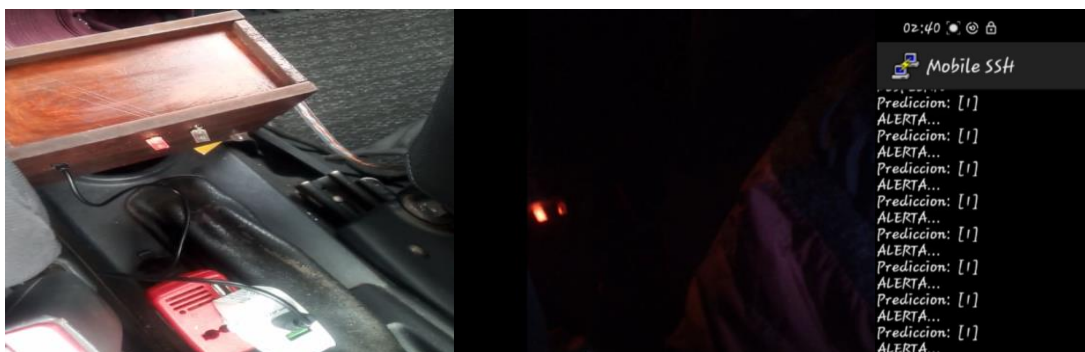




- Cuando la etiqueta de predicción se encuentra en 0 el led azul se enciende



- Cuando la etiqueta de predicción se encuentra en 1 el led rojo se enciende en conjunto con la alarma sonora que alertará al conductor.



Anexo G . Datasheet de dispositivos

Mindwave Mobile 2

Las especificaciones del sensor se las puede obtener ingresando a la pagina principal Neurosky o a su vez en el siguiente enlace:

<https://cdn.sparkfun.com/assets/0/6/a/8/9/MWMobile2TransitionDoc.pdf>

Bluetooth hc-05

Las especificaciones del producto se las puede obtener ingresando en el siguiente enlace:

<https://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf>

Arduino mega

Las especificaciones técnicas del producto se las puede obtener en la página principal de Arduino o en el siguiente enlace:

<https://docs.arduino.cc/static/b041a982c2b71c2b39349c223c58d72e/A000067-datasheet.pdf>

Raspberry Pi 3B+

El datasheet del dispositivo se encuentra disponible en la página principal de Raspberry o en el siguiente enlace: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>