



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE TELECOMUNICACIONES

Tema:

**DETECCIÓN DE NIVELES DE ESTRÉS EN ESTUDIANTES
UNIVERSITARIOS A PARTIR DE SEÑALES ELECTROFISIOLÓGICAS
MEDIANTE TÉCNICAS DE MACHINE LEARNING**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniera en Telecomunicaciones

ÁREA: Electrónica

LÍNEA DE INVESTIGACIÓN: Tecnología de la información y Sistemas de control

AUTOR: Shirley Lizbeth Pazmiño de la Cruz

TUTOR: Ing. Edgar Patricio Córdova Córdova, Mg.

Ambato - Ecuador

febrero – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: DETECCIÓN DE NIVELES DE ESTRÉS EN ESTUDIANTES UNIVERSITARIOS A PARTIR DE SEÑALES ELECTROFISIOLÓGICAS MEDIANTE TÉCNICAS DE MACHINE LEARNING, desarrollado bajo la modalidad Proyecto de Investigación por la señorita Shirley Lizbeth Pazmiño de la Cruz, estudiante de la Carrera de Ingeniería en Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024.

Ing. Edgar Patricio Córdova Córdova, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: DETECCIÓN DE NIVELES DE ESTRÉS EN ESTUDIANTES UNIVERSITARIOS A PARTIR DE SEÑALES ELECTROFISIOLÓGICAS MEDIANTE TÉCNICAS DE MACHINE LEARNING, es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024



Shirley Lizbeth Pazmiño de la Cruz

C.C 1850034057

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2024.



Shirley Lizbeth Pazmiño de la Cruz

C.C. 1850034057

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por la señorita Shirley Lizbeth Pazmiño de la Cruz, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado DETECCIÓN DE NIVELES DE ESTRÉS EN ESTUDIANTES UNIVERSITARIOS A PARTIR DE SEÑALES ELECTROFISIOLÓGICAS MEDIANTE TÉCNICAS DE MACHINE LEARNING, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Ana Pamela Castro Martín, MSc
PROFESOR CALIFICADOR

Ing, Clara Augusta Sánchez Benítez, Mg
PROFESOR CALIFICADOR

DEDICATORIA

Este trabajo de titulación está dedicado con profundo agradecimiento y amor a mi madre, Anita quien ha sido el pilar esencial de mi vida. Su incansable esfuerzo, sacrificio y apoyo incondicional han sido la fuente que ha guiado mi camino a lo largo de mi vida y travesía académica.

Shirley Lizbeth Pazmiño de la Cruz

AGRADECIMIENTO

A Dios, fuente de sabiduría y guía, por haberme acompañado en este viaje académico.

A mi familia, a mi mamá, Anita; a mi papá, Ángel; y a mis hermanos Darío, David, y especialmente a Wagner quienes han sido mi compañía constante y la razón que me impulsa a seguir avanzando.

A Daniel, su compañía, apoyo incondicional y ánimos fueron una fuente inagotable de motivación que me impulsaron a no desistir en mi carrera universitaria.

Mi reconocimiento especial al Ing. Patricio Córdova, por su valioso tiempo, las ideas sugeridas y su orientación durante el desarrollo de este proyecto.

También quiero expresar mi gratitud a mis amigos Isa y Palan, por estar presente en los últimos semestres y compartir risas y experiencias. Y un agradecimiento especial a mi amigo Ing. Jona, quien estuvo siempre a mi lado, alentándome a alcanzar mis metas.

Shirley Lizbeth Pazmiño de la Cruz

ÍNDICE GENERAL DE CONTENIDOS

PORTADA	i
APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS	xii
ÍNDICE DE FIGURAS	xv
ÍNDICE DE ANEXOS	xx
RESUMEN EJECUTIVO	xxi
ABSTRACT	xxii
CAPÍTULO I. MARCO TEÓRICO	1
1.1 Tema de investigación	1
1.1.1 Planteamiento del problema	1
1.2 Antecedentes investigativos	2
1.3 Fundamentación teórica	4

1.3.1 Estrés.....	4
1.3.2 Estrés en estudiantes universitarios.....	9
1.3.3 Señales electrofisiológicas	13
1.3.4 Actividad electrodérmica (EDA)	14
1.3.5 Electromiografía.....	15
1.3.6 Sistemas electrónicos	17
1.3.7 Inteligencia artificial	19
1.3.8 Machine Learning	19
1.3.9 Aplicación móvil.....	25
1.3.10 Base de datos.....	28
1.3.11 Cloud Computing.....	28
1.4 Objetivos	29
1.4.1 Objetivo general.....	29
1.4.2 Objetivos específicos	29
CAPÍTULO II. METODOLOGÍA	30
2.1 Materiales.....	30
2.2 Métodos.....	31

2.2.1 Modalidad de la investigación	31
2.2.2 Población y muestra	31
2.2.3 Recolección de información.....	32
2.2.4 Procesamiento y análisis de datos	32
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	34
3.1 Análisis y discusión de los resultados	34
3.2 Desarrollo de la propuesta.....	34
3.2.1 Requerimientos del sistema.....	34
3.2.2 Selección del hardware	35
3.2.3 Selección de tecnologías	44
3.2.4 Arquitectura del sistema.....	50
3.2.5 Implementación del circuito de monitoreo de electro señales	51
3.2.6 Implementación del modelo de machine Learning	60
3.2.7 Desarrollo de la aplicación móvil	85
3.2.8 Registro y almacenamiento de datos.....	96
3.2.9 Pruebas de funcionamiento	106
3.2.10 Autonomía del sistema.....	113
3.2.11 Presupuesto	114
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	116
4.1 Conclusiones	116
4.2 Recomendaciones.....	117

REFERENCIAS BIBLIOGRÁFICAS.....	119
ANEXOS.....	126

ÍNDICE DE TABLAS

Tabla 1. Respuesta del organismo por estrés	7
Tabla 2. Estresores académicos.....	10
Tabla 3. Indicadores del estrés académico	11
Tabla 4. Escala de valores del inventario SISCO	12
Tabla 5. Señales electrofisiológicas	14
Tabla 6. Características de los sensores	18
Tabla 7. Tipos de inteligencia artificial.....	19
Tabla 8. Algoritmos de Aprendizaje automático	20
Tabla 9. Partes del machine Learning	22
Tabla 10. Proceso de Machine Learning.....	22
Tabla 11. Estándares para el diseño de una app móvil	28
Tabla 12. Característica del Hardware seleccionado	30
Tabla 13. Características del Software seleccionado	30
Tabla 14. Fórmula para el tamaño muestra.	32
Tabla 15. Recolección de información.....	32
Tabla 16. Parámetros técnicos del muscle sensor V3	36
Tabla 17. Parámetros técnicos del Myoware Muscle.....	37
Tabla 18. Parámetros técnicos del Sensor EMG SEN0240	38
Tabla 19. Tabla comparativa entre sensores EMG.	38

Tabla 20. Parámetros técnicos del Grove GSR Sensor	39
Tabla 21. Parámetros técnicos del Sensor CJMCU-6701	40
Tabla 22. Parámetros técnicos del Techtonics GSR	40
Tabla 23. Tabla comparativa entre sensores GSR	41
Tabla 24. Tabla comparativa entre microcontroladores.....	43
Tabla 25. Comparación de baterías para sensor EMG	44
Tabla 26. Comparación de baterías para la placa.....	44
Tabla 27. Tabla comparativa entre tecnologías de comunicación inalámbrica	45
Tabla 28. Tabla comparativa entre protocolos de comunicación.....	46
Tabla 29. Comparación entre lenguajes de programación	47
Tabla 30. Tabla comparativa entre bases de datos	48
Tabla 31. Comparativa de servicios de almacenamiento en la nube.....	48
Tabla 32. Tabla comparativa de los métodos de entrenamiento	49
Tabla 33. Rangos de las mediciones GSR y EMG.....	61
Tabla 34. Clase y etiquetas de los datos de EMG y GSR	62
Tabla 35. Preguntas escogidas del formulario SISCO	62
Tabla 36. Rangos de los resultados del inventario SISCO.....	68
Tabla 37. Codificación de las respuestas a valores numéricos	68
Tabla 38. Clase y Etiquetado de datos	68
Tabla 39. Clases de la base de datos	69
Tabla 40. Desempeño de los algoritmos de clasificación	74

Tabla 41. Resultado aplicando la técnica de validación cruzada	77
Tabla 42. Cumplimiento de los estándares de diseño para app.....	88
Tabla 43. Prueba fuera de la época de evaluaciones	108
Tabla 44. Prueba bajo estresores académicos	110
Tabla 45. Cálculo de error del sensor GSR.....	111
Tabla 46. Cálculo de error del sensor EMG.....	111
Tabla 47. Resultado prueba de consistencia de datos GSR	112
Tabla 48. Resultado prueba de consistencia de datos EMG	112
Tabla 49. Potencia de la placa.....	113
Tabla 50. Potencia del sensor EMG.....	113
Tabla 51. Potencia de las baterías	113
Tabla 52. Presupuesto de implementación del prototipo	114

ÍNDICE DE FIGURAS

Figura 1. Secuencia del estrés	5
Figura 2. Recepción del factor estresante	6
Figura 3. Programación de la respuesta al estrés	6
Figura 4. Respuesta del organismo	7
Figura 5. Síndrome general de adaptación.....	8
Figura 6. Random Forest.....	22
Figura 7. Estructura de la matriz de confusión	23
Figura 8. Esquema básico de la arquitectura del sistema.....	34
Figura 9. Muscle Sensor V3.....	36
Figura 10. Myoware Muscle Sensor	37
Figura 11. Sensor EMG SEN0240.....	37
Figura 12. Grove – GSR Sensor.....	39
Figura 13. Skin Sensor Module CJMCU 6701	40
Figura 14. Techtonics GSR.....	40
Figura 15. Esp8266 NodeMCU.....	42
Figura 16. ESP32 Tiny S3.....	42
Figura 17. ESP32 NodeMCU.....	42
Figura 18. Esquema general de la arquitectura del sistema	51
Figura 19. Colocación de los electrodos para EMG	52
Figura 20. Diagrama de conexión del sensor EMG	52

Figura 21. Colocación de los electrodos para GSR.....	53
Figura 22. Diagrama de conexión del sensor GSR	54
Figura 23. Circuito electrónico del sistema.....	54
Figura 24. Diseño esquemático del circuito	55
Figura 25. Diseño del PCB.....	56
Figura 26. Impresión del PCB con CNC.....	56
Figura 27. Impresión de las pistas del PCB	56
Figura 28. Diseño de la caja contenedora	57
Figura 29. Circuito integrado en el case.....	57
Figura 30. Diagrama de flujo del sensor EMG	58
Figura 31. Diagrama de flujo del sensor GSR	59
Figura 32. Diagrama del proceso y transmisión de datos	60
Figura 33. Resultados del formulario SISCO primera pregunta	63
Figura 34. Resultados del formulario SISCO segunda pregunta	64
Figura 35. Resultados del formulario SISCO tercera pregunta.....	64
Figura 36. Resultados del formulario SISCO cuarta pregunta.....	65
Figura 37. Resultados del formulario SISCO quinta pregunta	66
Figura 38. Resultados del formulario SISCO sexta pregunta	66
Figura 39. Resultados del formulario SISCO séptima pregunta	67
Figura 40. Resultados del formulario SISCO octava pregunta	68
Figura 41. Carga de los datos desde el archivo CSV	69

Figura 42. Separación del dataframe.....	70
Figura 43. Estructura del método train test Split [65].....	71
Figura 44. Selección de método Resampling	71
Figura 45. Algoritmos seleccionados	72
Figura 46. Flujograma de la técnica de validación cruzada basado en [66]	72
Figura 47. Estructura de la matriz de confusión basado en [47].....	73
Figura 48. Matrices de confusión de cada algoritmo	74
Figura 49. Precisión de entrenamiento, testeo y prueba.....	76
Figura 50. Desempeño del algoritmo	77
Figura 51. Precisión sobre el entrenamiento del algoritmo.....	78
Figura 52. Entrenamiento del algoritmo Random Forest.....	78
Figura 53. Ajuste “gini” Random Forest con 25 estimadores.....	79
Figura 54. Ajuste “entropy” Random Forest con 25 estimadores.....	80
Figura 55. Ajuste “entropy” Random Forest con 50 estimadores.....	80
Figura 56. Ajuste Random Forest con 60 estimadores	81
Figura 57. Ajuste Random Forest con 75 estimadores	81
Figura 58. Ajuste Random Forest con 100 estimadores	81
Figura 59. Región de decisión.....	82
Figura 60. Importancia de los pesos de las variables con Random Forest.....	82
Figura 61. Pesos de las variables.....	83
Figura 62. Mapa de calor de la matriz de correlación.....	83

Figura 63. Matriz de correlación.....	83
Figura 64. Aplicación del modelo Random Forest	84
Figura 65. Sistema operativo para la instalación de flutter	85
Figura 66. Descarga de Flutter desde el sitio web oficial	85
Figura 67. Variables de entorno para la instalación de Flutter	86
Figura 68. Edición de la variable path para la instalación de Flutter.....	86
Figura 69. Verificación de la instalación de Flutter.....	87
Figura 70. Instalación de la extensión de Flutter	87
Figura 71. Instalación de la extensión de Dart.....	87
Figura 72. Paleta de colores para aplicación de salud.....	88
Figura 73. Interfaz gráfica de inicio.....	89
Figura 74. Controladores de texto.....	90
Figura 75. Conexión con el servidor.....	90
Figura 76. Interfaz gráfica de la ventana registro	90
Figura 77. Validación de usuario	91
Figura 78. Interfaz gráfica de la ventana de login.....	91
Figura 79. Creación de los dashboard.....	92
Figura 80. Actualización de datos en tiempo real	92
Figura 81. Interfaz gráfica de la ventana de Dashboard.....	92
Figura 82. Creación de la ventana de dashboard.....	93
Figura 83. Interfaz gráfica de la ventana de formulario.....	94

Figura 84. Creación de la ventana resultado	94
Figura 85. Interfaz gráfica de la ventana de resultados.....	95
Figura 86. Interfaz gráfica de la ventana de reporte.....	95
Figura 87. Arquitectura API.....	96
Figura 88. Esquema general del almacenamiento de datos.....	97
Figura 89. Diagrama de las relaciones de la base de datos	97
Figura 90. Base de datos para el almacenamiento de información	98
Figura 91. Tabla de registro de usuarios	98
Figura 92. Tabla del registro del formulario	99
Figura 93. Tabla de registro de los sensores.	99
Figura 94. Tabla de resultados del sistema	100
Figura 95. Registro en 000WebHost.....	105
Figura 96. Creación del sitio de alojamiento web.....	105
Figura 97. Sincronización del servidor con la base de datos	106
Figura 98. Sujeto 1 colocación de sensores	107
Figura 99. Prueba de funcionamiento	107
Figura 100. Lectura de GSR y EMG del sujeto 1	108
Figura 101. Resultados de la aplicación móvil	108
Figura 102. Prueba de funcionamiento del sujeto 2.....	109
Figura 103. Resultados bajo estresores académicos	109

ÍNDICE DE ANEXOS

Anexo A. Instrumento del Inventario SISCO SV-21	126
Anexo B. Certificado de validación del instrumento	128
Anexo C. Procesamiento del instrumento para estrés académico	129
Anexo D. Datasheet ESP32 Tiny S3.....	132
Anexo E. Datasheet sensor GSR.....	133
Anexo F. Datasheet Muscle Sensor V3.....	134
Anexo G. Muestra del Dataset usado para el entrenamiento	135
Anexo H. Código de la conexión del microcontrolador	136
Anexo I. Certificado de validación del dispositivo.....	139
Anexo J. Manual de Usuario.....	140
Anexo K. Pruebas de funcionamiento del sistema.....	144

RESUMEN EJECUTIVO

Al iniciar sus estudios superiores, los jóvenes se ven sometidos a presiones académicas, expectativas sociales y múltiples responsabilidades, generando niveles elevados de estrés que inciden tanto en su rendimiento académico como en su salud mental y bienestar general. Este proyecto de investigación ha perseguido el desarrollo de un sistema electrónico que emplee señales electrofisiológicas y técnicas de machine learning para identificar niveles de estrés en estudiantes universitarios en este entorno desafiante.

Para alcanzar dicho objetivo, se ha llevado a cabo una revisión bibliográfica exhaustiva en bases de datos médicos y científicos, focalizándose en señales electrofisiológicas asociadas al estrés académico. Se han seleccionado sensores específicos, como el Grove GSR y Muscle Sensor V3, y todo el proceso se ha desarrollado en el microcontrolador ESP32 Tiny S3, aprovechando su capacidad y tamaño para crear un dispositivo portátil. Los datos procesados se almacenaron en un servidor web mediante peticiones PHP. Las métricas de evaluación del algoritmo de machine learning revelaron una alta precisión del 0.96, respaldada por un índice de 0.97 en el conjunto de entrenamiento, 0.97 en el conjunto de prueba y 0.9 en la etapa de validación con el método de validación cruzada. Estos resultados confirman que el algoritmo Random Forest gestiona de manera eficiente la clasificación y el entrenamiento, sobresaliendo en comparación con otros algoritmos evaluados. En cuanto a la confiabilidad, el sistema ha logrado un porcentaje del 93,68% en la lectura de datos, consolidando así su capacidad para detectar niveles de estrés y prevenir posibles colapsos en la salud de los estudiantes.

Palabras clave: Aprendizaje automático, estrés académico, GSR, EMG

ABSTRACT

At the beginning of their higher education journey, young individuals face academic pressures, social expectations, and multiple responsibilities, resulting in heightened stress levels that impact both their academic performance and overall mental health and well-being. This research project has aimed to develop an electronic system utilizing electrophysiological signals and machine learning techniques to identify stress levels in university students within this challenging environment.

To achieve this goal, an exhaustive literature review has been conducted in medical and scientific databases, focusing on electrophysiological signals associated with academic stress. Specific sensors, such as Grove GSR and Muscle Sensor V3, have been selected, and the entire process has been implemented on the ESP32 Tiny S3 microcontroller, leveraging its capacity and size to create a portable device. Processed data has been stored on a web server through PHP requests.

Evaluation metrics of the machine learning algorithm revealed a high precision of 0.96, supported by an index of 0.97 in the training set, 0.97 in the test set, and 0.9 in the validation stage with cross-validation. These results confirm that the Random Forest algorithm efficiently manages classification and training, outperforming other evaluated algorithms. Regarding reliability, the system has achieved a percentage of 93.68% in data readings, thus solidifying its ability to detect stress levels and prevent potential health collapses in students.

Keywords: Machine Learning, academic stress, GSR, EMG

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

DETECCIÓN DE NIVELES DE ESTRÉS EN ESTUDIANTES UNIVERSITARIOS A PARTIR DE SEÑALES ELECTROFISIOLÓGICAS MEDIANTE TÉCNICAS DE MACHINE LEARNING

1.1.1 Planteamiento del problema

El estrés ha sido reconocido como una de las principales preocupaciones de salud en la sociedad moderna. Este fenómeno, ampliamente conocido como la "enfermedad del siglo XXI", se atribuye a los cambios en los estilos de vida, la acelerada rutina diaria, factores como la falta de descanso adecuado y una alimentación deficiente pueden contribuir a la aparición de estrés, lo que a su vez aumenta el riesgo de desarrollar enfermedades, incluyendo problemas cardiovasculares [1].

La Organización Mundial de la Salud (OMS), informa que más de 450 millones de personas en todo el mundo tienen problemas de salud mental. El estrés tiene gran impacto en la población mundial, lo que significa que cualquier persona en cualquier parte del mundo puede experimentar altos niveles de estrés. De acuerdo con las estadísticas, aproximadamente 3 de cada 10 personas en el mundo manifiestan sentirse preocupadas (42%), estresadas (41%), con dolor físico (31%), tristes (28%) y con ira (23%) a diario [1].

En Latinoamérica, según una encuesta realizada por las Naciones Unidas, se obtuvo que el 52% ha manifestado un nivel de estrés alto y un 47% ha pasado por ataques de pánico o ansiedad, con respecto al número total de encuestados [2]. De acuerdo con investigaciones realizadas por la Escuela Latinoamericana de Medicina, en Latinoamérica predomina el estrés académico entre estudiantes universitarios, superando el 67% en la categoría de estrés moderado, respecto a la población del estudio [3].

En Ecuador, según los resultados de la Encuesta Nacional de Salud y Nutrición del año 2022, se ha observado que alrededor del 45% de la población ecuatoriana experimenta algún grado de estrés en algún momento de su vida. Este hallazgo resalta la prevalencia significativa del estrés en el país y subraya la importancia de comprender y abordar adecuadamente esta condición. Debido a que, cuando se experimenta de manera persistente o en niveles elevados, puede tener consecuencias negativas para la salud en general [4].

El estrés en estudiantes universitarios tiene origen debido a las demandas académicas a las que están expuestos. En el transcurso de la carrera universitaria, los estudiantes deben soportar tensiones psicológicas que en muchas ocasiones se convierten en estrés, el cual puede producir síntomas físicos o psicológicos, entre los síntomas físicos se tienen: fatiga, cefaleas, problemas de digestión, tensión muscular y en los síntomas psicológicos se puede desarrollar desconcentración, problemas de memoria, ansiedad, depresión, bloqueos mentales [5].

1.2 Antecedentes investigativos

Mediante la bibliográfica basada en artículos científicos y trabajos de titulación publicados en los principales repositorios de universidades, se encuentran los siguientes antecedentes.

Víctor Tomalá de la Universidad Técnica de Ambato en el año 2018 desarrolló el tema “SISTEMA ELECTRÓNICO DE MONITOREO DE NIVELES DE ESTRÉS ACADÉMICO” presenta un sistema electrónico integrado con una tarjeta Arduino UNO y una Raspberry pi3 de bajo coste, donde se mide la frecuencia cardíaca y la temperatura como datos fisiológicos que posteriormente se almacenan en la base de datos MySQL, con dichos valores adquiridos en una interfaz se detecta el nivel de estrés académico que presenta una persona. Según el análisis de resultados de esta investigación, los niveles de estrés de los estudiantes en bajo nivel de tensión académica con respecto a un alto nivel de tensión académica, se puede observar una variación significativa, llegando así hasta un nivel de 10 con rango de 1-10, sabiendo que no se encuentra en estado crítico, ya que está en rangos normales de temperatura y frecuencia cardíaca y llegan hasta los niveles de alerta [6].

Carlos Navarro de España, en el año 2018 plantea el tema “DETECCIÓN DE LOS NIVELES DE ESTRÉS Y ANSIEDAD EN PILOTOS APLICANDO TÉCNICAS DE MACHINE LEARNING” en el cual realiza un estudio a partir de la respuesta galvánica de la piel para determinar la fase de estrés, esto se realiza a partir del sensor Grove GSR, con los datos obtenidos se ha implementado una herramienta de Machine Learning para obtener un modelo de predicción sobre los estados de estrés y ansiedad en el que se han analizado varios algoritmos de predicción. Se ha realizado una interfaz en Matlab para la aplicación de los procedimientos de Machine Learning en el cual se pueden visualizar en tiempo real los datos del sensor GSR y se logra determinar la fase de estrés. Como resultados de esta investigación se obtuvo que el algoritmo SVM y KNN son los más idóneos a la hora de determinar un modelo de predicción, debido a que los resultados superaron todos los algoritmos con el 70% en acierto respecto a la detección de los niveles de estrés [7].

David Sandoval Rodríguez en el año 2019 en la ciudad de Madrid desarrolló el tema de investigación “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA DETECCIÓN DEL ESTRÉS MEDIANTE REDES NEURONALES CONVOLUCIONALES A PARTIR DE IMÁGENES TÉRMICAS” con el que se obtuvo un sistema de detección de estrés a base de un análisis profundo de imágenes extraídas de videos térmicos, en el que se ha realizado un procesado de las imágenes para medir la evolución de la temperatura y cuantificar el nivel de estrés. Como modelo de aprendizaje automático se ha utilizado la arquitectura inception v3 que se basa en la red convolucional con tasas de aprendizaje cíclico. Como resultado de esta investigación se obtuvo que el modelo desarrollado para la detección de estrés cumple con las especificaciones planteadas por el proyecto con un bajo índice de error, al tener una precisión del 97% [8].

Andrés Navarro de Chile en el año 2021 presentó el tema de investigación “DESARROLLO DE UN SISTEMA WEARABLE PARA LA DETECCIÓN DE ESTRÉS MEDIANTE APRENDIZAJE AUTOMÁTICO BASADO EN RITMO CARDIACO”, desarrolló un dispositivo electrónico wearable como un clasificador de estados de estrés mediante herramientas de aprendizaje automático. El dispositivo se basa en la medición de señales de ECG ante variaciones de HR, con el análisis de variables de HR, Delta HR y Temperatura corporal con la ayuda del algoritmo KNN,

se obtuvo que el uso de la variable Delta HR es más efectiva para detectar el estrés. Este dispositivo procesa los datos obtenidos y la entrega de forma serial vía Wi-Fi con protocolo MQTT. Los principales resultados obtenidos de las pruebas realizadas indican que el escenario menos favorable ocurre con HR utilizando el clasificador Naive Bayes, donde su accuracy promedio corresponde a 61.25 [%] y su F1 a 62.87[%]. Mientras que los mejores resultados se obtienen con Delta HR, con un accuracy promedio de 70,25% y F1 de 70,74%. Por lo anteriormente dicho se puede concluir que el Delta HR es una variable fisiológica que por sí sola pudiera permitir la detección de cuadros de estrés [9].

Samia Tuttillo de la Universidad Tecnica del Norte en el año 2021, en su investigación con el tema “SISTEMA BIOFEEDBACK DE SENSORES DE DETECCIÓN DE ESTRÉS MEDIANTE ALGORITMOS DE APRENDIZAJE DE MÁQUINAS EN ESTUDIANTES UNIVERSITARIOS” con el que se elaboró un sistema Biofeedback de predicción de estrés en estudiantes universitarios, que se encuentra conformado por dos sistemas embebidos que son Arduino UNO y MySignals los cuales mediante filtros digitales procesan las bioseñales obtenidas por sensores biomédicos como Airflow y a través de aprendizaje automático con el algoritmo KNN determinan el nivel de estrés de los pacientes. Como resultados se obtuvo que el 30% de los sujetos de prueba resultaron con un nivel de estrés normal lo cual no es de alarmarse ya que sus respuestas fisiológicas se encuentran moderadas, el 70% restante presentó un incremento en sus respuestas fisiológicas, esto puede deberse a su ajeteo en la vida diaria provocada por sus diferentes estresores, este 70% de sujetos de prueba realizaron el respectivo Biofeedback lo cual ayuda a que sus respuestas fisiológicas se normalicen lo que conlleva a su relajación y disminución de nivel de estrés [10].

1.3 Fundamentación teórica

1.3.1 Estrés

El estrés es un proceso en el que las demandas ambientales comprometen o superan la capacidad adaptativa de un organismo, dando lugar a cambios biológicas y psicológicos que pueden situar a las personas en riesgo de enfermedad [11].

En esta definición se determinan tres aspectos importantes [12]:

- Demandas ambientales. Se refiere a los agentes estresantes.
- Una persona para evaluar su percepción de estrés.
- Respuestas de tipo afectivo, conductual o biológico. Constituyen las respuestas de estrés.

Es decir, se establece la secuencia detallada en la Figura 1.



Figura 1. Secuencia del estrés [12]

El estrés no suele ser únicamente el producto de enfrentarse a situaciones negativas, peligrosas o que requieran un esfuerzo por parte de la persona, sino que situaciones consideradas como positivas, como una actividad que implique una decisión reducida o rutinaria pueden sobrecargar los mecanismos de ajuste. En medida de que el sujeto aplique los mecanismos de ajuste adecuados ante las demandas ambientales o acontecimientos, se tendrá la adaptación o inadaptación de estas [12].

Como adaptación se entiende al conjunto de respuestas que tienen lugar en el organismo y que como resultado conllevan reacciones físicas y/o mentales en un intento de enfrentar los efectos de los estresores [12].

a. Fisiología del estrés

La respuesta fisiológica se refiere a la reacción que ocurre en el organismo humano en respuesta a estímulos estresantes. Cuando el cuerpo humano se presenta ante una situación de estrés, se presentan una variedad de reacciones fisiológicas que ocupan lugar en el organismo las cuales activan el sistema nervioso vegetativo y el eje hipofisopararrenal [13].

El sistema nervioso vegetativo conocido por sus siglas SNV, se compone de estructuras nerviosas cuya función es regular y controlar la actividad de los órganos internos de manera involuntaria. Mientras que el eje hipofisopararrenal (HSP) se

compone por el hipotálamo y la hipófisis que se encuentran situados en la base del cerebro, además de las glándulas suprarrenales que se ubican sobre los riñones [13]. Estos dos sistemas se encargan de liberar sustancias y hormonas que se transportan mediante la sangre para regular la actividad de los órganos.

El proceso fisiológico del estrés se basa en el análisis del factor estresante y en el síndrome general de adaptación [14]

- **Análisis del factor estresante.** El análisis del factor estresante se estudia de acuerdo con tres etapas. La primera etapa se basa en la recepción del factor estresante y se realiza un filtro de la información sensorial a través del tálamo como se observa en la Figura 2 [14].

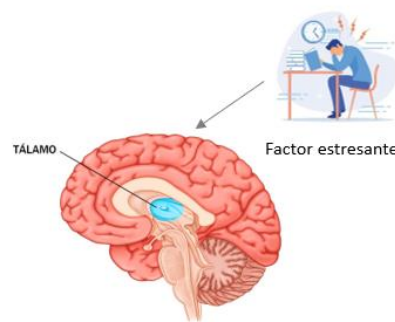


Figura 2. Recepción del factor estresante [14]

En la segunda etapa, se lleva a cabo la programación de la respuesta al estrés, en la cual actúa el córtex prefrontal como se observa en la Figura 3, que desempeña funciones como la toma de decisiones, la atención y la memoria a corto plazo, y el sistema límbico, mediante el que se realiza una comparación entre la nueva situación y los recuerdos previos, y la respuesta se determina en función de la experiencia acumulada [14].

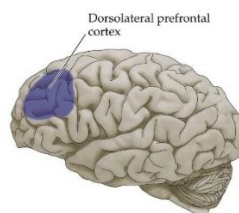


Figura 3. Programación de la respuesta al estrés [15]

Por último, en la tercera etapa se activa la respuesta del organismo, involucrando la amígdala (relacionada con la memoria emocional) y el hipocampo (relacionado con la memoria explícita). Esta respuesta requiere la participación del complejo hipotálamo-hipofisario, así como de la formación reticular y el locus coeruleus, esto se puede observar en la Figura 4 [14].

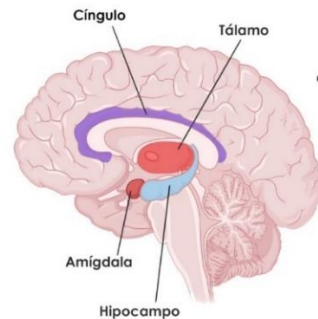


Figura 4. Respuesta del organismo[16]

El organismo responde de diferentes maneras con respecto al nivel de estrés sobre el que esté sometido, en la Tabla 1 se pueden observar las alteraciones que se producen en el organismo cuando se encuentra en una tensión inicial de estrés [13].

Tabla 1. Respuesta del organismo por estrés[13]

AFECCION	TENSION	RESPUESTA
Cerebro	Ideación clara y rápida	Dolor de cabeza, tics nerviosos, temblor, insomnio, pesadillas.
Humor	Concentración mental	Ansiedad, pérdida del sentido del humor.
Saliva	Muy reducida	Boca seca, nudo en la garganta.
Músculos	Mayor capacidad	Tensión y dolor muscular, tics.
Pulmones	Mayor capacidad	Hiperventilación, tos, asma
Estomago	Secreción acida aumentada	Ardores, indigestión. Vomito
Intestino	Aumenta la actividad motora	Diarrea, cólico, dolor, colitis
Vejiga	Flujo disminuido	Poliuria
Sexualidad	Irregularidades menstruales	Impotencia, dismenorrea.
Piel	Menor humedad	Sequedad, dermatitis, erupciones
Energía	Aumento del gasto energético, aumento del consumo de oxígeno.	Fatiga fácil
Corazón	Aumento del trabajo cardiaco	hipertensión, dolor precordial.

- **Síndrome general de adaptación.** El síndrome general de adaptación cuenta con tres etapas [14]:

La primera etapa es la de alerta. En respuesta a un factor estresante, el hipotálamo estimula las glándulas suprarrenales, específicamente la parte medular, para liberar adrenalina. El propósito de la adrenalina es proporcionar energía en caso de una necesidad urgente. Como resultado, se producen diversas respuestas en el cuerpo, como un aumento en la frecuencia cardíaca, la dilatación de los vasos sanguíneos y un aumento en la capacidad respiratoria, dilatación de las pupila y sudoración [14].

La segunda etapa es la de resistencia, que se activa solo si el estrés persiste. En esta etapa, las glándulas suprarrenales, específicamente la zona fasciculada, secretan una hormona llamada cortisol. El cual mantiene estable el nivel de glucosa en sangre para abastecer a los músculos, el corazón y el cerebro. Mientras que la adrenalina proporciona energía de manera inmediata, el cortisol asegura la renovación de las reservas. Esta etapa se caracteriza por la resistencia del organismo al estrés prolongado [14].

La tercera etapa es la de agotamiento, que tiene lugar si la situación de estrés persiste y se asocia con una alteración hormonal crónica, esto puede tener consecuencias orgánicas y psiquiátricas. Si la situación de estrés continúa durante mucho tiempo, el organismo puede llegar a estar abrumado e incluso agotado. En este punto, las hormonas secretadas gradualmente se vuelven menos efectivas y comienzan a acumularse en la circulación sanguínea. Como resultado, el organismo se ve inundado de hormonas que llegan a tener un impacto negativo en la salud, este proceso se observa en la Figura 5 [14].

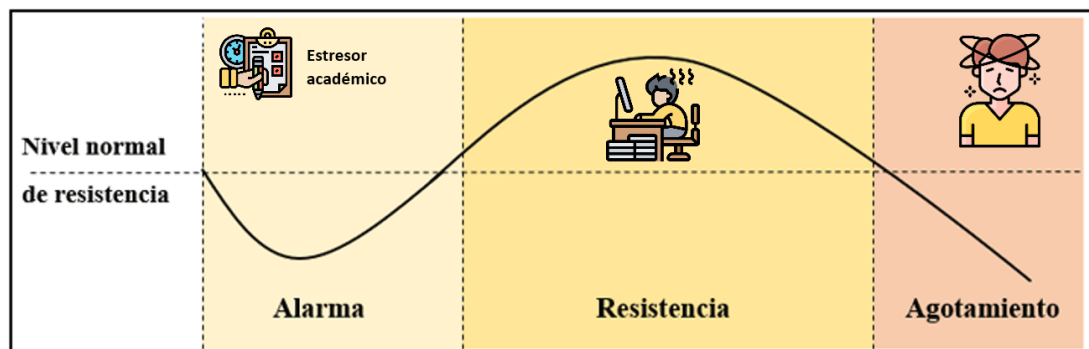


Figura 5. Síndrome general de adaptación [14]

1.3.2 Estrés en estudiantes universitarios

El estrés en estudiantes universitarios también conocido como estrés académico, se define como el malestar que se presenta en los estudiantes ya sea por factores físicos o emocionales que generan una presión considerable, impactando tanto en su desempeño académico como en su capacidad metacognitiva para la resolución de problemas [17].

El estrés en estudiantes representa una de las principales razones detrás de problemas de salud, tanto mentales como físicos, y puede incluso llevar al abandono de la universidad. Varios estudios se han llevado a cabo con el objetivo de identificar las fuentes que originan este estrés, conocidas como "factores estresantes", entre los cuales los más habituales son [17]:

- Procesos de evaluación
- Sobrecarga de trabajo
- Circunstancias en el desarrollo enseñanza-aprendizaje
- Relaciones sociales

La sensación de que hay una carga de trabajo excesiva por parte del estudiante se considera un elemento que aumenta la vulnerabilidad al estrés. Varios estudios han confirmado que un aumento en la cantidad objetiva de trabajo académico se relaciona con una percepción más alta de estrés, niveles más altos de ansiedad y un mayor número de problemas de salud [6].

a. Estresores académicos

Los estresores académicos pueden causar un desequilibrio en el funcionamiento del organismo, lo que se manifiesta a través de una variedad de síntomas. En estos casos, es responsabilidad del estudiante emplear estrategias de afrontamiento para restablecer dicho equilibrio. Es importante destacar que todas las personas experimentan reacciones de estrés en algún momento de sus vidas. Sin embargo, si la respuesta de estrés de un individuo persiste durante un período prolongado, puede agotar sus

recursos y dar lugar a una serie de problemas que pueden afectar su capacidad para llevar a cabo sus actividades diarias. Según lo señalado por Sánchez-Villena, el estrés académico tiende a ser más pronunciado en estudiantes de educación superior debido a las mayores demandas académicas que enfrentan [18].

Las acciones llevadas a cabo durante el curso académico, como deberes, evaluaciones, encuentros, diálogos y compromisos, son acontecimientos frecuentes en la vida de un estudiante universitario a medida que avanzan en su proceso educativo. Sin embargo, cuando se acumulan en exceso, pueden provocar una respuesta de estrés en el cuerpo debido a las demandas académicas y personales. Lo más destacado en esta situación son las consecuencias que impactan en áreas como lo académico, lo físico, lo psicológico y lo social [19].

En base al estudio realizado por J. Barraza se determinan los estresores académicos detallados en la Tabla 2 [19].

Tabla 2. Estresores académicos [19]

1. Competitividad grupal	11. Realización de un examen
2. Sobrecargas de tareas	12. Exposición de trabajos en clase
3. Exceso de responsabilidad	13. Intervención en el aula (responder a una pregunta del profesor, realizar preguntas, participar en coloquios, etc.
4. Interrupciones del trabajo	14. Trabajar en grupo
5. Ambiente físico desagradable	15. La tarea de estudio
6. Falta de incentivos	16. Realización de trabajos obligatorios para aprobar las asignaturas
7. Tiempo limitado para hacer el trabajo	17. Competitividad entre compañeros
8. Problemas o conflictos con los asesores	18. Falta de tiempo para poder cumplir con las actividades académicas
9. Problemas o conflictos con los compañeros	19. Masificación de las aulas
10. Las evaluaciones	20. Sobrecarga académica

b. Indicadores del estrés académico

Este conjunto de indicadores se articula de manera idiosincrática en las personas, de tal manera que el desequilibrio sistémico va a ser manifestado de manera diferente, en cantidad y variedad, por cada persona [19].

Tabla 3. Indicadores del estrés académico [19]

Indicadores	Ejemplos
Físicos: reacciones corporales	Dolores de cabeza, insomnio, problemas digestivos, fatiga crónica, sudoración excesiva, etc.
Psicológicos: relacionados con las funciones cognitivas o emocionales.	Desconcentración, bloqueo mental, problemas de memoria, ansiedad, depresión, etc.
Comportamentales: involucran la conducta.	Ausentismo de las clases, desgano para realizar las labores académicas, aislamiento, etc.

c. Inventario SISCO del estrés académico

Entre los instrumentos para medir los niveles de estrés académico, tiene mayor reconocimiento el inventario sistemático cognoscitivista también conocido por sus siglas SISCO, este inventario fue desarrollado por Arturo Barraza y la segunda versión actual del consta de 23 ítems, este inventario ha sido aplicado en varios estudios de medicina y salud en diferentes países internacionalmente [20].

En Ecuador se realizó un estudio sobre el estrés académico en estudiantes de Tecnología Superior, donde se aplicó como instrumento de medición el inventario sistemático cognoscitivista. Este estudio se realizó con 210 estudiantes donde el inventario presentó una alta confiabilidad igual a 0.894 siendo la sobrecarga de tareas uno de los principales causantes de estrés [21].

La aplicación del inventario SISCO segunda versión es autoadministrada y la contestación es de manera individual la cual tiene una duración máxima de 15 minutos. Consta de 23 ítems que se distribuyen de la siguiente manera [22]:

- El primer ítem es un filtro que determina si la persona encuestada es candidata o no para contestar el inventario.
- El segundo ítem se basa en una escala de cinco valores que determina el nivel de intensidad del estrés académico.
- Los siguientes 7 ítems, permiten la identificación de la frecuencia en que las demandas del entorno estudiantil se denominan estresores académicos.
- Los 7 ítems que continúan tienen el propósito de identificar la frecuencia de los síntomas o respuesta a los estresores académicos.

- Y por último consta de 7 ítems que permiten la identificación del uso de estrategias de enfrentamiento.

Este inventario se aplica con la escala de valores que se observa en la Tabla 4.

Tabla 4. Escala de valores del inventario SISCO [22]

Nunca	Casi nunca	Rara vez	Algunas veces	Casi siempre	Siempre
N	CN	RV	AV	CS	S
0	1	2	3	4	5

- **Niveles de estrés académico.** Para determinar los niveles de estrés académico de acuerdo con el inventario SISCO es necesario obtener la media general, para esto se debe tomar en cuenta las preguntas tres, cuatro y cinco. Y a cada respuesta se le asigna el valor determinado en la Tabla 4, se obtiene la media por cada ítem y posteriormente la media general para así transformar la media en porcentaje [22].

Con el porcentaje se interpretan los resultados con la siguiente métrica que corresponde al baremo indicativo [22]:

- De 0 a 33% nivel leve de estrés
- De 34% a 60% nivel moderado de estrés
- De 61% al 100% nivel severo de estrés
- **Manejo del estrés académico.** El manejo del estrés académico es fundamental para el bienestar de los estudiantes y su éxito en el entorno educativo. La planificación y organización, junto con el establecimiento de metas realistas y una gestión eficiente del tiempo, son fundamentales. El cuidado personal, incluyendo el sueño adecuado y la actividad física, contribuye a mantener un equilibrio saludable. El apoyo social, tanto de amigos como de profesionales, así como la práctica de técnicas de relajación y la identificación de pensamientos negativos, son herramientas valiosas para combatir el estrés. Aprender a decir no y reservar tiempo para recreación también son componentes esenciales. En última instancia, reconocer la individualidad en la gestión del estrés y buscar ayuda cuando sea necesario son claves para promover el bienestar académico [23].

Para cada nivel de estrés académico se tienen los siguientes consejos [23]:

Estrés alto: Para gestionar el estrés elevado como estudiante, es crucial implementar estrategias multifacéticas. Planifica y organiza tu tiempo de estudio, dividiendo tareas en partes manejables. Practica técnicas de relajación, como la meditación o la respiración profunda, y asegúrate de realizar ejercicio regularmente. Mantén una dieta equilibrada, prioriza el sueño y busca apoyo social cuando sea necesario. La aplicación consistente en que estos consejos puedan ayudar a mejorar tu bienestar general y a manejar eficazmente el estrés académico.

Estrés moderado: Cuando enfrentas un nivel de estrés moderado, adopta medidas preventivas y de manejo. Equilibra tus responsabilidades, establece metas realistas y organiza tu tiempo. Integra pausas durante el estudio y practica actividades relajantes. Mantén hábitos alimenticios saludables, asegúrate de descansar adecuadamente y busca apoyo social. Esta gestión proactiva contribuye a mantener un equilibrio saludable en tu vida cotidiana.

Estrés bajo: Cuando el nivel de estrés es bajo, es importante mantener prácticas saludables para prevenir que aumente. Continúa organizando tus tareas, aprovecha la oportunidad para incorporar hábitos de autocuidado, como el ejercicio regular y la alimentación equilibrada. Aunque el estrés sea mínimo, sigue priorizando el descanso adecuado y mantén conexiones sociales positivas. Utiliza este periodo para consolidar hábitos que contribuyan a tu bienestar general y fortalezcan tu resiliencia ante posibles desafíos.

1.3.3 Señales electrofisiológicas

Las mediciones electrofisiológicas corresponden a registros vinculados con la actividad eléctrica del cuerpo humano u otros organismos, ofreciendo datos valiosos acerca del desempeño de sistemas biológicos, con especial énfasis en el sistema nervioso y cardiovascular. Entre las variables más frecuentemente registradas se encuentran [24]:

Tabla 5. Señales electrofisiológicas [24]

Señal electrofisiológica	Descripción
Electrocardiograma (ECG o EKG)	Un registro de la actividad eléctrica del corazón que muestra patrones de ritmo cardíaco y puede ayudar a diagnosticar problemas cardíacos.
Electroencefalograma (EEG)	Un registro de la actividad eléctrica del cerebro, que se utiliza en la investigación y el diagnóstico de trastornos neurológicos.
Electromiografía (EMG)	Registra la actividad eléctrica de los músculos y se utiliza para evaluar la función muscular, detectar trastornos neuromusculares y guiar la rehabilitación [25].
Actividad electrodérmica (EDA)	Mide la conductividad eléctrica de la piel y se utiliza para evaluar la respuesta al estrés, la ansiedad y las emociones.

1.3.4 Actividad electrodérmica (EDA)

La actividad electrodérmica de la piel (EDA), también reconocida como Respuesta Galvánica de la Piel (GSR), implica cambios en las propiedades eléctricas de la piel en respuesta a la producción de sudor. Estas alteraciones en la conductancia cutánea pueden ser evaluadas mediante la aplicación de una corriente continua de baja intensidad de manera no invasiva [26].

La actividad electrodérmica está vinculada a la activación del eje Hipotálamo-Hipofisario-Adrenal (HHA), el cual estimula las glándulas sudoríparas de la piel. Este fenómeno se reconoce como uno de los principales enfoques psicofisiológicos para medir procesos psicológicos, especialmente en contextos de estrés [26].

a. Medida electrodérmica

La respuesta electrodérmica (RED) es un fenómeno actualmente bien asociado con la actividad de las glándulas sudoríparas. Para cuantificarla, se emplean dos medidas fundamentales: la resistencia o conductancia (exosomática) y la detección de voltajes (endosomática). Estas mediciones pueden ser tanto tónicas (nivel de profundidad-L) como fásicas (respuesta variable en el tiempo-R) [27].

La medición suele llevarse a cabo en la palma mediante el uso de electrodos. Cuando se utilizan electrodos reversibles, se minimizan la polarización y los potenciales sesgados. Las señales registradas se distinguen por presentar una recuperación gradual,

con una duración aproximada de 40 segundos. Las amplitudes de la fase son alrededor de 2 microsiemens para la conductividad y de 10 a 20 milivoltios para el potencial, dependiendo del área del electrodo [27].

b. Medición del estrés con la actividad electrodérmica

Los sensores de actividad electrodérmica, conocidos como respuesta galvánica de la piel, se emplean comúnmente para evaluar el estrés a través de la resistencia de la piel y su reacción al sudor. Estudios han evidenciado la eficacia de determinar el nivel de estrés en individuos con una tasa de éxito del 91% mediante el uso exclusivo del sensor GSR/EDA [28].

1.3.5 Electromiografía

La electromiografía, abreviada como EMG, es una técnica que se fundamenta en la evaluación y registro de la actividad eléctrica generada por los músculos esqueléticos. Se distinguen dos tipos: la EMG intramuscular y la EMG superficial. La primera implica el uso de electrodos insertados en el tejido muscular para obtener datos sobre el comportamiento de las fibras musculares. Por otro lado, la EMG superficial requiere electrodos en contacto con la piel y proporciona información acerca del estado de los músculos debajo de dichos electrodos [29].

Relacionado al estrés, la EMG superficial conocida con sus siglas sEMG, se considera la técnica correcta para reconocer el estrés mental, debido a su comodidad, bienestar y corresponde a un método no invasivo. Un sEMG determina el conjunto de contribuciones eléctricas que generan las unidades motoras ubicadas en los músculos, están son detectadas por un grupo de electrodos que se ubican en la superficie de la piel. Estas señales eléctricas se analizan en términos de amplitud y frecuencia [29].

La señal sEMG tiene un ancho de banda que abarca desde 0 hasta aproximadamente 500 Hz. Los valores de la señal fluctúan en el rango de unos pocos microvoltios a un milivoltio, dependiendo de la ubicación y los electrodos. Los valores máximos se encuentran en la franja que va desde 0.1 hasta 1 mV [29].

a. Electrodo para EMG

La corriente en el cuerpo se facilita mediante el transporte de iones. Los electrodos de sEMG convierten la corriente iónica en el tejido en una corriente electrónica. Estos electrodos se dividen principalmente en dos tipos: electrodos secos y electrodos húmedos[29].

Los electrodos de gel de plata/cloruro de plata, conocidos como electrodos de AgCl, son los más frecuentemente utilizados. Estos electrodos húmedos funcionan como electrodos no polarizables, posibilitando una circulación ininterrumpida de carga entre el electrodo y el cuerpo. Los electrodos de AgCl son asequibles y económicos. Mantienen una relativa estabilidad en las mediciones biopotenciales gracias a la reacción química equilibrada entre el electrodo de plata (Ag) y el revestimiento de cloruro de plata (AgCl), estableciendo así una interfaz electroquímica sólida entre el medio iónico y el electrodo. A pesar de estas ventajas, no son la elección ideal para un uso prolongado, ya que el gel electrolítico tiende a deshidratarse con el tiempo, lo que afecta la calidad de la señal. Adicionalmente, algunas personas pueden experimentar sensibilidad al gel electrolítico, provocando irritación cutánea y reacciones alérgicas [30].

Los electrodos secos, aunque tienden a tener una mayor impedancia entre el electrodo y la piel en comparación con los electrodos húmedos, se destacan por ser reutilizables y su idoneidad para mediciones a largo plazo. En arreglos de electrodos, se prefieren los secos para evitar posibles cortocircuitos entre electrodos debido a la conductividad del gel cuando están colocados cercanos. A pesar de su impedancia inicialmente alta (entre 100 k Ω y 400 k Ω), los electrodos secos muestran una disminución a valores de 40 k Ω a 80 k Ω después de 20 minutos debido a la acumulación de sudor y humedad. Curiosamente, a largo plazo, los electrodos secos mantienen una impedancia más estable en comparación con los electrodos húmedos, que tienden a secarse y afectar la calidad de la señal con el tiempo [31].

b. Adquisición de datos EMG

La adquisición de la señal electromiográfica (EMG) se realiza a través de un dispositivo de electromiografía, y esta señal se ve afectada por tres categorías de

factores. En primer lugar, se encuentran los factores causales, que abarcan la estructura y ubicación del electrodo, así como factores fisiológicos, anatómicos y bioquímicos. En segundo lugar, se ubican los factores intermedios, que comprenden fenómenos físicos y fisiológicos influenciados por uno o más factores causales. Finalmente, se encuentran los factores deterministas, que incluyen el número de unidades motoras activas, la velocidad de disparo de las unidades motoras, la interacción mecánica entre las fibras musculares, y características como la amplitud, duración y forma del potencial de acción de la unidad motora [32].

Esta interdependencia de factores presenta desafíos al caracterizar y analizar la señal EMG, ya que la señal adquiere ruido durante su trayecto a través de los tejidos y existe la posibilidad de registrar señales de diversas unidades motoras, lo que dificulta obtener una señal pura o precisa [32].

1.3.6 Sistemas electrónicos

Un sistema electrónico se configura como una disposición física de componentes o elementos interconectados entre sí, con el propósito de cumplir con una necesidad específica. En este contexto, se pueden identificar varias partes esenciales para su operación. Estas partes incluyen la etapa de captación de señales de entrada, que se realiza a través de transductores, también conocidos como sensores de entrada. Estos transductores transforman una variable física específica en una señal eléctrica. Posteriormente, dicha señal es procesada mediante un microcomputador o una tarjeta programable. Finalmente, la señal procesada se visualiza a través de un dispositivo de salida y se utiliza para controlar los actuadores conectados al sistema electrónico [33].

a. Microcontroladores

El microcontrolador se define como un circuito integrado programable, también conocido como "chip", que alberga una variedad de componentes encapsulados. Su función principal radica en gestionar uno o varios procesos, siendo utilizado para ejecutar una tarea específica. Una vez programado, su responsabilidad se limita a administrar la tarea asignada. En sus líneas de entrada/salida, tiene la capacidad de admitir la conexión de actuadores y sensores. [34].

Los elementos fundamentales del microcontrolador incluyen [34]:

- **Procesador o CPU (Unidad Central de Proceso):** Encargado de ejecutar cada instrucción y supervisar el correcto desarrollo del proceso.
- **Diferentes tipos de memorias:** Responsables de almacenar tanto las instrucciones como los diversos datos requeridos por estas.
- **Líneas de Entrada/Salida:** Establecen la conexión entre el microcontrolador y el entorno externo, o viceversa.

Los sistemas electrónicos se emplean para adquirir, procesar, almacenar y transmitir información en un sistema de gestión de información. Existe una estrecha relación entre los sistemas de gestión de información y los sistemas electrónicos, ya que estos últimos proporcionan la infraestructura tecnológica esencial para la implementación y operación de los sistemas de gestión de información [35].

b. Sensores de entrada

Un sensor es un dispositivo, que a partir de la energía del medio donde se mide, da una señal de salida transducible que es función de la variable medida. Se basa en la ampliación de los sentidos para adquirir un conocimiento de cantidades físicas que por su naturaleza no pueden ser percibidas directamente por los sentidos. Se trata de un dispositivo que, aprovechando la energía del entorno, genera una señal de salida proporcional a la magnitud que se busca medir [36].

Tabla 6. Características de los sensores [36]

Características	
Exactitud	Calidad o grado de un instrumento de medición para proporcionar una lectura cercana al valor verdadero de la magnitud medida
Precisión	Se busca que la precisión de la medición sea óptima, lo que implica la presencia o ausencia de una variación aleatoria mínima en la medición de la variable.
Rango de funcionamiento	El sensor debe poseer un amplio rango de funcionamiento, manteniendo la exactitud y precisión en toda su extensión.
Velocidad de respuesta	El transductor debe tener la capacidad de responder rápidamente a los cambios en la variable detectada, idealmente con una respuesta instantánea.
Calibración	Se espera que el sensor sea fácil de calibrar, con procedimientos y tiempos mínimos para llevar a cabo este proceso.
Fiabilidad	El sensor debe exhibir una alta fiabilidad, evitando fallos frecuentes durante su operación.

1.3.7 Inteligencia artificial

La Inteligencia Artificial (IA) se define como la creación de sistemas o dispositivos que emulan la inteligencia humana para llevar a cabo diversas tareas, y tienen la capacidad de mejorar de manera iterativa mediante la información que adquieren. Según la perspectiva de Marvin Minsky, uno de los precursores de la IA, se trata de la ciencia destinada a construir máquinas capaces de realizar actividades que, si fueran ejecutadas por seres humanos, demandarían inteligencia [37].

Un punto destacado acerca de la Inteligencia Artificial es su inclusión en una variedad de disciplinas, que abarcan desde el reconocimiento de voz, procesamiento de lenguaje natural, visión por computadora, hasta la adquisición de conocimientos, robótica, planificación y optimización, entre otras áreas. El objetivo es dotar a un sistema con la capacidad de percibir, razonar, interactuar y aprender [38].

A continuación, se detallan algunos tipos de inteligencia artificial.

Tabla 7. Tipos de inteligencia artificial [38]

Algoritmo	Descripción
Procesamiento de lenguaje natural	El NLP se fundamenta en proporcionar a las máquinas la habilidad de leer y comprender los idiomas que utilizamos como seres humanos [39].
Visión por computadora	Se trata de la obtención automatizada de información a partir de imágenes. La Visión por Computadora involucra una combinación de programación, modelado y conceptos matemáticos [40].
Aprendizaje profundo (Deep Learning)	Es una metodología de enseñanza computacional que utiliza datos etiquetados y redes neuronales con capas múltiples para lograr la ejecución de tareas de manera más cercana a la naturalidad humana. [41].
Machine Learning	También conocido como Aprendizaje Automático, implica la creación de técnicas que posibilitan que las computadoras aprendan a abordar problemas mediante el análisis de datos y la identificación de patrones en ellos [42].

1.3.8 Machine Learning

Machine Learning, también conocido como Aprendizaje Automático, representa una vertiente de la Inteligencia Artificial centrada en el desarrollo de técnicas que posibilitan que las computadoras aprendan a abordar problemas mediante el análisis de datos y la identificación de patrones en ellos. En este contexto, los datos

desempeñan un papel crucial, ya que se utilizan para generar un programa capaz de generalizar comportamientos e inferencias, permitiendo así trabajar con conjuntos más extensos de datos [42].

a. Tipos de algoritmos de aprendizaje

El machine Learning tiene como finalidad lograr el aprendizaje autónomo de las máquinas o sistemas que ayudan hoy en día al ser humano [43]. Se puede definir en cuatro tipos de algoritmos:

Tabla 8. Algoritmos de Aprendizaje automático [43]

Tipo de algoritmo	Descripción
Aprendizaje supervisado	Implica asignar etiquetas a conjuntos de datos para que los patrones identificados puedan ser utilizados posteriormente en la clasificación de nuevos conjuntos de información.
Aprendizaje no supervisado	Cuando un conjunto de datos carece de etiquetas y la única manera de organizarlo es a través de la revisión de similitudes o diferencias que faciliten su clasificación.
Aprendizaje semisupervisado	El aprendizaje semisupervisado opera con conjuntos de datos que incluyen tanto muestras etiquetadas como no etiquetadas. Por lo general, la proporción de datos etiquetados es significativamente menor que la de datos no etiquetados.
Aprendizaje de refuerzo	Se refiere a situaciones en las cuales los datos del sistema carecen de etiquetas en un principio, pero a medida que se llevan a cabo diversas acciones y transcurre un periodo determinado, el sistema recibe retroalimentación y se actualiza con nueva información.

b. Principales algoritmos de aprendizaje supervisado

Los algoritmos de machine Learning son esenciales para impulsar los procesos de aprendizaje, ya que permiten adquirir la información necesaria para tomar decisiones o anticipar el comportamiento de los datos. A continuación, se describen algunos de los algoritmos más utilizados [44]:

- **Linear Classifiers.** se refiere a un conjunto de técnicas de machine Learning que buscan establecer límites de decisión lineales en el espacio de características para realizar tareas de clasificación. Dentro de este algoritmo se encuentran [44]:

- Logical Regression
- Naive Bayes Classifier
- Support Vector Machine
- ***K-Means Clustering.*** es un algoritmo de agrupamiento utilizado en machine learning para clasificar datos en grupos o clústeres. Su objetivo principal es particionar un conjunto de datos en k grupos, donde k es un número predefinido. El algoritmo logra esto asignando cada punto de datos al clúster cuyo centroide (punto central) es el más cercano en términos de distancia euclidiana [44].
- ***Boosting.*** Es un conjunto de técnicas en machine Learning diseñadas para mejorar el rendimiento predictivo de modelos más débiles o base. El objetivo principal de estas técnicas es combinar varios modelos de manera secuencial, asignando un peso a cada modelo en función de su capacidad para corregir las predicciones erróneas de los modelos anteriores [44].
- ***Decision Tree.*** Este modelo se organiza en forma de un árbol invertido, donde cada nodo representa una decisión basada en una característica o atributo, y las ramas del árbol conducen a nodos adicionales o a las hojas, que representan las predicciones o resultados finales. Cada nodo del árbol evalúa una condición específica sobre una característica del conjunto de datos y dirige el flujo hacia las ramas correspondientes. Este proceso continúa hasta llegar a las hojas, donde se toma una decisión final o se hace una predicción basada en las condiciones evaluadas en el camino [44].
- ***Random Forest.*** es un algoritmo de aprendizaje supervisado que se basa en la construcción de múltiples árboles de decisión durante el proceso de entrenamiento y combina sus resultados para mejorar la precisión y la robustez del modelo. Es una técnica de ensamblado, específicamente de "bagging" (Bootstrap Aggregating), que utiliza bootstrap para construir múltiples conjuntos de datos de entrenamiento y luego entrena un árbol de decisión en cada uno de estos conjuntos como se observa en la Figura 6 [44].

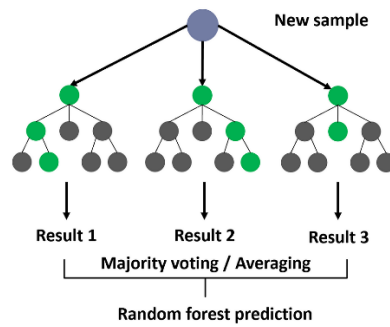


Figura 6. Random Forest [42]

c. Partes de Machine Learning

El procedimiento de Machine Learning comprende diversas componentes esenciales que colaboran de manera conjunta para alcanzar la automatización del aprendizaje de un modelo. Estos componentes se detallan en la Tabla 9 [42]:

Tabla 9. Partes del machine Learning [42]

Estructura	Descripción
Data format (Base de datos)	Conjunto de datos se refiere a un conjunto finito de vectores reales que poseen diversas características. Desde una perspectiva probabilística, la información derivada de los vectores, que forman parte de una distribución estadística multivariante, se fundamenta en condicionar el conjunto inicial para asegurar que todas las muestras sean independientes y tengan una distribución idéntica.
Learnability (Capacidad de aprendizaje)	La capacidad de aprendizaje de un modelo paramétrico se divide en dos aspectos: una estructura estática y un conjunto dinámico de parámetros.

d. Proceso del Machine Learning

El proceso de machine Learning consta de varias etapas las cuales se describen a continuación en la Tabla 10.

Tabla 10. Proceso de Machine Learning [45]

Proceso	Descripción
Definición del problema a resolver	Es esencial tener una comprensión clara de los siguientes elementos: - Variable que se pretende predecir. - Familiaridad con los datos de entrada y su disponibilidad. - Identificación del tipo de problema que se va a abordar.
Adquisición de los datos	Es una etapa crucial ya que impacta directamente en la idoneidad del modelo.
Elegir una medida o indicador de éxito	La métrica debe estar estrechamente vinculada al tipo de problema que se está abordando.

Proceso	Descripción
Preparación de los datos	En este paso, se busca realizar un análisis exhaustivo de los datos disponibles con el fin de comprender la relación y los efectos que tienen en el objetivo de predicción establecido.
Partición de datos	En el conjunto de datos de prueba, generalmente se asigna un 30 % del total del conjunto de datos, reservando el 70 % restante para el entrenamiento. Sin embargo, es importante considerar la cantidad total de datos disponible en el conjunto de datos.
Entrenamiento	Después de analizar la información disponible, llega el momento de seleccionar un modelo o algoritmo de aprendizaje automático que sea apropiado para el objetivo establecido.
Evaluación del modelo	Esta elección se realiza conforme a las métricas de evaluación previamente presentadas en la sección de indicadores de éxito.
Predicción	Con el modelo seleccionado, se procede a su implementación para realizar predicciones. Durante esta fase, se tienen la capacidad de introducir nuevos datos al modelo con el fin de obtener resultados precisos.

e. Métrica o indicador de éxito

La elección de la métrica debe estar estrechamente vinculada al tipo de problema en cuestión. En situaciones de regresión, se opta por métricas como el error cuadrático medio, mientras que, en problemas de clasificación, se recurre a métricas como la precisión, la exactitud y el recuerdo[46].

- **Matriz de confusión.** Una matriz de confusión es un cuadro que registra las predicciones frente a los valores reales mediante filas y columnas. Esta tabla es esencial para evaluar el rendimiento del modelo, ya que proporciona claridad sobre cómo se comporta, identificando si clasifica correctamente o confunde una clase con otra. En el caso de un clasificador binario, la estructura típica de la matriz de confusión se muestra en la Figura 7 [46].

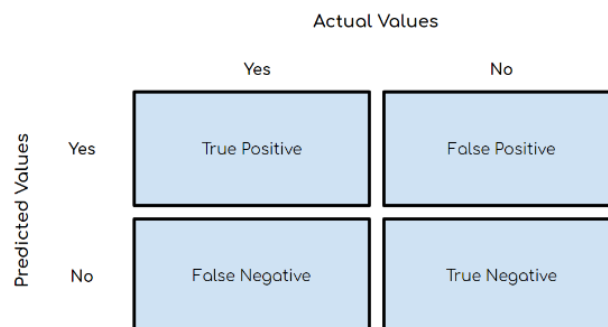


Figura 7. Estructura de la matriz de confusión [46]

La matriz de confusión consta de los siguientes elementos [46]:

- Verdaderos positivos (VP): ocurren cuando la clase de observación y la clase de predicción son ambas verdaderas.
- Falsos negativos (FN): se presentan cuando la clase de observación es verdadera pero la predicción es falsa.
- Falsos positivos (FP): se dan cuando la clase de observación es falsa pero la predicción es verdadera.
- Verdaderos negativos (VN): suceden cuando tanto la clase de observación como la predicción son falsas.
- **Precisión.** La precisión se define como la proporción de las muestras positivas clasificadas correctamente (VP) en relación con el total de muestras clasificadas como positivas (VP o FP). Esta métrica proporciona una medida de la confiabilidad del modelo de aprendizaje automático al identificar correctamente las muestras positivas [46].

$$Precision = \frac{VP}{VP + FP} \quad (1)$$

- **F1-Score 21.** Este indicador es de gran relevancia en el campo del aprendizaje automático, ya que condensa de manera efectiva el rendimiento predictivo de un modelo al combinar dos métricas clave: "Precisión" y "Recall". Su cálculo se realiza mediante la siguiente ecuación [46].

$$F1 = \frac{2VP}{2VP + FP + FN} \quad (2)$$

- **Recall (TPR).** Es una métrica que evalúa la capacidad del algoritmo para prever la clase positiva. A diferencia de la "Precisión", esta métrica no depende del número de clasificaciones de muestras negativas. Cuando el objetivo es minimizar los falsos negativos, se busca que el "Recall" se aproxime a 1. La ecuación para su cálculo es la siguiente [46].

$$TPR = \frac{VP}{VP + FN} \quad (3)$$

f. Protocolo de evaluación

- **Validación cruzada.** La validación cruzada es una técnica estadística empleada para evaluar el desempeño de los modelos de aprendizaje automático. Su objetivo es validar la capacidad del modelo para generalizar a nuevos datos, siendo particularmente útil en situaciones donde el modelo cuenta con pocos puntos de datos o cuando estos presentan desequilibrios [46].
- **Matriz de correlación.** El coeficiente de correlación, un valor comprendido entre -1 y 1, indica la fuerza y dirección de la relación lineal entre dos variables. Un valor cercano a 1 sugiere una correlación positiva fuerte, indicando que el aumento en el valor de una variable coincide con el aumento de la otra. Por otro lado, un valor cercano a -1 señala una correlación negativa fuerte, indicando que el aumento en una variable se asocia con la disminución de la otra. Un valor cercano a 0 sugiere falta de correlación, indicando que no hay una relación lineal discernible entre las variables [47].

Para los coeficientes de correlación ‘corrcoef’ [47]:

- 1 implica correlación positiva fuerte.
- -1 implica correlación negativa fuerte.
- 0 indica falta de correlación.
- Los valores de los coeficientes de correlación están en el rango de -1 a 1.

1.3.9 Aplicación móvil

Una aplicación móvil, comúnmente denominada "app", es un programa informático desarrollado para su ejecución en dispositivos móviles como teléfonos inteligentes y tabletas. Estas aplicaciones tienen el propósito de realizar funciones específicas que

abarcen desde facilitar la comunicación y el entretenimiento hasta ofrecer servicios especializados [48].

a. Aplicaciones nativas

Estas aplicaciones están diseñadas para un único sistema operativo móvil, por lo que se llaman nativas, ya que son específicas de una plataforma o dispositivo en particular. La mayoría de las aplicaciones móviles actuales están diseñadas para sistemas como Android o iOS. En resumen, no es posible instalar ni utilizar una aplicación de Android en un iPhone ni viceversa [48].

b. Etapas para el desarrollo de aplicaciones móviles

El desarrollo de las aplicaciones móviles se basa en cinco etapas, análisis, diseño, desarrollo, pruebas de funcionamiento y entrega [48].

- **Análisis.** Durante esta etapa, se examinan las solicitudes o necesidades de los individuos o la entidad para la cual se está creando el servicio móvil, es decir, el "cliente". El objetivo es establecer las características del entorno o contexto de la aplicación. Se llevan a cabo tres actividades: recopilar requisitos, organizarlos por categorías y adaptar el servicio según las necesidades identificadas [48].
- **Diseño.** En esta fase, la meta es representar visualmente la solución propuesta a través de diagramas o esquemas, teniendo en cuenta la mejor opción al integrar aspectos técnicos, funcionales, sociales y económicos. Si no se logra el resultado deseado en la etapa de prueba de funcionamiento, se vuelve a esta fase para realizar ajustes [48].
- **Desarrollo.** En esta fase, el propósito es llevar a cabo la implementación del diseño en un producto de software. Durante esta etapa, se realizan actividades como la codificación, pruebas unitarias y documentación del código [48].
- **Pruebas de funcionamiento.** El propósito de esta fase es comprobar el rendimiento de la aplicación en diversos escenarios y condiciones. Para lograrlo, se llevan a cabo diversas tareas: emulación y simulación, dispositivos reales [48].

- **Entrega.** Una vez completada la depuración de la aplicación y abordados los requerimientos de última hora del cliente, se considera que la aplicación está concluida y se procede a entregar el ejecutable, el código fuente, la documentación y el manual del sistema [48].

c. Estándares para el diseño visual e informacional de las aplicaciones móviles

Para el desarrollo de las aplicaciones móviles se tomó en cuenta los estándares para el diseño visual e informacional de aplicaciones móviles [49]. Los cuales indican lo siguiente [49] :

- Las aplicaciones deben soportar las distintas resoluciones de los dispositivos.
- El instalador de la aplicación no debe superar los 10MB.
- Generar y proveer los manuales de uso necesarios para el usuario que utiliza la aplicación móvil.
- La paleta de colores a utilizar se debe definir en el manual de identidad visual para aplicaciones móviles, y deben estar relacionados al objetivo de la aplicación.
- La animación no debe verse excesiva, para que no haya distracciones por parte del usuario.

d. Norma ISO 9241-11

Esta norma establece que la ergonomía de software, incluyendo interfaces de usuario. Esta norma, titulada "ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability," proporciona orientación sobre la usabilidad y la interacción persona-computadora [50].

Esta norma establece que el uso adecuado del color es especial en las aplicaciones para proporcionar continuidad visual y ayuda en la comunicación a los usuarios. Los colores en las aplicaciones deben funcionar bien juntos, no en conflicto [50].

Los estándares que favorecen la organización, estructuración y el diseño visual e informacional de las aplicaciones móviles se muestran en la Tabla 11.

Tabla 11. Estándares para el diseño de una app móvil [50]

Estándar	Indicadores
Diseño visual	Colores Iconografía Foco visual Componentes de diseño a partir del lenguaje de programación.
Diseño informacional	Sistema de navegación Sistema de búsqueda Criterios de accesibilidad Criterios de usabilidad

1.3.10 Base de datos

Una base de datos consiste en un conjunto de datos organizado y estructurado que se guarda en dispositivos electrónicos, siendo comúnmente gestionado por un sistema de administración para controlar la información almacenada [51].

a. Base de datos relacional

Una base de datos relacional es responsable de almacenar y proporcionar acceso a datos relacionados mediante identificadores que representan la información en tablas. En estas tablas, cada fila constituye un registro con una clave única, mientras que las columnas contienen los atributos de los datos. Cada registro por lo general tiene un valor para cada atributo, facilitando así la relación entre los datos [51].

1.3.11 Cloud Computing

La computación en la nube implica el almacenamiento y acceso a datos y programas a través de Internet en lugar del disco duro de una computadora local. Dentro de los servicios de la computación en la nube, se pueden identificar tres tipos [52]:

El Software como Servicio (SaaS) implica que una empresa se suscribe a una aplicación y accede a ella a través de Internet mediante un modelo de alquiler. Por otro lado, la Plataforma como Servicio (PaaS) permite a una empresa desarrollar aplicaciones personalizadas para uso interno. La Infraestructura como Servicio (IaaS) involucra a empresas como Google, Microsoft y Rackspace que ofrecen servicios de

alquiler de máquinas virtuales, con la posibilidad de acceder a planes gratuitos por un tiempo limitado según las necesidades [52].

1.4 Objetivos

1.4.1 Objetivo general

Implementar un sistema para la detección de niveles de estrés en estudiantes universitarios a partir de señales electrofisiológicas mediante técnicas de machine Learning.

1.4.2 Objetivos específicos

- Determinar las variables fisiológicas que definen el estrés en estudiantes universitarios.
- Seleccionar un algoritmo de machine Learning para la detección de niveles de estrés
- Diseñar un sistema electrónico para la adquisición y monitoreo de señales electrofisiológicas para la detección de estrés utilizando machine Learning.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

En el marco de la investigación destinada al desarrollo del presente proyecto, se emplearon instrumentos específicos para la captura de datos. Se utilizó el sensor Seeedstudio Grove – GSR Sensor para medir la actividad electrodérmica y el Muscle Sensor V3 para la actividad electromiográfica. La recopilación de datos se llevó a cabo mediante el lenguaje de programación Dart, en conjunto con el framework Flutter, vinculado a la base de datos creada en MySQL y alojada en el servicio web denominado 000WebHosting, el cual opera como servidor. Para el procesamiento, se utilizó un microcontrolador ESP32-TinyS3. Además, se emplearon baterías recargables; para alimentar el sensor muscular, se usaron dos baterías recargables de 9V, y para alimentar el microcontrolador, se utilizó una batería Nano Tech de litio de 3.7V. En la Tabla 12 se puede observar la descripción de los elementos usados en el hardware del sistema mientras que, en la Tabla 13 se observa el software seleccionado.

Tabla 12. Característica del Hardware seleccionado

Material	Características	Descripción de Uso
ESP32 Tiny S3	Microcontrolador WiFi basado en ESP32	Encargado de la conectividad WiFi y la gestión de datos entre los sensores y la aplicación Python a través de la red inalámbrica.
Seeedstudio Grove – GSR	Sensor de actividad electrodérmica	Mide la conductancia de la piel para evaluar la respuesta galvánica de la piel, utilizada en la captura de señales relacionadas al estrés.
Muscle Sensor V3	Sensor electromiográfico	Captura la actividad eléctrica de los músculos, permitiendo analizar la actividad muscular y su relación con el nivel de estrés.
Batería de 9V recargable	Son una opción de energía portátil que puede recargarse y reutilizarse varias veces.	Se usa para alimentar el sensor muscular.
Batería de litio 3.7V	Son conocidas por tener una buena cantidad de ciclos de recarga	Se usa para alimentar el microcontrolador.

Tabla 13. Características del Software seleccionado

Software	Características	Descripción de Uso
Flutter	Desarrollar aplicaciones móviles para plataformas como Android e iOS desde un solo código base.	Se uso para desarrollar la aplicación móvil que controla el sistema.
MySQL	Sistema de gestión de bases de datos relacional (RDBMS) de código abierto.	Usado para administrar los datos provenientes de los sensores y los ingresados mediante la aplicación móvil.

Software	Características	Descripción de Uso
000WebHosting	Proveedor de servicios de alojamiento web gratuito	Se uso como servidor para almacenar la base de datos y gestionar las APIS creadas para comunicación.
HTTP	Protocolo de comunicación utilizado en la World Wide Web para la transferencia de datos.	Protocolo usado para realizar las comunicaciones entre las peticiones PHP y el servidor.

2.2 Métodos

2.2.1 Modalidad de la investigación

El presente trabajo de investigación tiene la característica de ser un proyecto de investigación aplicada, ya que se buscará dar solución al problema de acumulación de estrés a través de la puesta en práctica de conocimientos teóricos.

La investigación es bibliográfica, ya que fue sustentada con la recopilación de información tales como libros, artículos científicos, Datasheet de dispositivos, publicaciones en internet y tesis previas con la implementación de machine Learning en detección de estrés.

El proyecto es una investigación de campo, debido a que se recopiló información relevante para la detección de estrés en estudiantes universitarios de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

2.2.2 Población y muestra

La investigación se desarrolló en la FISEI con una población finita aproximada de 324 estudiantes, siendo 77 de la carrera de Telecomunicaciones, 81 de la carrera de Industrial, 75 de la carrera de Software y 91 de la carrera de Tecnologías de la Información que corresponde a los estudiantes de séptimo semestre del período académico abril 2023 – septiembre 2023. Aplicando el método de muestreo probabilístico se seleccionó una muestra significativa de 75 estudiantes para el desarrollo de la investigación.

La fórmula para calcular el tamaño muestral en este método es [53]:

Tabla 14. Fórmula para el tamaño muestra [53].

Fórmula	Donde:
$n = \frac{(N * Z^2 * q * p)}{e^2 * (N - 1) + Z^2 * q * p}$	n = es el tamaño muestral necesario. N = Tamaño de la población. Z = es el valor crítico de la distribución normal estándar correspondiente al nivel de confianza deseado. p = es la estimación de la proporción de interés en la población. q = es el complemento de p, es decir, q = 1 - p. e = es el error máximo permitido, que indica la precisión deseada en la estimación.

Cálculo para el tamaño muestral:

$$n = \frac{(324 * 1.96^2 * 0.5 * 0.5)}{0.10^2 * (324 - 1) + 1.96^2 * 0.5 * 0.5} = 74,48 \approx 75$$

2.2.3 Recolección de información

Con el fin de obtener información relevante para el desarrollo del proyecto, se hará uso de una variedad de recursos confiables. Se explorarán los diversos repositorios de universidades a nivel nacional, así como libros, revistas científicas, la guía del docente tutor y la información que se obtendrá a través de encuestas realizadas.

Tabla 15. Recolección de información

Criterio	Descripción
¿Para qué?	Determinar el nivel de estrés y prevenir efectos secundarios del mismo.
¿De qué personas u objetos?	De los estudiantes universitarios de la FISEI.
¿Cómo?	A través de una encuesta.
¿Con qué?	Cuestionario realizado en línea.
¿Cuándo?	En el período de septiembre 2023 a febrero 2024
¿Dónde?	En la FISEI

2.2.4 Procesamiento y análisis de datos

Para llevar a cabo el procesamiento y análisis de datos en el contexto del proyecto de investigación, se ha establecido una serie de pasos metodológicos. Estos pasos incluyen:

- Recopilación de datos o información provenientes de diversas fuentes.
- Revisión minuciosa y sistemática de la información obtenida
- Análisis de la fundamentación teórica mediante el enfoque conceptuales que respaldan las variables de investigación.
- Deducción cuidadosa de la información más relevante y útil para el proyecto de investigación.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

3.2 Desarrollo de la propuesta

El desarrollo de la propuesta se fundamenta en la explicación y justificación de la toma de decisiones necesarias para la implementación del proyecto en base al esquema general de arquitectura del sistema, donde se divide el proceso en etapas. A partir de dicho esquema, se detallan los componentes y tecnologías necesarias para el desarrollo. Además, se relacionan los objetivos planteados con los resultados obtenidos durante la implementación. El objetivo de este apartado es proporcionar una perspectiva detallada y comprensible del proceso que se realizó para la elaboración de la propuesta.

Se usó como base la arquitectura básica de un sistema electrónico enfocado a los requerimientos del sistema como se muestra en la Figura 8, con lo que se pudo determinar tres etapas para el desarrollo del sistema.

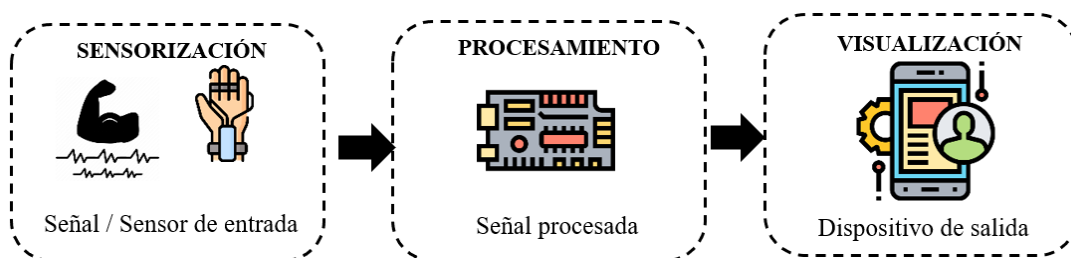


Figura 8. Esquema básico de la arquitectura del sistema

3.2.1 Requerimientos del sistema

El sistema de detección de niveles de estrés académico se basa en un prototipo electrónico en conjunto con un sistema de gestión de información en este caso una aplicación móvil para la visualización de datos, su objetivo es ser una herramienta de apoyo para profesionales de la salud mental además de los estudiantes universitarios. Es por esta razón que debe cumplir ciertos requerimientos que se detallan a continuación.

a. Requerimientos del sistema electrónico

- Garantizar que el dispositivo sea portable, cómodo y compacto.
- Debe ser no invasivo en la adquisición de señales.
- Se debe integrar sensores específicos para medir señales electrofisiológicas y obtener datos precisos.
- La comunicación entre el prototipo y la aplicación móvil debe ser inalámbrica.
- Debe contar con un sistema de carga controlado.

b. Requerimientos de la aplicación móvil

- Recopilar y preparar un conjunto de datos diverso y representativo que incluya situaciones académicas típicas y niveles variados de estrés para entrenar y evaluar los modelos.
- Diseñar una interfaz de usuario intuitiva que permita a los usuarios interactuar con el sistema, visualizar los resultados y recibir retroalimentación sobre sus niveles de estrés académico.
- Proporcionar documentación clara y completa para usuarios y desarrolladores, y asegurarse de que el sistema sea fácil de usar.

3.2.2 Selección del hardware

Se considero los requerimientos del sistema y se realizó un análisis comparativo entre distintos componentes electrónicos presentes en el mercado para seleccionar los componentes más adecuados de acuerdo con cada etapa principal del sistema.

a. Etapa de sensorización

Esta etapa es la encargada de la adquisición de los datos que se obtienen mediante las señales electrofisiológicas para así enviarlos al microcontrolador para ser procesados.

- **Sensor EMG**

Estos dispositivos son sensores que tienen como componentes amplificadores operacionales cuya función es medir un diferencial de potencial mediante electrodos que se conectan en el músculo del paciente, en la Tabla 19 se analiza una comparación entre los sensores disponibles en el mercado y que se han utilizado para investigaciones previas.

Muscle Sensor 8832 V3

El Muscle Sensor V3 (Figura 9) es un dispositivo que mide la actividad eléctrica de los músculos (EMG). Su diseño está orientado para su integración directa con un microcontrolador. Su característica principal es la medición de la actividad eléctrica amplificada, rectificadora y suavizada. El valor de salida se ajusta según la cantidad de actividad registrada en el músculo seleccionado. Utiliza electrodos para detectar la señal EMG generada por la contracción muscular. Este sensor es conocido por ser fácil de usar y versátil en aplicaciones relacionadas con la monitorización de la actividad eléctrica de los músculos de acuerdo con las especificaciones detalladas en la Tabla 16



Figura 9. Muscle Sensor V3[54]

Tabla 16. Parámetros técnicos del muscle sensor V3[54]

Parámetros técnicos	Descripción
Voltaje de la Fuente de alimentación	Min ($\pm 3,5$ V) TYP (± 5 V) Máx. (± 18 V)
Ajuste de ganancia, ganancia = $207 \cdot (X/1k\Omega)$	Min ($0,01 \Omega$ ($0,002x$)) TYP ($50 k\Omega$ ($10,350x$)) Máx ($100 k\Omega$ ($20,700x$))
Voltaje de señal de salida (rectificado y suavizado)	Min ($0V$) Máx ($+ V_s$)
Voltaje de entrada diferencial	Min (0 mV) TYP ($2-5mV$) Máx ($+ V_s / ganancia$)
Dimensiones	$25.4mm \times 25.4mm$ (1pulgada x 1 pulgada)

Myoware Muscle Sensor (AT-04-001)

El Myoware Muscle Sensor (Figura 10) tiene la función de cuantificar la activación muscular mediante la medición del potencial eléctrico a esto se le conoce como electromiografía. Está diseñado para su conexión directa con un microcontrolador por lo que ofrece una señal amplificada, rectificadora e integrada a su salida. En la Tabla 17 se analizan sus especificaciones técnicas.



Figura 10. Myoware Muscle Sensor[55]

Tabla 17. Parámetros técnicos del Myoware Muscle[55]

Parámetros técnicos	Descripción
Voltaje de suministro	Min (+ 2,9 V) TYP (+3,3V o +5V) Max (+ 5,7 V)
Potenciómetro de ganancia	Min (0,01 Ω) TYP (50 k Ω) Máx. (100 k Ω)
Voltaje de señal de salida	Min (0V) Máx. (+ Vs)
Impedancia de entrada	TYP (110 G Ω)
Corriente de suministro	TYP (9 mA) Máx. (14 mA)
Sesgo de entrada	TYP (1 pA)
Dimensiones	52.3mm x 0.82mm

Sensor análogo EMG SEN0240

Este sensor (Figura 11) se caracteriza por detectar la actividad muscular y neural a través de la captación de señales EMG con un circuito filtrado integrado. La salida de la señal es analógica y se clasifica como inductivo activo, que ofrece una recopilación de señales de alta calidad y se caracteriza por su sencillez de uso, sus especificaciones técnicas se pueden observar en la Tabla 18.



Figura 11. Sensor EMG SEN0240[56]

Tabla 18. Parámetros técnicos del Sensor EMG SEN0240[56]

Parámetros técnicos	Descripción
Voltaje de suministro (Vs)	Min (+ 3,3 V) TYP (± 3 V) Máx (5.5 V)
Rango de detección	Min ($\pm 1,5$ mV)
Voltaje de señal de salida	Min (0V) Máx (+ Vs)
Dimensiones	22mm*35mm (0.87*1.38) pulgadas

Selección del sensor EMG

Para seleccionar el sensor adecuado para el desarrollo del proyecto, se tomó en cuenta los requerimientos del sistema para ello se realizó una comparativa entre las especificaciones y en el precio de los tres sensores descritos con anterioridad, esta información se obtuvo de sus Datasheet y tiendas electrónicas como, Amazon, Mercado Libre entre otros.

Tabla 19. Tabla comparativa entre sensores EMG.

Sensor	Muscle Sensor 8832 V3	EMG AT-04-001	EMG SEN0240
Parámetros			
No Invasivo	Si	Si	Si
Su colocación cubre los músculos involucrados	Si	Si	No
Posee ganancia ajustable	Si	Si	Si
Su tamaño es considerable para un dispositivo portable	Si	No	Si
Filtros configurables	Si	Si	Si
Compatibilidad con el microcontrolador	Si	Si	Si
Tipo de señal de salida	Analógica	Analógica	Analógica
Costo	\$45.00	\$100	\$60
Disponibilidad	Ecuador	Ecuador	Estados Unidos

Para seleccionar el sensor muscular, se toma en cuenta los requerimientos del sistema por lo tanto se llega a la conclusión que es óptimo utilizar el Muscle Sensor 8832 V3, debido a que posee ganancia ajustable y a comparación de los otros sensores, su tamaño es ideal para la aplicación de un sistema portable, además cuenta con electrodos que permiten obtener la señal eléctrica de los músculos necesarios.

- **Sensor GSR/EDA**

La función de un sensor GSR (Galvanic Skin Response) o EDA (Electrodermal Activity) es medir la conductancia eléctrica de la piel. Esta conductancia varía con la

cantidad de sudor en la piel, que a su vez está relacionada con la actividad nerviosa y el nivel de estrés. Este sensor se coloca en los pulpejos de los dedos, ya que esta área es propensa a la sudoración y permite una medición eficaz de la conductancia eléctrica de la piel. Algunos de los sensores analizados se detallan a continuación.

Seeedstudio Grove – GSR Sensor

Este módulo (Figura 12) está diseñado para detectar cambios en la conductividad eléctrica de la piel, que pueden estar relacionados con factores como el estrés, la ansiedad o la actividad emocional. Este sensor suele utilizarse en aplicaciones que buscan cuantificar la respuesta fisiológica de la piel ante estímulos emocionales o situaciones de estrés. Sus parámetros técnicos se visualizan en la Tabla 20.

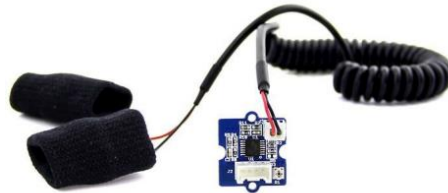


Figura 12. Grove – GSR Sensor[57]

Tabla 20. Parámetros técnicos del Grove GSR Sensor [57]

Parámetros técnicos	Descripción
Voltaje	3.3V a 5V
Corriente	0.72 mA
Dimensiones	20 x 20 mm

CJMCU-6701 GSR

El sensor GSR CJMCU-6701 GSR (Figura 13) es un dispositivo diseñado para medir la respuesta galvánica de la piel, también conocida como conductancia de la piel. Este tipo de sensores detectan cambios en la conductividad eléctrica de la piel, que pueden variar en respuesta a factores emocionales o de estrés, en la Tabla 21 se observan las especificaciones técnicas del sensor CJMCU-6701 GSR.



Figura 13. Skin Sensor Module CJMCU 6701[58]

Tabla 21. Parámetros técnicos del Sensor CJMCU-6701

Parámetros técnicos	Descripción
Voltaje	2.5V a 6V
Corriente	0.52 mA
Dimensiones	25 x 22 mm

Techtonics GSR Galvanic Skin Response

El módulo (Figura 14) de respuesta galvánica de la piel GSR), se refiere a la propiedad que provoca que nuestra piel se convierta momentáneamente en un mejor conductor de electricidad durante estos estados emocionales. En la Tabla 22 se analizan las especificaciones técnicas del sensor GSR de Techtonics.



Figura 14. Techtonics GSR [59]

Tabla 22. Parámetros técnicos del Techtonics GSR [59]

Parámetros técnicos	Descripción
Voltaje	3.3 V a 5V
Corriente	1 mA
Dimensiones	34 x 22 mm

Selección del sensor GSR

En la Tabla 23 se realizó una comparativa de los sensores antes mencionados, destacando sus parámetros compatibles con los requerimientos del sistema, además de su costo que fue estimado por páginas web de venta libre como mercado libre, Amazon y Ebay.

Tabla 23. Tabla comparativa entre sensores GSR

Parámetro \ Sensor	Grove – GSR Sensor	CJMCU-6701 GSR	Techtonics GSR
No invasivo	Si	Si	Si
Su tamaño es considerable para un dispositivo portable	Si	Si	No
Sensibilidad	Ajustable con potenciómetro	No ajustable	Ajustable
Compatibilidad con el microcontrolador	Si	Si	Si
Costo	\$25.00	\$20	\$20
Disponibilidad	Estados Unidos	China	India

Se eligió en Grove GSR sensor de Seedstudio debido a que cumple con todos los parámetros necesarios, además de que incluye electrodos los cuales permiten medir con mayor precisión la actividad electrodérmica, una característica que destaco para su elección fue su disponibilidad debido a que es el dispositivo que se encuentra más cercano y tiene buen accesible.

b. Etapa de procesamiento

- ***Selección del microcontrolador***

Un microcontrolador es un dispositivo programable que integra un procesador, memoria y periféricos en un solo chip, diseñado para controlar tareas específicas en sistemas electrónicos. La función principal de un microcontrolador es ejecutar instrucciones almacenadas en su memoria para realizar diversas operaciones de control en un sistema. A continuación, se detallan los microcontroladores escogidos para realizar una comparación.

ESP8266 NodeMCU

El ESP8266 (Figura 15) es el nombre de un microcontrolador creado por Espressif Systems. Este microcontrolador funciona como una solución integral para la conexión WiFi, actuando como un puente entre microcontroladores existentes y la red inalámbrica, además de tener la capacidad de ejecutar aplicaciones de forma autónoma [60].



Figura 15. Esp8266 NodeMCU[60]

ESP32 Tiny S3

El ESP32-S3 (Figura 16) es un sistema en un solo chip (SoC) de bajo consumo que se basa en un microcontrolador (MCU) y cuenta con Wi-Fi integrado de 2.4 GHz y Bluetooth Low Energy. Este dispositivo incluye un potente microprocesador dual de alto rendimiento (Xtensa 32 bits LX7), un coprocesador de baja potencia, una banda base para Wi-Fi, otra para Bluetooth LE, un módulo de RF y varios periféricos [61].



Figura 16. ESP32 Tiny S3[61]

ESP32 NodeMCU

El microcontrolador ESP32 NodeMCU (Figura 17) es una plataforma de desarrollo compacta y versátil basada en el chip ESP32 de Espressif Systems. Este dispositivo integra capacidades de conectividad WiFi y Bluetooth, permitiendo una amplia gama de aplicaciones IoT y proyectos embebidos [62].



Figura 17. ESP32 NodeMCU [62]

Para seleccionar el microcontrolador óptimo para el proyecto se realizó una tabla comparativa (Tabla 24), para determinar sus especificaciones técnicas y parámetros importantes que tengan relación con los requerimientos del sistema, además de analizar su costo y disponibilidad, esto se analizó en portales web como Mercado Libre, Amazon y proveedores electrónicos de Ecuador.

Tabla 24. Tabla comparativa entre microcontroladores

Sensor Parámetros	ESP8266 NodeMCU	ESP32 Tiny S3	ESP32 NodeMCU
Voltaje	3.3 V	3.3 V – 5V	3.3 V
Dimensiones	58mm x 31mm x 13mm	35mm x 18mm x 4.3 mm	52 x 31 mm x 10 mm
Pines Digitales	13	9	24
Pines Analógicos	1	8	16
Protocolo Wifi	802.11 b/g/n	802.11 b/g/n a 2.4 GHz	802.11 b/g/n/d/e/i a 2.4 GHz
I2C	Si	Si	Si
Frecuencia	80MHz	80MHz	160MHz
Flash	No	8Mb	4Mb
Sram	No	512kB	4Mb
Costo	\$12	\$20	\$13
Disponibilidad	Ecuador	Estados Unidos	Ecuador

Con el análisis comparativo de los microcontroladores, se ha escogido el microcontrolador ESP32 Tiny S3, debido a sus especificaciones técnicas ya que cumplen con los requerimientos del sistema, uno de los más importantes hace referencia a las dimensiones del microcontrolador debido a que el objetivo es que el dispositivo sea portable y la Esp32 Tiny s3 es ideal para este tipo de proyectos además de que cuenta con los pines analógicos necesarios para la conexión de los sensores elegidos.

c. Suministro de energía

El suministro de energía se eligió en base a las características de los sensores necesarios para el proyecto, además del microcontrolador elegido. Para la alimentación del sensor EMG, se realizó una comparativa de baterías que cumplen con las especificaciones de voltaje y corriente especificadas en la Tabla 16.

Tabla 25. Comparación de baterías para sensor EMG

Parametros	Bateria Li-Ion	Bateria Nickel
Capacidad	650mAh	175mAh
Voltaje	9 V	9V
Dimensiones	48 x 16 x 26 mm	48 x 16 x 26 mm
Característica	Recargable	Recargable

En base al análisis realizado en la Tabla 25, se eligió la batería de Li-Ion de 9V debido a que esta tiene una corriente de 650mAh, lo que permite mayor tiempo de uso de la batería.

Con el análisis de las especificaciones del microcontrolador definidas en la Tabla 24, se escogió una batería Lipo Nanotech. Después de evaluar las opciones, se optó por la tecnología de batería Litio-Ion. Esta elección se fundamenta en la mayor durabilidad que estas baterías ofrecen en comparación con las baterías de polímero de litio (Lipo), así como en su mayor densidad de energía. Además, las baterías de Litio-Ion presentan una tasa de conversión superior y están disponibles en el mercado

Tabla 26. Comparación de baterías para la placa

Parametros	Bateria Lipo Nanotech	Bateria Lipo Megatronica
Capacidad	600mA	350Ma
Voltaje	3.7 V	3.7 V
Dimensiones	4 x 2 x 1 cm	5 x 3 x 1 cm
Peso	17 g	21 g
Celdas	1	1

Un circuito de carga controlada es esencial para este tipo de baterías, y en el mercado hay varios módulos de carga disponibles para este propósito. No obstante, la carga de la batería se llevará a cabo mediante la tarjeta TinyS3, ya que esta tarjeta incorpora un módulo TP4056 diseñado para recargar baterías Lipo de celda única con un voltaje de hasta 4.2V. Además, cuenta con un circuito de carga compartida que posibilita que el dispositivo cargue la batería y funcione simultáneamente.

3.2.3 Selección de tecnologías

El desarrollo del proyecto requiere la implementación de tecnologías de comunicación inalámbrica junto con un protocolo para la publicación de datos. Para determinar las

tecnologías óptimas para el sistema se realizó un análisis comparativo, con el objetivo de identificar las tecnologías que se ajusten a los requerimientos del sistema.

a. Comunicación inalámbrica

La selección adecuada de tecnologías de comunicación inalámbrica es esencial para garantizar la eficiencia y precisión del sistema propuesto. La comunicación inalámbrica desempeña un papel crucial en la transmisión oportuna y confiable de datos entre el sistema electrónico y el sistema de gestión, para ello se realizó una tabla comparativa entre las principales tecnologías de corto alcance:

Tabla 27. Tabla comparativa entre tecnologías de comunicación inalámbrica

Tecnologías \ Parámetros	Bluetooth	WiFi
Alcance	Corto (hasta 100 metros)	Largo (varios cientos de metros)
Ancho de banda	Bajo (varía según la versión)	Alto (adecuado para transmisión de datos)
Frecuencia	2.4 GHz	2.4 GHz
Seguridad	Cifrado AES	WEP, WPA y WPA2
Consumo de energía	Bajo (eficiente para dispositivos móviles)	Moderado (puede ser mayor en modo activo)
Topología de red	Punto a punto y punto-multipunto	Infraestructura y punto-multipunto

Para realizar la comunicación inalámbrica entre el sistema electrónico y la aplicación móvil, se ha seleccionado la tecnología Wifi en base a la comparación realizada en la Tabla 27, debido a que se tiene la necesidad de establecer una conexión robusta y de largo alcance, fundamental para la transmisión confiable de datos EMG y GSR. Aunque el consumo de energía es moderado, su capacidad para soportar aplicaciones de alta velocidad y la infraestructura de red existente en entornos educativos hacen que WiFi sea la opción más idónea para asegurar la transmisión eficiente y continua de datos.

b. Protocolo de comunicación

Para el desarrollo de la aplicación es necesaria la selección de un protocolo de comunicación eficiente que se ajuste a los requerimientos del sistema. A continuación, se presenta una comparativa en la Tabla 28 entre los protocolos HTTP, MQTT y COAP, considerando aspectos relevantes para el proyecto:

Tabla 28. Tabla comparativa entre protocolos de comunicación

Protocolo Parámetros	HTTP	MQTT	CoAP
Tipo de Protocolo	Protocolo de transferencia de hipertexto	Protocolo de mensajería	Protocolo de aplicación restringida
Overhead de Mensajes	Moderado a alto (cabeceras en cada req.)	Bajo a moderado (encabezados más ligeros)	Bajo (estructura eficiente)
Confianza en la Entrega	Si	No	Opcional
Uso Común	Aplicaciones web y APIs RESTful	Sistemas de mensajería en tiempo real	Dispositivos IoT y redes de sensores
Eficiencia	Eficiente para transferencias de datos grandes.	Eficiente en términos de ancho de banda y consumo de energía.	Eficiente en redes de baja potencia.
Arquitectura	Solicitud-Respuesta	Publicación-Suscripción	Solicitud-Respuesta

Para el protocolo de comunicación se escogió HTTP, debido a la necesidad de establecer APIs robustas y ampliamente compatibles. HTTP es ampliamente utilizado en aplicaciones web y ofrece una estructura clara de solicitudes y respuestas, lo que facilita la implementación de APIs RESTful. Dado que el proyecto implica la creación de APIs para la comunicación entre el prototipo y la aplicación móvil, HTTP emerge como la opción preferida por su versatilidad, confiabilidad y aceptación generalizada en entornos de desarrollo web e IoT.

c. Lenguaje de programación

Se realizó una comparación entre los lenguajes de programación con mayor relevancia en el desarrollo de aplicaciones móviles en la actualidad como se puede observar en la Tabla 29. De los cuales se consideró el lenguaje Dart y Python, el primero es un lenguaje de programación desarrollado por Google, diseñado para ser eficiente, rápido y fácil de aprender, además es conocido por ser el lenguaje de programación principal utilizado junto con el framework Flutter para el desarrollo de aplicaciones móviles. Y la sintaxis clara y legible de Python facilita la implementación de algoritmos de machine Learning.

Tabla 29. Comparación entre lenguajes de programación

Característica	Dart	Python	JavaScript
Tipo de lenguaje	Estáticamente tipado	Dinámicamente tipado	Dinámicamente tipado
Paradigma principal	Orientado a objetos, funcional	Multiparadigma (OOP, funcional)	Multiparadigma (OOP, funcional)
Uso principal	Desarrollo móvil con Flutter, desarrollo web	Desarrollo web, scripting, automatización	Desarrollo web, scripting
Sintaxis y Legibilidad	Sintaxis clara y legible, similar a C, Java	Sintaxis clara y legible, enfatiza la legibilidad	Sintaxis flexible, similar a C, Java
Comunidad y Soporte	Comunidad en crecimiento, buen soporte para Flutter	Amplia comunidad, amplio soporte, muchas bibliotecas y frameworks	Enorme comunidad, fuerte soporte, numerosos frameworks y bibliotecas
Frameworks	Flutter (para desarrollo móvil), AngularDart	Django (para desarrollo web), Flask	React, Angular, Vue (para desarrollo web)
Manejo de sincronía	Soporte integrado para asincronía, futuros y streams	Soporte para asincronía mediante palabras clave como async y await	Basado en callbacks, promesas, async/await
Desarrollo web	Desarrollo web posible con frameworks como AngularDart	Ampliamente utilizado en desarrollo web con frameworks como Django y Flask	Principalmente utilizado para desarrollo web con frameworks como React, Angular, y Vue
Desarrollo móvil	Flutter para desarrollo móvil, soporte nativo para iOS y Android	Frameworks como Kivy, BeeWare, PyQt para desarrollo móvil	React Native, NativeScript, Cordova para desarrollo móvil

d. Base de datos

La elección de la base de datos es crucial para el almacenamiento y gestión eficientes de los datos recopilados. La funcionalidad de la base de datos debe adaptarse a las necesidades específicas del proyecto, facilitando la manipulación y recuperación de información de manera confiable. A continuación, se presenta una tabla comparativa en la Tabla 30 entre MySQL, SQLite y Microsoft SQL, de las cuales se seleccionó MySQL debido a la necesidad de un sistema robusto y escalable que admita operaciones transaccionales y sea compatible con múltiples plataformas. MySQL es conocido por su rendimiento confiable, amplia aceptación en la comunidad de desarrolladores y su capacidad para manejar grandes cantidades de datos de manera eficiente. Además, al ser de código abierto, proporciona una solución rentable para proyectos académicos y de investigación.

Tabla 30. Tabla comparativa entre bases de datos

Base de datos Parámetro	MySQL	SQLite	Microsoft SQL
Licencia	Gratuita	Gratuita	Pagada
Tipo de base de datos	Relacional	Relacional	Relacional
Sistema Operativo	Windows, Linux, MAC, Android, iOS	Windows, MAC, Android, iOS, Linux	Windows, Linux
Interfaz Grafica	Si	No	Si
Escalabilidad	Alta	Limitada	Alta
Tamaño maximo	Depende del dispositivo	1 TB	524. 272 TB
Compatibilidad multiplataforma	Si	Si	Si
Lenguaje	SQL	SQL	SQL

e. Cloud computing

La selección del servicio de cloud computing es crucial para alojar y gestionar la aplicación, la base de datos y otros recursos en la nube de manera eficiente y segura, para ello se realizó una tabla comparativa que se visualiza en la Tabla 31 entre diferentes proveedores de servicios en la nube, donde se observan buenas prestaciones por parte de 000Web Hosting, Google Cloud y Microsoft Azure.

Tabla 31. Comparativa de servicios de almacenamiento en la nube

Base de datos Parámetros	000Web Hosting	Google Cloud	Microsoft Azure
Licencia	Gratuita	Pagada / Gratuita	Pagada
Base de datos	MySQL	MySQL	Azure SQL
Seguridad	Alta	Alta	Alta
Lenguaje	Python, JavaScript, PHP	Linux / Python	Linux, Python
Dominio	No	Si	Si
Memoria	Ilimitado	Ilimitado	Ilimitado

Se ha elegido el servicio de 000Web Hosting se basa en la naturaleza del proyecto, donde se requiere una solución que ofrezca recursos gratuitos, facilidad de uso y un soporte técnico suficiente para las necesidades específicas del sistema. Aunque 000Web Hosting puede tener limitaciones en comparación con proveedores más grandes, su enfoque accesible y la disponibilidad de recursos gratuitos hacen que sea una opción viable para proyectos académicos e investigativos.

f. Método de entrenamiento de algoritmo de ML

Para entrenar un algoritmo o modelo de aprendizaje automático, se debe utilizar un conjunto de datos para que el modelo aprenda patrones y relaciones entre las características de entrada y salida. A continuación, se describen algunos métodos comunes para realizar el entrenamiento.

Tabla 32. Tabla comparativa de los métodos de entrenamiento

Métodos	Descripción	Características
División de Datos (train_test_split)	Divide el conjunto de datos en conjuntos de entrenamiento y prueba.	<ul style="list-style-type: none"> - Facilita la evaluación del rendimiento del modelo. - Ayuda a evitar el sobreajuste. - Esencial para medir la generalización del modelo.
Supervisado vs. No Supervisado	En el aprendizaje supervisado, se proporcionan ejemplos de entrada y salida al modelo durante el entrenamiento. En el no supervisado, el modelo debe encontrar patrones sin datos de salida previos.	<ul style="list-style-type: none"> - Supervisado: Requiere etiquetas para entrenar. - No supervisado: Descubre patrones sin etiquetas
Ajuste de Parámetros	Ajusta los parámetros del modelo para optimizar su rendimiento.	<ul style="list-style-type: none"> - Optimiza el rendimiento del modelo. - Contribuye a la mejora de la precisión.
Validación Cruzada	Divide el conjunto de datos en subconjuntos para realizar múltiples rondas de entrenamiento y evaluación.	<ul style="list-style-type: none"> - Reduce la variabilidad en la estimación del rendimiento. - Mejora la evaluación del modelo. - Útil para conjuntos de datos limitados.
Regularización	Aplica técnicas para evitar el sobreajuste del modelo, penalizando coeficientes grandes o utilizando técnicas como L1 o L2.	<ul style="list-style-type: none"> - Controla la complejidad del modelo. - Previene el sobreajuste. - Mejora la generalización.
Algoritmos de Optimización	Utiliza algoritmos para minimizar la función de pérdida del modelo, como el descenso del gradiente estocástico (SGD).	<ul style="list-style-type: none"> - Iterativamente mejora los parámetros del modelo. - Converge hacia mínimos locales. - Puede ser sensible a la inicialización.
Validación y Ajuste Continuo	Valida el rendimiento del modelo en datos no vistos y ajusta continuamente el modelo según sea necesario, crucial para modelos desplegados en entornos en tiempo real.	<ul style="list-style-type: none"> - Monitorea y mejora el rendimiento a lo largo del tiempo. - Adaptación a cambios en los datos o el entorno. - Mantenimiento continuo del modelo.

Entre estos métodos se eligió La División de Datos (train_test_split) debido a que es fundamental para evaluar el rendimiento del modelo en datos no vistos y determinar su capacidad de generalización. Este método se utiliza en la fase inicial del proceso de entrenamiento y prueba para dividir el conjunto de datos en segmentos dedicados a entrenamiento y evaluación.

3.2.4 Arquitectura del sistema

El funcionamiento del sistema detector de estrés a través de señales electrofisiológicas se describe en dos etapas fundamentales:

La primera etapa corresponde a la sensorización y procesamiento, que consiste en la adquisición de las señales electrofisiológicas mediante la colocación de electrodos EMG en áreas específicas del brazo derecho del paciente que corresponde a la señal de electromiografía. Simultáneamente, se implementa un sensor de respuesta galvánica de la piel (GSR) en el mismo brazo en zonas de alta conductividad, como los dedos. En esta etapa el microcontrolador Esp32 TinyS3 actúa como la unidad central de procesamiento, ya que además de permitir la adquisición de señales de los sensores, establece la comunicación inalámbrica con el servidor web utilizando la tecnología WiFi y lleva a cabo la divulgación de los datos adquiridos mediante APIs desarrolladas en PHP, que permiten la comunicación entre el microcontrolador y el servidor web mediante peticiones.

La segunda etapa se encarga del registro y almacenamiento de datos, donde se ha establecido un servidor web que permite la administración de la información proveniente de los sensores. Este servidor facilita la interacción con los datos mediante la implementación de APIs desarrolladas en PHP. Dichas APIs se encargan de gestionar el intercambio de información entre el prototipo y la interfaz de usuario, posibilitando la transmisión eficiente de datos sensoriales. Además, el servidor web administra una base de datos construida en MySQL, centralizando el almacenamiento y asegurando la integridad de la información. Para la visualización y control de datos, se ha creado una aplicación móvil utilizando el framework de Flutter, que constituye el backend de la interfaz de usuario y proporciona herramientas robustas para el procesamiento, almacenamiento, control y presentación de la información mediante dashboards intuitivos y eficaces. Este enfoque integral garantiza una gestión eficiente y coherente de los datos desde su origen hasta la interfaz de usuario final. La arquitectura del sistema se puede analizar en la Figura 18 que muestra el esquema general del sistema.

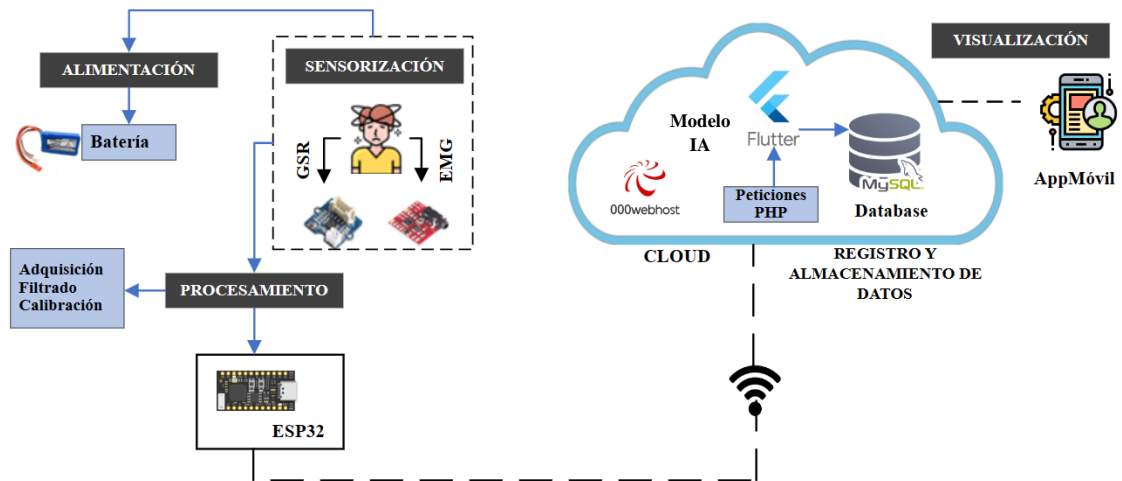


Figura 18. Esquema general de la arquitectura del sistema

3.2.5 Implementación del circuito de monitoreo de electro señales

En esta etapa se describe la implementación del circuito electrónico para el monitoreo y adquisición de las señales electrofisiológicas, para ello es importante conocer el funcionamiento de los sensores usados en el proyecto.

a. Funcionamiento del sensor EMG

Para la medición de la señal de electromiografía se usó el sensor Muscle Sensor V3, como se explicó en el apartado 3.2.2. Este sensor se encarga de amplificar y procesar la actividad eléctrica del músculo y la convierte en una señal analógica simple.

- **Medición de la señal de electromiografía**

Para que la adquisición de la señal EMG, es necesaria la utilización de tres electrodos que transmiten las señales musculares. Cada electrodo se debe colocar en la posición correcta referente al músculo a monitorear. En este proyecto se estudia la señal EMG relacionada a la intervención del estrés en el cuerpo humano por lo que se eligió el músculo largo supinador del antebrazo, los electrodos se deben colocar de acuerdo con el siguiente método que hace referencia a cada color del electrodo como se observa en la Figura 19.

Electrodo verde: Se debe colocar en el centro del músculo deseado.

Electrodo amarillo: Se coloca en el extremo del músculo deseado.

Electrodo rojo: Se coloca en una zona no muscular cerca del músculo deseado.

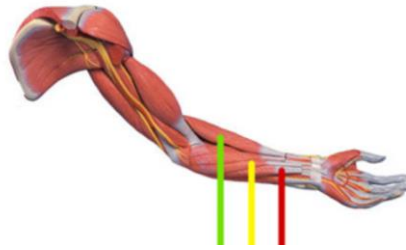


Figura 19. Colocación de los electrodos para EMG[32]

Para la conexión del sensor con el microcontrolador se debe seguir los requerimientos de dicho sensor, este consta de dos pines para la fuente de alimentación, dos pines GND y un pin para la señal. La fuente de alimentación consta de dos baterías, el terminal positivo de la primera se conecta en el pin +Vs del sensor y su terminal negativo se conecta con el terminal positivo de la segunda batería, realizando así un puente que constituye el GND que va conectado al pin GND del sensor. El terminal negativo de la segunda batería se conecta al pin -Vs del sensor.

Para conectar el sensor con el microcontrolador se conecta un pin analógico del microcontrolador en este caso la ESP32 TinyS3. Y el segundo GND se conecta con el GND del microcontrolador como se observa en la Figura 20. Esta conexión se realizó en base a su Datasheet descrito en el Anexo F.

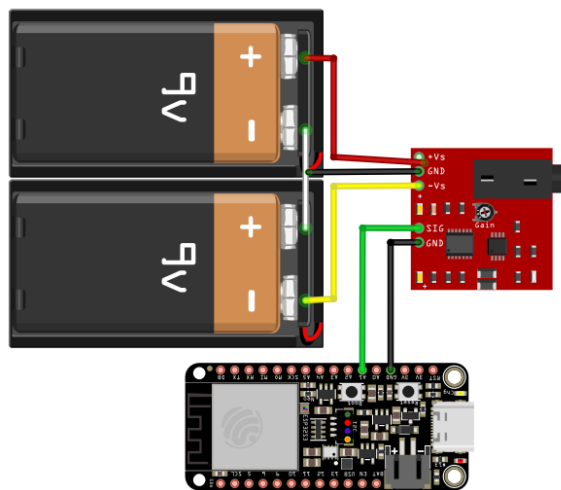


Figura 20. Diagrama de conexión del sensor EMG

b. Funcionamiento del sensor GSR

Para la adquisición de datos de la actividad electrodérmica se usó el sensor Grove GSR como se explica en el apartado 3.2.2, este sensor registra las variaciones en la conductancia eléctrica de la piel, relacionadas con la actividad de las glándulas sudoríparas.

- ***Medición de la actividad electrodérmica***

Para medir la actividad electrodérmica es necesaria la colocación adecuada de los electrodos para obtener lecturas precisas y consistentes. La ubicación más común para la medición GSR es en los dedos de la mano, debido a que esta zona tiene una alta concentración de glándulas sudoríparas. Antes de la colocación de los electrodos se debe tener en cuenta los siguientes aspectos:

- Se debe limpiar la piel en la ubicación seleccionada con un paño húmedo, para limpiar cualquier factor que puede interferir en la conductancia eléctrica.
- Los electrodos se deben colocar en la parte inferior de los dedos, de preferencia en el dedo índice y en el anular como se observa en la Figura 21.
- Los electrodos deben estar en contacto directo con la piel.



Figura 21. Colocación de los electrodos para GSR[57]

Para la conexión entre el Grove GSR sensor y el microcontrolador se debe tener en cuenta los pines del sensor, este consta de un pin para la fuente de alimentación Vcc, un pin para tierra GND y un pin para la señal. Su conexión es sencilla y rápida debido que se conecta a los mismos puertos que tiene el microcontrolador. En la Figura 22 se puede observar el diagrama de conexión del sensor seleccionado. Esta conexión se realizó en base a su Datasheet descrito en el Anexo E.

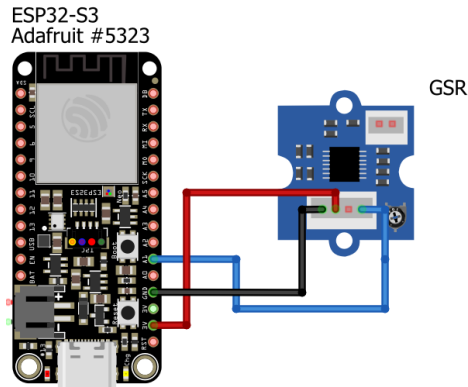


Figura 22. Diagrama de conexión del sensor GSR

c. Diseño electrónico

El diseño electrónico se desarrolló en base a las conexiones de los sensores descritos con anterioridad todo en base a los Datasheet de cada uno, la conexión directa al microcontrolador ESP32 Tiny S3 se desarrolló siguiendo el Datasheet del mismo que se puede analizar en el Anexo D. Después de realizar la conexión del sensor Muscle Sensor V3 y el sensor GSR, en el circuito se ha implementado una batería Lipo de 3.7V como se analizó en la Tabla 26, para alimentar el microcontrolador además se integró un módulo TP4056 que permite la carga de la batería antes mencionada. Para el control de encendido o apagado se ha incrementado un switch que permite el paso de energía de la batería al microcontrolador ESP32 Tiny S3. Este diseño electrónico se visualiza en la Figura 23.

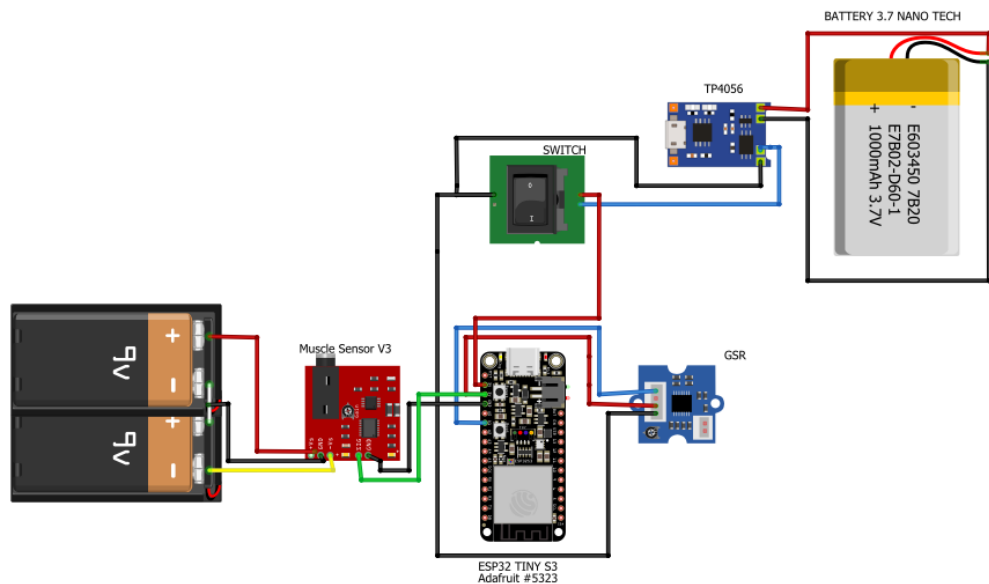


Figura 23. Circuito electrónico del sistema

Con la visualización del diseño electrónico realizado, se procede a realizar el diseño esquemático del circuito para ello se usó el software de simulación EasyEDA que es un entorno de diseño intuitivo para esquemáticos electrónicos, con la instalación de las librerías necesarias y siguiendo el circuito diseñado se realizó el diseño esquemático detallado en la Figura 24.

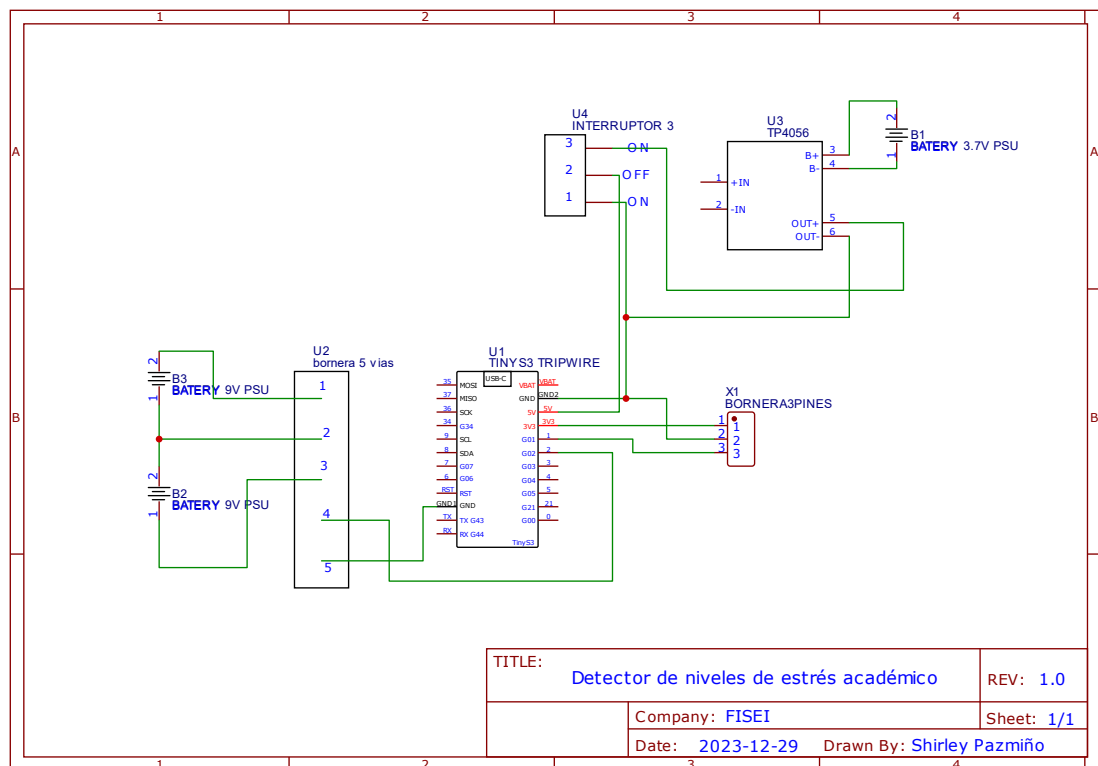


Figura 24. Diseño esquemático del circuito

d. PCB del circuito para la conexión entre los sensores y la ESP32 Tiny S3

Para la fabricación del PCB (Printed Circuit Board), se tuvo en cuenta el diseño esquemático realizado en la Figura 24, el diseño layout del circuito se desarrolló teniendo en cuenta que las dimensiones del PCB deben ser reducidas. En la Figura 25, se puede observar el diseño de las pistas y la ubicación de los componentes del circuito electrónico. Con este diseño se logró obtener una placa con las siguientes dimensiones, 32.893mm x 44.450mm, que resulta ser un buen tamaño para un dispositivo portable.

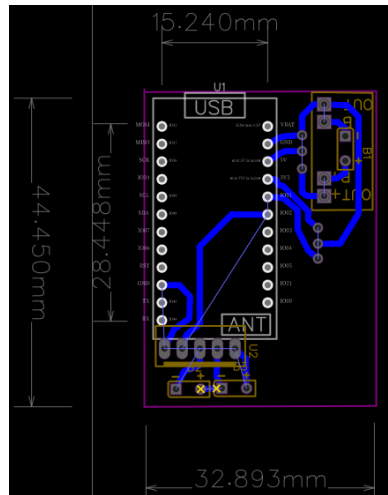


Figura 25. Diseño del PCB

Para la fabricación del PCB, se empleó el método CNC para ello se generó el archivo Gerber. Para su grabado se usó una baquelita de fibra de vidrio (FR4) debido a que una de sus características principales es su durabilidad. En la Figura 26 y en la Figura 27 se observa el resultado de la impresión de la placa con este método.

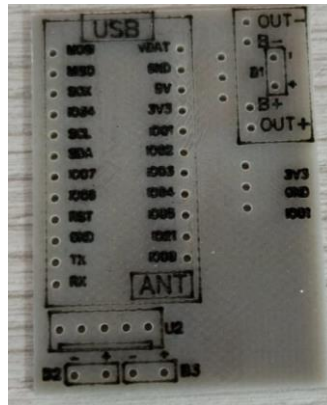


Figura 26. Impresión del PCB con CNC

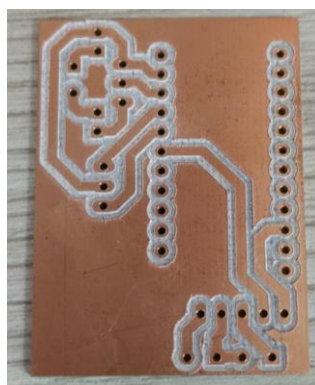


Figura 27. Impresión de las pistas del PCB

e. Diseño de la caja contenedora

Para el diseño de caja contenedora se tomó en cuenta el tamaño del PCB como se observa en la Figura 25. La caja contenedora se realizó con el método de impresión 3D, donde se tomó en cuenta los puertos y el tamaño de los componentes del dispositivo como se observa en la Figura 28 y su implementación se puede observar en la Figura 29.

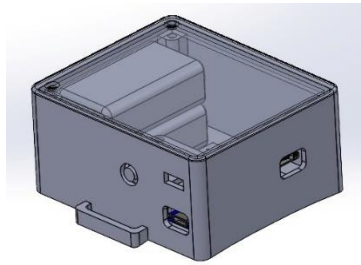


Figura 28. Diseño de la caja contenedora



Figura 29. Circuito integrado en el case

f. Acondicionamiento de los sensores

- *Acondicionamiento de la señal del EMG*

Para el acondicionamiento de la señal EMG, es necesario realizar la inicialización de variables para realizar la lectura del sensor, para este proceso se utiliza la función **analogRead()** para obtener el valor analógico del sensor EMG conectado al pin **emgPin** . Este valor representa la amplitud de la señal eléctrica capturada por el sensor. Con el valor de la lectura del sensor se realiza una conversión a voltaje, que consiste en convertir el valor analógico a voltaje. La lectura analógica devuelve un valor entre 0 y 4095, y se divide por el rango máximo (4096 en este caso) y se multiplica por el voltaje de referencia (**referenciaVoltaje**) para obtener el voltaje correspondiente a la

señal. Una vez realizada la conversión es necesario realizar el acondicionamiento para microvoltios, dado que el sensor EMG tiene una ganancia específica (**gananciaSensor**), se multiplica el voltaje por un factor que tiene en cuenta esta ganancia y se escala a microvoltios. Finalmente, se imprime el valor resultante en microvoltios en el puerto serie para su monitorización. También se incluye un retardo de 1 segundo (**delay(1000)**) entre las lecturas para evitar una excesiva frecuencia de muestreo. En la Figura 30 se muestra el flujograma del sensor EMG, que determina el algoritmo utilizado.

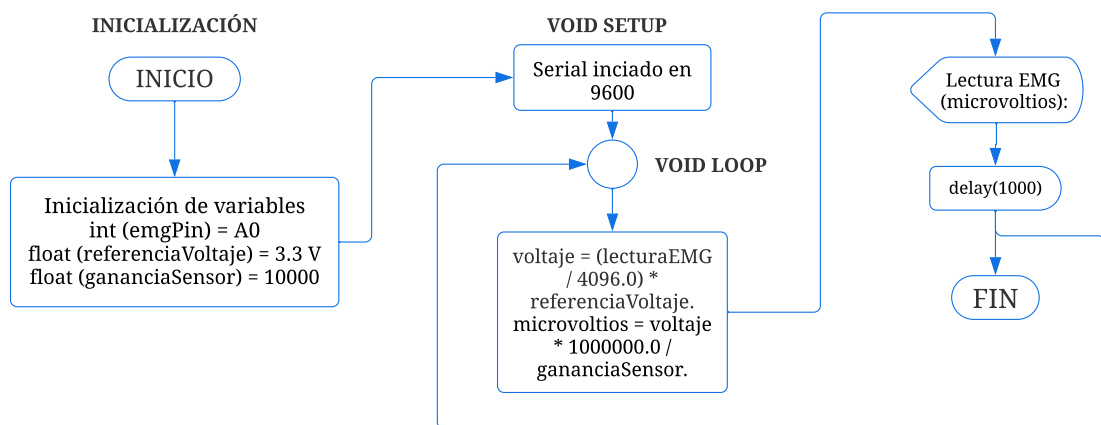


Figura 30. Diagrama de flujo del sensor EMG

- **Acondicionamiento de la señal del sensor GSR**

Para el acondicionamiento de la señal GSR, primero se definen los parámetros específicos para la configuración del sensor GSR, incluyendo el pin al que está conectado, la tensión de referencia de la ESP32, la resistencia de calibración y la constante de conversión a microsiemens. Posteriormente, con la función **setup()**, se inicia la comunicación serial a una velocidad de 9600 baudios para permitir la salida de datos al monitor serial. Se inicia el bucle principal (**loop**) y se lee el valor analógico del sensor GSR utilizando **analogRead()**. Una vez que se tenga el valor medido se convierte el valor leído a microsiemens mediante una serie de cálculos que involucran la conversión a voltaje, el cálculo de la conductancia y la multiplicación por la

constante de conversión. Finalmente se imprime el valor resultante en microsiemens en el monitor serial y se incorpora un retardo de 1 segundo entre las lecturas para evitar una frecuencia de muestreo excesiva. En la Figura 31 se puede analizar el flujograma del sensor GSR.

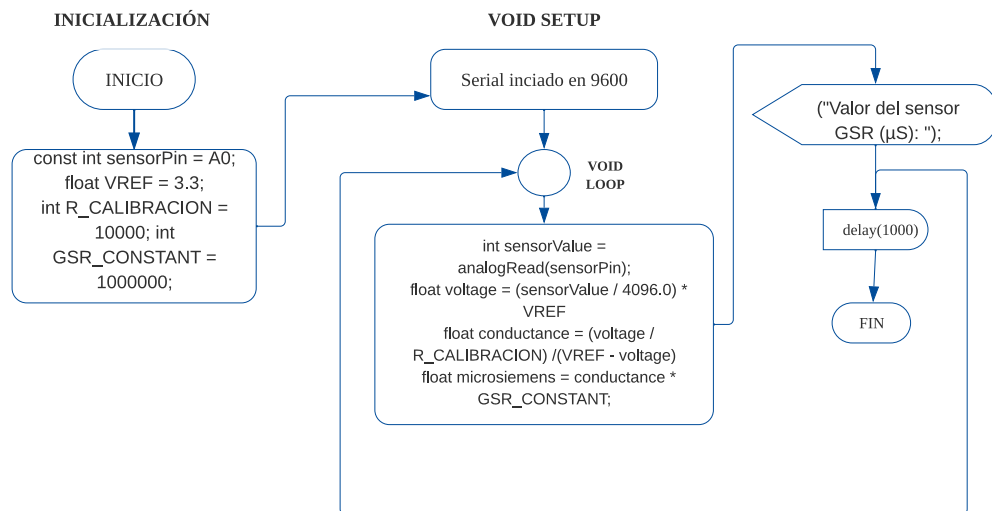


Figura 31. Diagrama de flujo del sensor GSR

g. Procesamiento y transmisión de datos

Para el procesamiento y transmisión de datos se usó el microcontrolador Esp32 Tiny S3. Su funcionamiento consiste en establecer comunicación inalámbrica mediante la tecnología WiFi y la conexión con el protocolo http, mediante peticiones realizadas a través de APIs para su publicación en la app móvil, cuando se realiza dicha conexión se realiza la lectura de los datos dentro del void Loop, donde se realiza el acondicionamiento y calibración de las señales. En la Figura 32 se puede analizar el diagrama de procesos del procesamiento de datos y la publicación de la información mediante peticiones PHP. El código de programación de este procesamiento se puede analizar en el Anexo H.

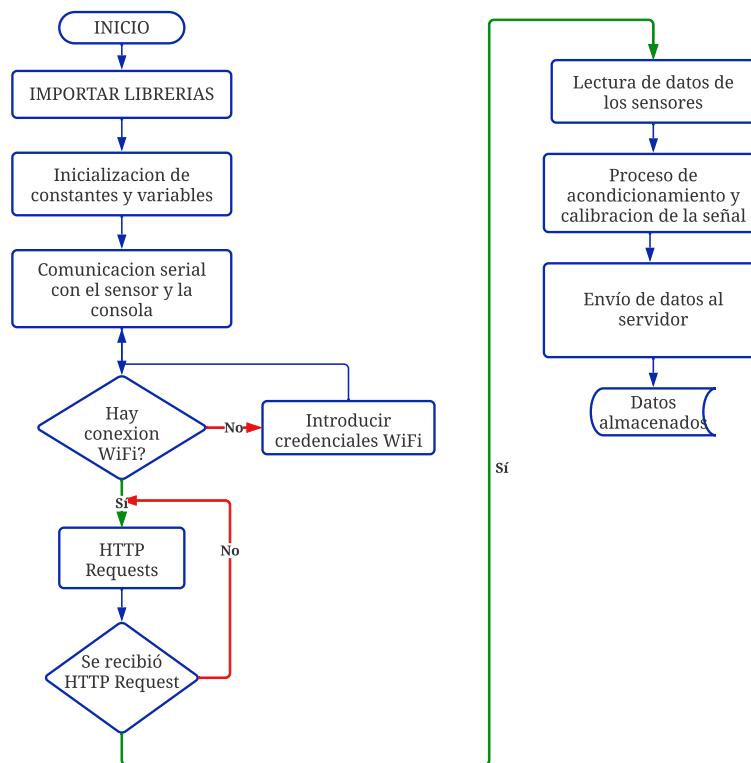


Figura 32. Diagrama del proceso y transmisión de datos

3.2.6 Implementación del modelo de machine Learning

a. Obtención de datos

Para la implementación del modelo de machine Learning fue necesario utilizar bases de datos validadas, debido a que los datos obtenidos con los sensores implementados no son suficientes para el entrenamiento. Se uso una base de datos obtenida en Physionet, una plataforma gestionada por los miembros del *MIT Laboratory for Computational Physiology*, establecida en 1999 bajo el auspicio de *National Institutes of Health (NIH)*, y que tiene como objetivo ofrecer acceso libre a grandes bancos de datos clínicos y fisiológicos.

La base de datos utilizada corresponde al proyecto “Wearable and automotive systems for affect recognition from physiology”, realizado en el MIT por Healey como parte de su tesis doctoral que consiste en mediciones corporales realizadas a varios jóvenes que conducen en entornos estresantes. Entre estas variables se tiene ECG, EMG, HR, GSR de los cuales existe 3900 datos de cada variable fisiológica [63].

Para el proyecto desarrollado se usaron las variables electrofisiológicas que corresponden a la actividad electrodérmica (EDA) y electromiografía (EMG), que son las variables que corresponden a los datos obtenidos por los sensores escogidos en la sección 3.1.1 . La medición de la EDA se realiza con electrodos en el dedo índice de la mano derecha, mientras que la EMG se mide en los músculos ubicados en el brazo de la misma mano. Esta base de datos determina el estrés en dos clasificaciones: estresado y no estresado, enfocándose en las mediciones obtenidas teniendo en cuenta el valor más alto como estresado y la medición más baja como no estresado.

El proyecto integra la recopilación de los datos electrofisiológicos mencionados, así como una evaluación psicológica a través de un formulario estructurado. Para la obtención de datos psicológicos se incorporó un formulario (Anexo A) diseñado para la evaluación del estrés académico mencionado en el punto 1.3.2 que se basa en el Inventario SISCO del estrés académico, para su aplicación en estudiantes universitarios se obtuvo la validación del Psc. Clin. Vinicio Ramírez especialista en Intervención Psicológica el certificado de validación se encuentra en el Anexo B. Este formulario (Anexo A) se aplicó a los estudiantes de la carrera de Telecomunicaciones, y se obtuvo un total de 121 datos correspondientes a preguntas estratégicas que se procesaron de acuerdo con los indicadores de dicho formulario donde se obtuvo tres niveles de estrés: estrés leve, estrés moderado y estrés severo.

b. Preprocesamiento de datos

- ***Procesamiento de los datos EDA y EMG***

Los datos correspondientes a EDA y EMG, obtenidos de la base de datos fueron etiquetados en base a tres niveles de estrés: estrés bajo, estrés moderado y estrés alto en base a los indicadores determinados por la investigación origen de la base de datos, que se encuentra resumido en la Tabla 33 [63].

Tabla 33. Rangos de las mediciones GSR y EMG [63]

Nivel	Equivalente	Rango GSR	Rango EMG
Bajo	0	< 0.7	< 0.5
Moderado	1	[0.8: 3.8]	[0.6: 2.8]
Alto	2	>3.9	>2.9

En base a los rangos definidos en la Tabla 33, se procesó la base de datos teniendo en cuenta las clases y las etiquetas como se observa en la Tabla 34.

Tabla 34. Clase y etiquetas de los datos de EMG y GSR

Clase	Equivalente	Etiqueta
Nivel de estrés	0	Bajo
	1	Moderado
	2	Alto

- **Procesamiento de los datos SISCO**

La implementación del formulario se llevó a cabo en un conjunto de 121 estudiantes pertenecientes a la carrera de Telecomunicaciones. Durante este proceso, se establecieron ocho parámetros clave destinados al entrenamiento y evaluación. Las preguntas extraídas para este procesamiento se observan en la Tabla 35.

Tabla 35. Preguntas escogidas del formulario SISCO [22]

N°	Preguntas
1	Con la idea de obtener mayor precisión y utilizando una escala del 1 al 5 señala tu nivel de estrés, donde (1) es poco y (5) mucho.
2	¿Con que frecuencia te estresa la sobrecarga de tareas y trabajos escolares que tengo que realizar todos los días?
3	¿Con que frecuencia te estresa la forma de evaluación de mis profesores/as?
4	¿Con que frecuencia te estresa el tipo de trabajo que me piden los profesores?
5	¿Con que frecuencia te estresa tener tiempo limitado para hacer el trabajo que me encargan los/as profesores/as?
6	¿Con que frecuencia se te presenta fatiga crónica (cansancio permanente) cuando estás estresado?
7	¿Con que frecuencia se te presentan problemas de concentración cuando estás estresado?
8	¿Con que frecuencia se te presenta desgano para realizar las labores escolares cuando estás estresado?

El análisis de resultados de este instrumento de evaluación se detalla a continuación. Este análisis permite determinar la factibilidad de las preguntas para ser incorporadas en la etapa de entrenamiento.

Como consecuencia de la primera pregunta, se derivan los siguientes resultados cuantitativos en relación con los niveles de estrés entre los encuestados. El 1.7% de

los participantes indicó poseer un nivel 1 de estrés, mientras que el 3.3% manifestó encontrarse en el nivel 2 de estrés. El 19% de los encuestados evaluó su nivel de estrés como grado 3, el 42.1% lo situó en el nivel 4, caracterizado como casi alto, y finalmente, el 33.9% estableció un nivel 5 de estrés, categorizado como alto. Estos porcentajes proporcionan una representación cuantitativa de las percepciones de los encuestados en relación con sus niveles individuales de estrés esto se puede visualizar en la Figura 33.

Con la idea de obtener mayor precisión y utilizando una escala del 1 al 5 señala tu nivel de estrés, donde (1) es poco y (5) mucho.
121 respuestas

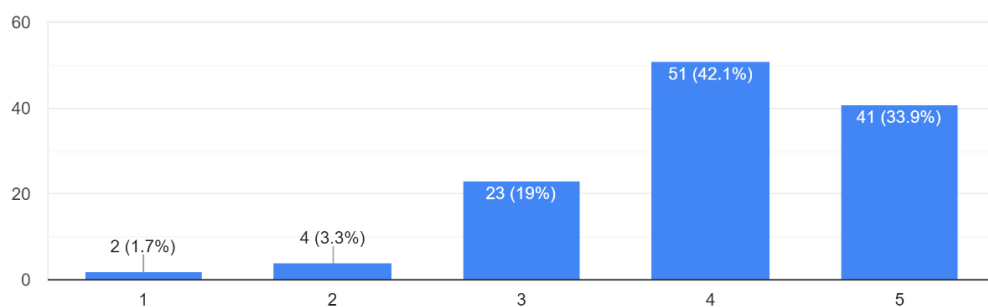


Figura 33. Resultados del formulario SISCO primera pregunta

Como resultado del segundo indicador del instrumento correspondiente la frecuencia con la que la sobrecarga de tareas y trabajos escolares que estresa a los estudiantes se obtuvo que, un total de 37 estudiantes (30.58%) indicó que "Algunas Veces", mientras que 10 estudiantes (8.26%) respondieron "Casi Nunca". Por otro lado, 44 estudiantes (36.36%) seleccionaron la opción "Casi Siempre". Un grupo menor, compuesto por 7 estudiantes (5.79%), mencionó experimentar dichas situaciones "Rara Vez". Por último, 23 estudiantes (19.01%) indicaron que experimentan estos eventos "Siempre". En el análisis de la pregunta, se destaca que la opción "Casi Siempre" fue la más seleccionada, con 44 respuestas, representando aproximadamente el 36.36% del total de encuestados. Este resultado sugiere que un segmento significativo de la población estudiantil experimenta de manera frecuente niveles de estrés asociados con la carga académica diaria, esto se puede visualizar en la Figura 34.

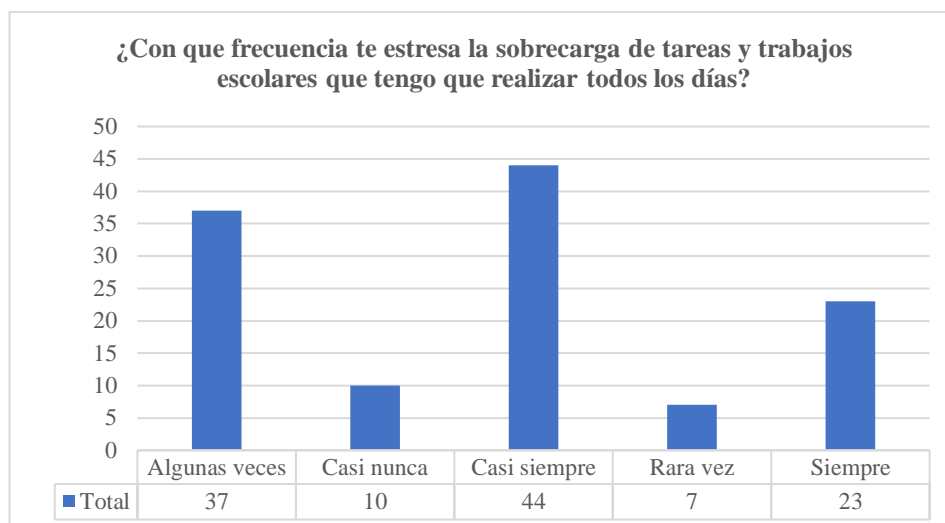


Figura 34. Resultados del formulario SISCO segunda pregunta

Con respecto al tercer indicador sobre la frecuencia con la que la forma de evaluación de los profesores que estresa a los estudiantes se obtuvo que, 31 estudiantes (25.62%) indicaron que esto ocurre "Algunas Veces", 13 estudiantes (10.74%) seleccionaron la opción "Casi Nunca", 36 estudiantes (29.75%) afirmaron que esto sucede "Casi Siempre", 16 estudiantes (13.22%) señalaron que es "Rara Vez", y 25 estudiantes (20.66%) indicaron que se sienten estresados "Siempre". El análisis destaca que la opción "Casi Siempre" obtuvo la mayor marcación, representando aproximadamente el 29.75% del total de respuestas. Este resultado sugiere que un porcentaje significativo de la población estudiantil experimenta estrés relacionado con la forma de evaluación de los profesores en la mayoría de las situaciones de evaluación, esto se puede observar en la Figura 35.

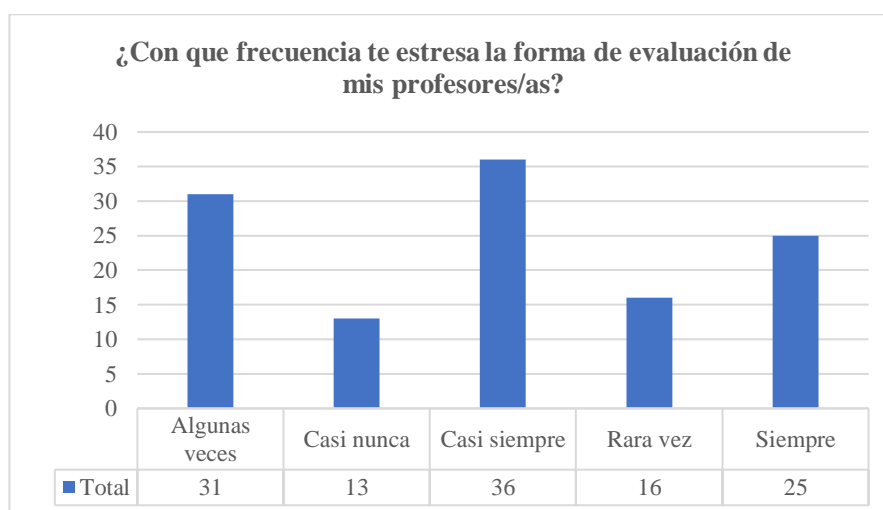


Figura 35. Resultados del formulario SISCO tercera pregunta

En relación con la indagación sobre el estrés derivado del tipo de trabajo asignado por los profesores se obtuvo que 26 estudiantes (21.49%) indicaron que esto ocurre "Algunas Veces", 18 estudiantes (14.88%) seleccionaron la opción "Casi Nunca", 30 estudiantes (24.79%) afirmaron que esto sucede "Casi Siempre", 10 estudiantes (8.26%) señalaron que es "Rara Vez", y 28 estudiantes (23.14%) indicaron que se sienten estresados "Siempre". De manera destacada, la opción "Casi Siempre" obtuvo la mayor marcación, representando aproximadamente el 24.79% del total de respuestas. Este resultado sugiere que un segmento significativo de la población estudiantil experimenta estrés asociado con el tipo de trabajos asignados por los profesores en la mayoría de las ocasiones esto se puede visualizar en la Figura 36.

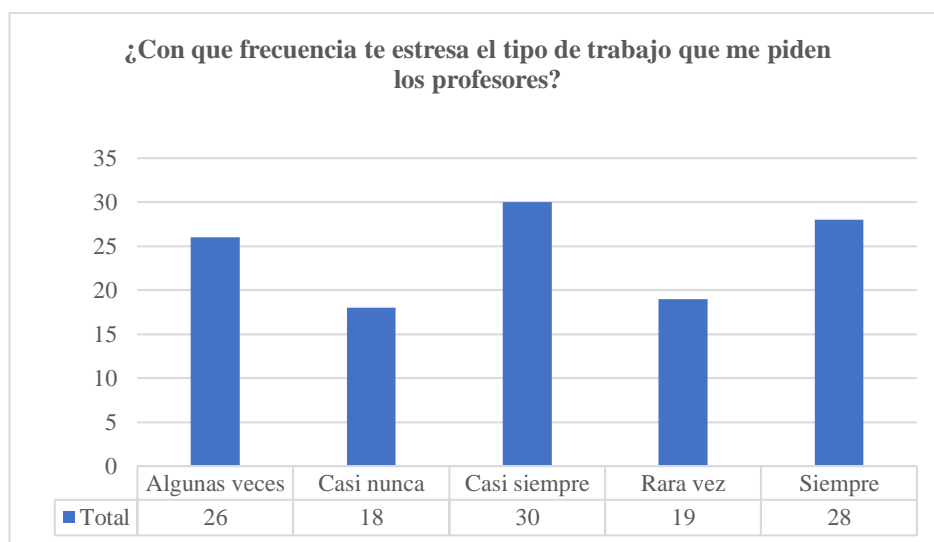


Figura 36. Resultados del formulario SISCO cuarta pregunta

En el análisis de la pregunta relacionada con el estrés asociado al tiempo limitado para realizar los trabajos asignados, se observa que 32 estudiantes (26.45%) indicaron que esto ocurre "Algunas Veces", 9 estudiantes (7.44%) seleccionaron la opción "Casi Nunca", 39 estudiantes (32.23%) afirmaron que esto sucede "Casi Siempre", 13 estudiantes (10.74%) señalaron que es "Rara Vez", y 28 estudiantes (23.14%) indicaron que se sienten estresados "Siempre". De manera destacada, la opción "Casi Siempre" obtuvo la mayor marcación, representando aproximadamente el 32.23% del total de respuestas. Este resultado sugiere que un porcentaje significativo de la población estudiantil experimenta estrés derivado del tiempo limitado para completar las tareas académicas en la mayoría de las ocasiones, esto se puede observar en la Figura 37.

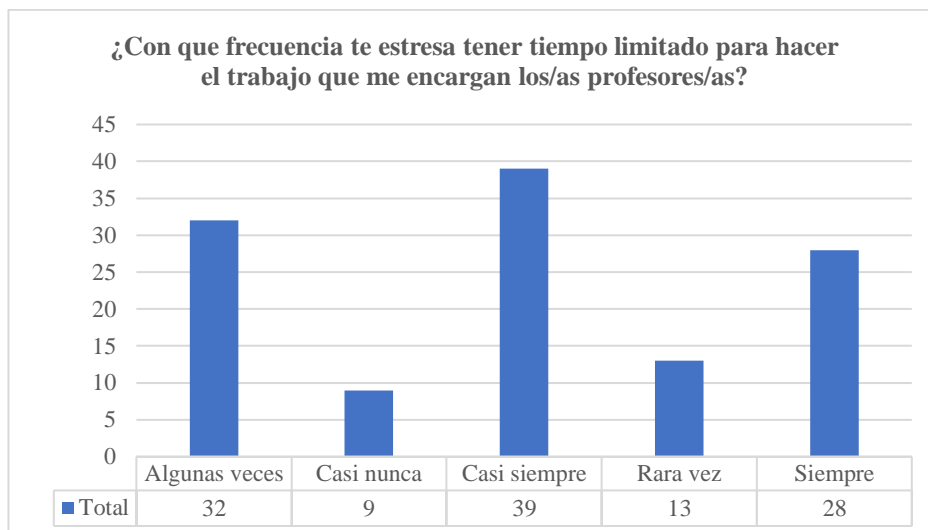


Figura 37. Resultados del formulario SISCO quinta pregunta

En el análisis de la pregunta relacionada con las reacciones asociadas a la fatiga crónica cuando los estudiantes experimentan, 30 estudiantes (24.79%) indicaron experimentar fatiga crónica "Algunas Veces", 19 estudiantes (15.70%) seleccionaron la opción "Casi Nunca", 31 estudiantes (25.62%) afirmaron que esto sucede "Casi Siempre", 6 estudiantes (4.96%) señalaron que es "Nunca", 22 estudiantes (a 18.18%) indicaron que se presenta "Rara Vez", y 13 estudiantes (10.74%) afirmaron que se manifiesta "Siempre". Destaca que la opción "Casi Siempre" obtuvo la mayor marcación, representando aproximadamente el 25.62% del total de respuestas, esto se puede observar en la Figura 38.

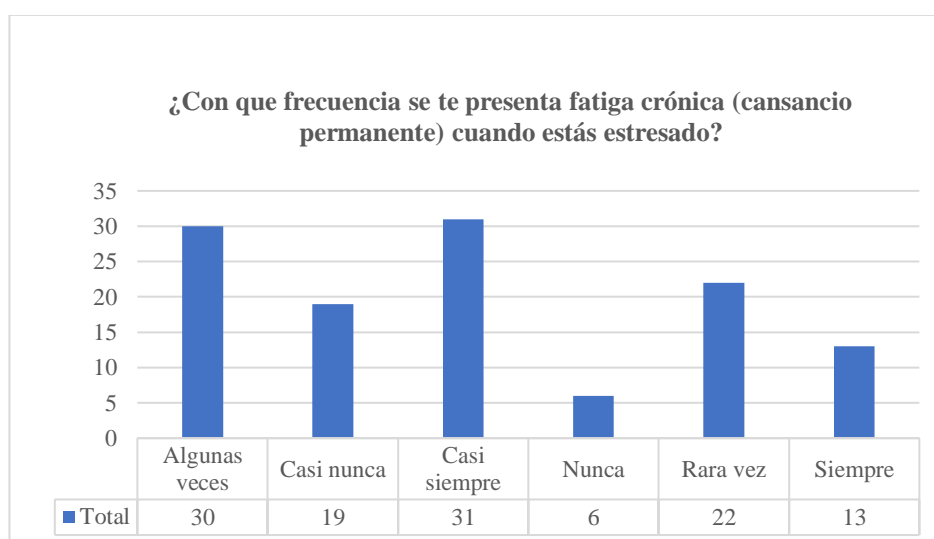


Figura 38. Resultados del formulario SISCO sexta pregunta

En el análisis de la pregunta relacionada con la frecuencia de problemas de concentración, 34 estudiantes (28.10%) indicaron experimentar problemas de concentración "Algunas Veces", 15 estudiantes (12.40%) seleccionaron la opción "Casi Nunca", 31 estudiantes (25.62%) afirmaron que esto sucede "Casi Siempre", 2 estudiantes (1.65%) señalaron que es "Nunca", 23 estudiantes (19.01%) indicaron que se presenta "Rara Vez", y 16 estudiantes (13.22%) afirmaron que se manifiesta "Siempre". Destaca que la opción "Algunas Veces" obtuvo la mayor marcación, representando aproximadamente el 28.10% del total de respuestas. Este resultado sugiere que un porcentaje significativo de la población estudiantil experimenta problemas de concentración en situaciones de estrés en ocasiones específicas, esto se puede observar en la Figura 39.

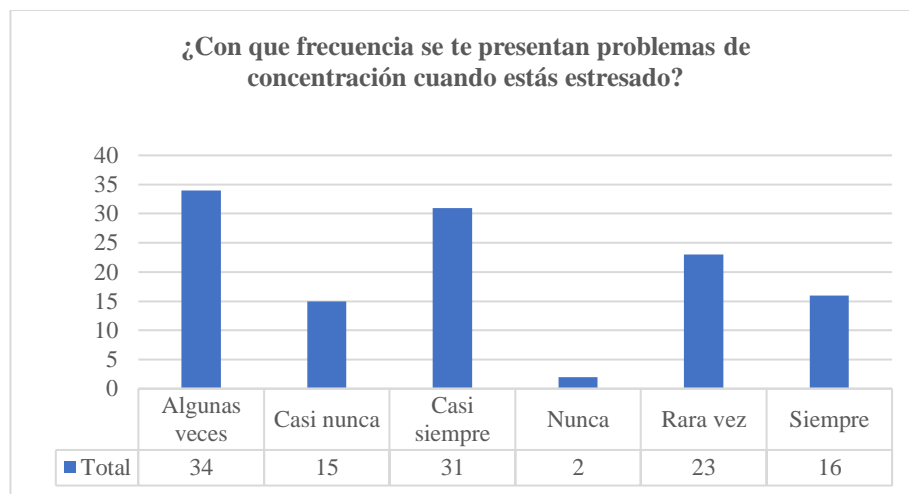


Figura 39. Resultados del formulario SISCO séptima pregunta

En el análisis de la pregunta relacionada con la frecuencia de desgano para realizar labores escolares cuando los estudiantes están estresados, 31 estudiantes (25.62%) indicaron experimentar desgano "Algunas Veces", 20 estudiantes (16.53%) seleccionaron la opción "Casi Nunca", 24 estudiantes (19.83%) afirmaron que esto sucede "Casi Siempre", 6 estudiantes (4.96%) señalaron que es "Nunca", 22 estudiantes (18.18%) indicaron que se presenta "Rara Vez", y 18 estudiantes (14.88%) afirmaron que se manifiesta "Siempre". Destaca que la opción "Casi Siempre" obtuvo la mayor marcación, representando aproximadamente el 19.83% del total de respuestas. Este resultado sugiere que un porcentaje significativo de la población estudiantil experimenta desgano para realizar labores escolares en la mayoría de las situaciones de estrés, esto se puede visualizar en la Figura 40.

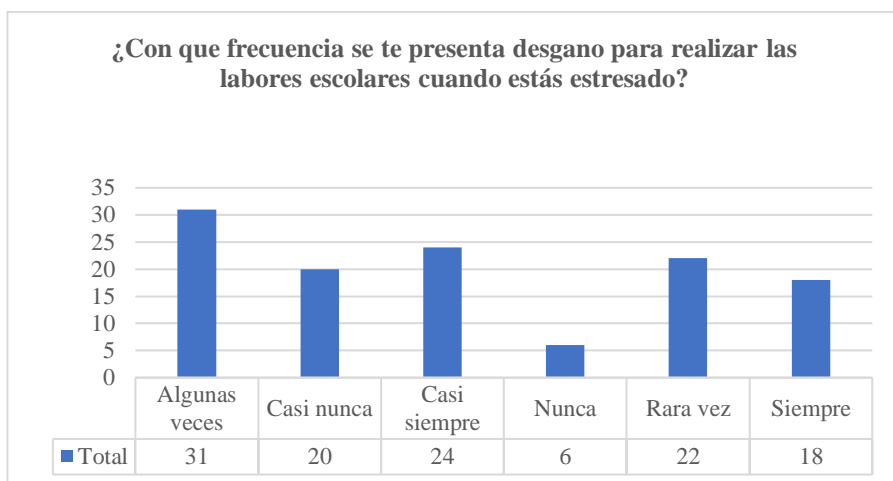


Figura 40. Resultados del formulario SISCO octava pregunta

Con la clasificación de estas preguntas se tabularon los datos, primero se codifico las respuestas a valores numéricos como se indica en la Tabla 37, posteriormente se realizó el promedio de cada respuesta, para clasificarlo en tres niveles de estrés bajo, moderado y alto. Esto se determinó de acuerdo con el baremo indicativo que se analizó en INVENTARIO SISCO SV-21 [22], el cual determina los rangos para determinar el nivel de estrés que se analiza en la Tabla 36 .

Tabla 36. Rangos de los resultados del inventario SISCO [22]

Nivel	Equivalente	Rango SISCO
Bajo	0	0 – 33%
Moderado	1	34% – 66%
Alto	2	67% - 100%

Tabla 37. Codificación de las respuestas a valores numéricos [22]

Nunca	Casi nunca	Rara vez	Algunas veces	Casi siempre	Siempre
N	CN	RV	AV	CS	S
0	1	2	3	4	5

Con la codificación realizada, se determinan los datos necesarios para realizar la etiquetación, la clase principal que se tiene en cuenta en este conjunto de datos es el nivel de estrés, y el etiquetada corresponderá al baremo indicativo que se estableció, es decir bajo, moderado y alto.

Tabla 38. Clase y Etiquetado de datos

CLASE	ETIQUETA
Nivel de estrés	Bajo
	Moderado
	Alto

Con estos parámetros se etiqueto la información que se obtuvo mediante el formulario aplicado. Teniendo como resultado el conjunto de datos que se puede analizar en el Anexo C.

c. Procesamiento y etiquetación del dataset

En el dataframe procesado anteriormente, es decir los datos de la actividad electromiográfica (EMG), actividad electrodérmica (EDA/GSR) y los datos obtenidos de la aplicación del formulario SISCO se realizó el proceso de etiquetación en conjunto analizando la relación entre los rangos de medición como se establece en la Tabla 33, los rangos establecidos en el formulario SISCO (Tabla 36), teniendo en cuenta esos aspectos se unifico en una sola base de datos toda esta información, con el formato de la Tabla 39, donde se establecieron las etiquetas en equivalentes, es decir para el nivel alto de estrés se estableció el 2, para el nivel moderado el 1 y para el nivel bajo de estrés el 0. La base de datos resultante junto con las etiquetas correspondiente se puede observar en el Anexo G.

Tabla 39. Clases de la base de datos

EMG	GSR	P1	P2	P3	P4	P5	P6	P7	P8	NIVEL
-----	-----	----	----	----	----	----	----	----	----	-------

d. Acondicionamiento del dataframe para el modelo de ML

Para realizar el acondicionamiento del dataset obtenido, se usó la librería de “*pandas*”, que permite transformar un archivo con extensión “.csv”, en un dataframe que se necesita para el entrenamiento del modelo de clasificación de Machine Learning. El formato del archivo del dataframe permite su modificación debido a que su formato es similar al de una tabla con filas y columnas.

Para transformar el archivo “.csv” en un dataframe y poder leerlo, se usó la herramienta “*pd.read_csv*”, en donde se especifica el lenguaje de programación que se utiliza, en este caso Python.

```
# Cargar los datos desde el archivo CSV
data = pd.read_csv('Shirley_bd_ML_CSV.csv', sep=';', engine='python')
print(data.head(3))
```

Figura 41. Carga de los datos desde el archivo CSV

e. Entrenamiento del algoritmo de machine learning

- **Separación de datos de entrenamiento**

Para el entrenamiento se debe separar el dataframe en 2, donde “x” hace referencia a la variable independiente y “y” corresponde a la variable dependiente, a este se le asigna la columna “Nivel”, que contiene los tres niveles de estrés, esto corresponde a las 3 clases que usara el modelo de predicción.

```
# Dividir los datos en características (X) y etiquetas (y)
X = data.drop('NIVEL', axis=1)
y = data['NIVEL']
```

Figura 42. Separación del dataframe

Para el entrenamiento se usó el método `train_test_split` (división de datos) de `scikit-learn`, que fue seleccionado en la sección 3.2.3. Este método consiste en separar el dataframe, para así destinar la cantidad de datos necesarios para entrenamiento y testeo del modelo. Para que esto fuese posible se importó la herramienta “`train_test_split`” y se estableció del conjunto de datos el 70% destinados al entrenamiento y el 30% destinados al conjunto de prueba [64].

Al asignar el 70% de los datos al conjunto de entrenamiento, se proporciona al modelo una cantidad sustancial de información para aprender patrones y relaciones en los datos.

El 30% restante se reserva para el conjunto de prueba, lo que permite evaluar el rendimiento del modelo en datos no vistos durante el entrenamiento. Este conjunto de prueba actúa como una medida objetiva de la capacidad del modelo para generalizar.

El método `train_test_split` de `scikit learn` tiene la siguiente estructura que se puede observar en la Figura 43, en la etapa 0 se tiene en cuenta la organización de los datos de manera que se ajusten al formato adecuado para la división de entrenamiento y prueba. En el caso de `scikit-learn`, esto implica dividir tu conjunto de datos completo en dos partes: Características (Features) y Objetivo (Target). Posteriormente, en la etapa 1 se separa el conjunto de datos en dos partes: una destinada al entrenamiento y otra a la prueba. Esto involucra la selección aleatoria de aproximadamente el 70% de

las filas para incluirlas en el conjunto de entrenamiento, mientras que el 30% restante se asigna al conjunto de prueba. Es importante notar que los colores en las secciones de "Features" y "Target" señalan dónde se ubicarán sus datos ("X_train", "X_test", "y_train", "y_test") durante una división específica de entrenamiento y prueba. En la etapa 2 se entrena el modelo en el conjunto de entrenamiento. Esto corresponde a "X_train" e "y_train" en la imagen. Finalmente, evalúa el rendimiento del modelo al probarlo en el conjunto de prueba, identificado como "X_test" e "y_test" en la imagen [65].

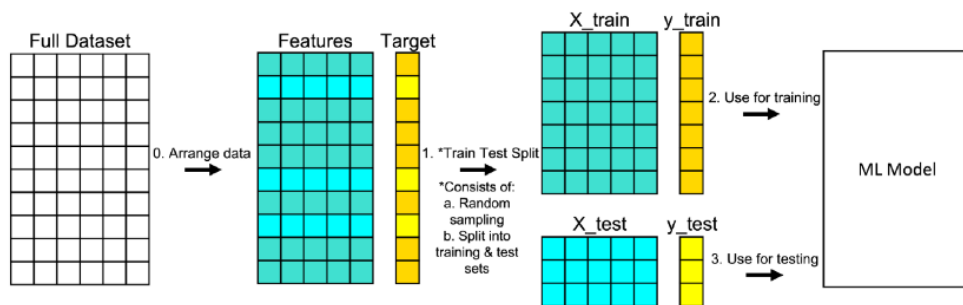


Figura 43. Estructura del método train test Split [65]

Para la comparación primero se debe importar las bibliotecas y módulos necesarios, se importan librerías como “*pandas*” para manipulación de datos, modelos de aprendizaje automático de “*scikit-learn*” y “*XGBoost*”, así como herramientas adicionales para el procesamiento de desequilibrios en las clases (*imblearn*).

Se dividen en características (X) y etiquetas (Y). Luego, se divide el conjunto de datos en conjuntos de entrenamiento y prueba. Se utiliza la técnica SMOTE (Synthetic Minority Over-sampling Technique) para abordar el desequilibrio de clases en el conjunto de entrenamiento. Esto se hace para cada conjunto de entrenamiento y prueba por separado.

```
# Seleccionar el método de resampling solo para el conjunto de entrenamiento
oversampler = SMOTE(sampling_strategy='auto', random_state=42)
X_train_resampled, y_train_resampled = oversampler.fit_resample(X_train, y_train)
```

Figura 44. Selección de método Resampling

f. Evaluación del algoritmo

Se eligen varios modelos de clasificación, para realizar pruebas con cada algoritmo seleccionado. Para ello es importante llamar a los algoritmos seleccionados como se observa en la Figura 45. Cada algoritmo seleccionado se usa mediante la función 'Classifier', excepto el algoritmo de regresión logística que se usa con la función 'LogisticRegression()' y el algoritmo Gaussian Naive Bayes con la función 'GaussianNB()'.

```
# Seleccionar modelos
models = {
    'Random Forest': RandomForestClassifier(),
    'K Nearest Neighbours': KNeighborsClassifier(),
    'Gradient Boosting': GradientBoostingClassifier(),
    'Extra Trees': ExtraTreesClassifier(),
    'Adaboost': AdaBoostClassifier(),
    'Decision Tree': DecisionTreeClassifier(),
    'XGBoost': XGBClassifier(),
    'MLP': MLPClassifier(max_iter=1000),
    'Linear SVM': SVC(kernel='linear', probability=True),
    'Logistic Regression': LogisticRegression(),
    'Gaussian Naive Bayes': GaussianNB()
}
```

Figura 45. Algoritmos seleccionados

Se realiza la comparación de modelos utilizando la técnica de validación cruzada que tiene la estructura que se puede analizar en la Figura 46, esto se realiza con el conjunto de entrenamiento y de prueba, con la función 'cross_val_score'.

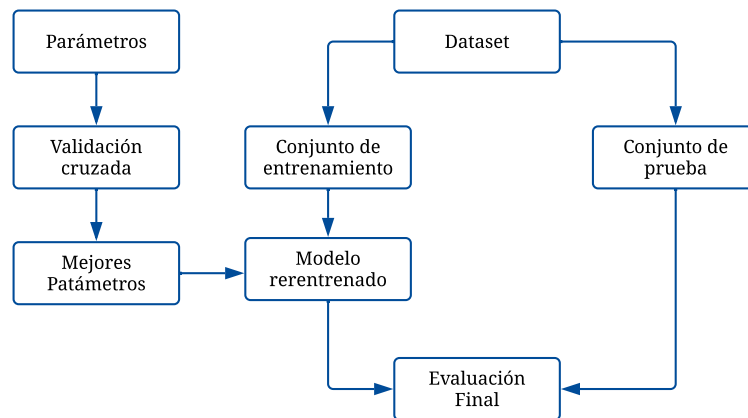


Figura 46. Flujograma de la técnica de validación cruzada basado en [66]

- **Matriz de confusión.** La evaluación del algoritmo se realiza mediante tres métricas que corresponden a 'Precision (P)', 'Recall (R)' y 'F1 Score(F1)'. Estas métricas se

obtienen mediante las matrices de confusión de cada algoritmo las cuales se utilizan para ilustrar el rendimiento del algoritmo, esta se basa en cuatro valores: Verdaderos Positivos (VP), Falsos Positivos (FN), Verdaderos Negativos (VN) y los Falsos Negativos (FN). La matriz será de 3x3 debido a que existen 3 clases para su clasificación. Y su estructura se divide como en la Figura 47.

		Predicción		
		A	B	C
Actual	A	AA	AB	AC
	B	BA	BB	BC
	C	CA	CB	CC

		Predicción		
		A	B	C
Actual	A	VP	FN	
	B		FN	VN
	C		FP	VN

		Predicción		
		A	B	C
Actual	A	VN	FP	VN
	B	FN	VP	FN
	C	VN	FP	VN

		Predicción		
		A	B	C
Actual	A	VN	VN	FP
	B	VN	VN	FP
	C	FN	FN	VP

Figura 47. Estructura de la matriz de confusión basado en [47]

Con el comando `“confusion_matrix (test_y,svc.predict(test_x_vector),labels = ['bajo','moderado,'alto])”`, se obtienen las matrices de confusión en Python que se observan a continuación.

```

Random Forest: Matriz de Confusión:
[[ 16 23  0]
 [  1 908 42]
 [  0 46 134]]
XGBoost: Matriz de Confusión:
[[ 21 18  0]
 [  4 887 60]
 [  0 26 154]]
K Nearest Neighbours: Matriz de Confusión:
[[ 34  5  0]
 [ 37 746 168]
 [  0 12 168]]
MLP: Matriz de Confusión:
[[ 33  6  0]
 [  0 898 53]
 [  0 13 167]]
Gradient Boosting: Matriz de Confusión:
[[ 23 16  0]
 [ 11 839 101]
 [  0 19 161]]
Linear SVM: Matriz de Confusión:
[[ 37  2  0]
 [  1 816 134]
 [  0  0 180]]
Extra Trees: Matriz de Confusión:
[[ 16 23  0]
 [  0 920 31]
 [  0 41 139]]
Logistic Regression: Matriz de Confusión:
[[ 37  2  0]
 [  1 828 122]
 [  0  0 180]]
Adaboost: Matriz de Confusión:
[[ 20 19  0]
 [  0 891 60]
 [  0 19 161]]
Gaussian Naive Bayes: Matriz de Confusión:
[[ 31  8  0]
 [ 22 797 132]
 [  0  6 174]]
Decision Tree: Matriz de Confusión:
[[ 20 19  0]
 [ 38 788 125]
 [  0 59 121]]

```

Figura 48. Matrices de confusión de cada algoritmo

Para determinar el algoritmo de ML se calculan los valores de las métricas mediante las ecuaciones **¡Error! No se encuentra el origen de la referencia.**) para la precisión, **REF_Ref155545550 \h ¡Error! No se encuentra el origen de la referencia.**) para el Recall y para el F1-Score **¡Error! No se encuentra el origen de la referencia.**). Los valores de las métricas de evaluación que corresponden a cada algoritmo se observan en la Tabla 40.

Tabla 40. Desempeño de los algoritmos de clasificación

Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
Random Forest	16	1130	0	30	Bajo	1	0,348	0,516
	1130	150	90	43	Moderado	0,926	0,963	0,944
	889	960	50	46	Alto	0,947	0,951	0,949
	TOTAL					0,958	0,754	0,803
Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
K Nearest Neighbours	34	1094	37	5	Bajo	0,479	0,872	0,618
	746	202	17	205	Moderado	0,978	0,784	0,87
	168	822	168	12	Alto	0,5	0,933	0,651
	TOTAL					0,652	0,863	0,713
Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1

Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
Gradient Boosting	23	1120	11	16	Bajo	0,676	0,59	0,63
	839	184	35	112	Moderado	0,96	0,882	0,919
	161	889	101	19	Alto	0,615	0,894	0,729
	TOTAL					0,75	0,789	0,759
Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
Extra Trees	16	1131	0	23	Bajo	1	0,41	0,582
	920	155	64	31	Moderado	0,935	0,967	0,951
	139	959	31	41	Alto	0,818	0,772	0,794
	TOTAL					0,918	0,717	0,776
Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
Adaboost	20	1131	0	19	Bajo	1	0,513	0,678
	891	181	38	60	Moderado	0,959	0,937	0,948
	161	930	60	19	Alto	0,729	0,894	0,803
	TOTAL					0,896	0,781	0,81
Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
Decision Tree	20	1093	38	19	Bajo	0,345	0,513	0,412
	788	141	78	163	Moderado	0,91	0,829	0,867
	121	865	125	59	Alto	0,492	0,672	0,568
	TOTAL					0,582	0,671	0,616
Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
XGBoost	21	1127	4	18	Bajo	0,84	0,538	0,656
	887	175	44	64	Moderado	0,953	0,933	0,943
	154	930	60	26	Alto	0,72	0,856	0,782
	TOTAL					0,837	0,776	0,794
Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
MLP	33	1131	0	6	Bajo	1	0,846	0,917
	898	200	19	53	Moderado	0,979	0,944	0,961
	167	937	53	13	Alto	0,759	0,928	0,835
	TOTAL					0,913	0,906	0,904
Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
Linear SVM	37	1130	1	2	Bajo	0,974	0,949	0,961
	816	217	2	135	Moderado	0,998	0,858	0,923
	180	856	134	0	Alto	0,573	1	0,729
	TOTAL					0,848	0,936	0,871
Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
Logistic Regression	37	1130	1	2	Bajo	0,974	0,949	0,961
	828	217	2	123	Moderado	0,998	0,871	0,93
	180	868	122	0	Alto	0,596	1	0,747
	TOTAL					0,856	0,94	0,879
Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1

Algoritmo	VP	VN	FP	FN	CLASE	P	R	F1
Gaussian Naive Bayes	31	1109	22	8	Bajo	0,585	0,795	0,674
	797	205	14	154	Moderado	0,983	0,838	0,905
	174	858	132	6	Alto	0,569	0,967	0,716
	TOTAL						0,712	0,867

- **Valores de precisión.** Con la técnica de validación cruzada mediante la función ‘scoring=make_scorer(accuracy_score)’, se determina la precisión de entrenamiento, precisión de testeo y la precisión de prueba como se observa en la Figura 49.

```

Random Forest: Precisión de validación cruzada Entrenamiento= 97.10%----
Random Forest: Precisión de validación cruzada Testeo= 97.02%
K Nearest Neighbours: Precisión de validación cruzada Entrenamiento= 91.42%----
K Nearest Neighbours: Precisión de validación cruzada Testeo= 90.26%
Gradient Boosting: Precisión de validación cruzada Entrenamiento= 94.85%----
Gradient Boosting: Precisión de validación cruzada Testeo= 95.23%
Extra Trees: Precisión de validación cruzada Entrenamiento= 97.89%----
Extra Trees: Precisión de validación cruzada Testeo= 97.97%
Adaboost: Precisión de validación cruzada Entrenamiento= 90.45%----
Adaboost: Precisión de validación cruzada Testeo= 86.89%
Decision Tree: Precisión de validación cruzada Entrenamiento= 90.54%----
Decision Tree: Precisión de validación cruzada Testeo= 90.92%
XGBoost: Precisión de validación cruzada Entrenamiento= 96.98%----
XGBoost: Precisión de validación cruzada Testeo= 97.13%
MLP: Precisión de validación cruzada Entrenamiento= 95.96%----
MLP: Precisión de validación cruzada Testeo= 96.53%
Linear SVM: Precisión de validación cruzada Entrenamiento= 92.27%----
Linear SVM: Precisión de validación cruzada Testeo= 92.82%
Logistic Regression: Precisión de validación cruzada Entrenamiento= 92.66%----
Logistic Regression: Precisión de validación cruzada Testeo= 93.38%
Gaussian Naive Bayes: Precisión de validación cruzada Entrenamiento= 91.60%----
Gaussian Naive Bayes: Precisión de validación cruzada Testeo= 92.43%
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
Random Forest: Precisión en conjunto de prueba = 89.91%
K Nearest Neighbours: Precisión en conjunto de prueba = 81.03%
Gradient Boosting: Precisión en conjunto de prueba = 87.44%
Extra Trees: Precisión en conjunto de prueba = 90.85%
Adaboost: Precisión en conjunto de prueba = 91.62%
Decision Tree: Precisión en conjunto de prueba = 78.80%
XGBoost: Precisión en conjunto de prueba = 90.77%
MLP: Precisión en conjunto de prueba = 93.08%
Linear SVM: Precisión en conjunto de prueba = 88.29%
Logistic Regression: Precisión en conjunto de prueba = 89.32%
Gaussian Naive Bayes: Precisión en conjunto de prueba = 85.64%

```

Figura 49. Precisión de entrenamiento, testeo y prueba

g. Selección del algoritmo de machine Learning

Para seleccionar el algoritmo adecuado, se analiza el resultado que se obtuvo al analizar las métricas de la Tabla 40. De esa tabla se extrae la información que corresponde al modelo y los datos de precisión, recall y F1-Score de ello se realizó un análisis que se observa en la Figura 50. Donde se tiene que el algoritmo Random Forest tiene 0.96 de precisión, 0.75 de recall y 0.8 de F1-Score, destacando del resto de algoritmos.

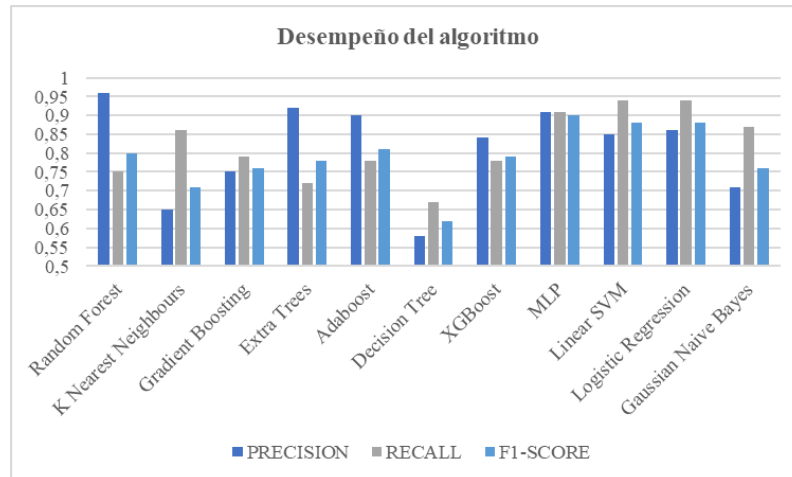


Figura 50. Desempeño del algoritmo

El resultado de comparación de los modelos de aprendizaje automático en base a la precisión de entrenamiento, testeo y prueba se visualiza en la Tabla 41, en la Figura 50 se analiza la capacidad de cada modelo que se ha sometido al entrenamiento de los cuales destaca el algoritmo de Random Forest con el 97.10% en entrenamiento el 97.02% en el testeo y el 90.9% en la etapa de prueba.

Tabla 41. Resultado aplicando la técnica de validación cruzada

Algoritmo	Entrenamiento	Testeo	Prueba
Random Forest	97,1	97,02	90,9
K Nearest Neighbours	91,42	90,26	81,03
Gradient Boosting	94,85	95,23	87,44
Extra Trees	96,89	96,97	90,85
Adaboost	90,45	86,89	91,62
Decision Tree	90,54	90,92	78,8
XGBoost	96,98	97,01	90,77
MLP	95,96	96,53	93,08
Linear SVM	92,27	92,82	88,29
Logistic Regression	92,66	93,38	89,32
Gaussian Naive Bayes	91,68	92,43	85,64

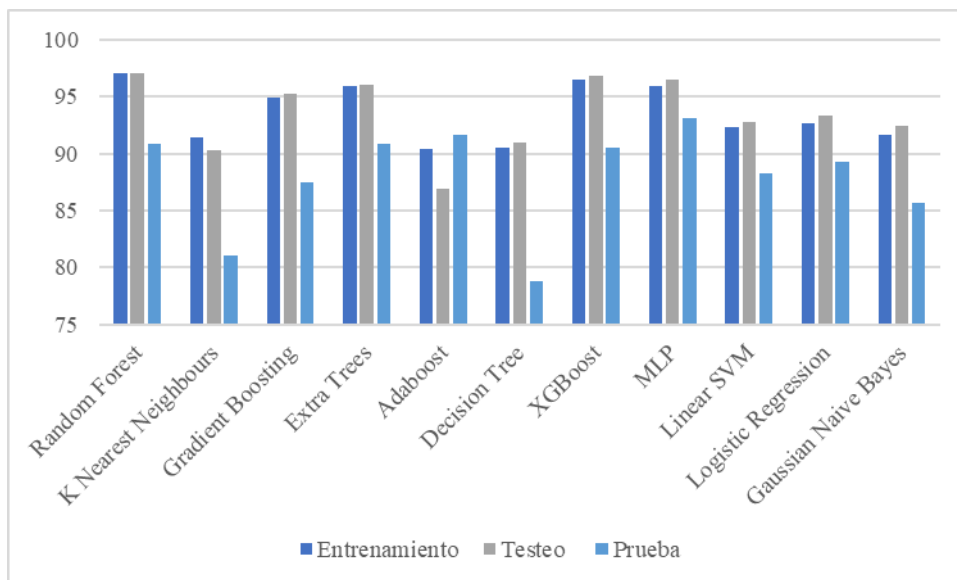


Figura 51. Precisión sobre el entrenamiento del algoritmo

Con el análisis realizado se definió que el mejor modelo para ser usado en el entrenamiento es el Random Forest. Finalmente se elige el modelo que mayor precisión tuvo con la función `models[selected_model_name]` y se entrena el algoritmo seleccionado con la siguiente función.

```
selected_pipeline = make_pipeline(StandardScaler(), selected_model)
selected_pipeline.fit(X_train_resampled, y_train_resampled)
```

Figura 52. Entrenamiento del algoritmo Random Forest

h. Ajuste del modelo Random Forest

Con el análisis que se realizó con anterioridad se eligió el modelo Random Forest, es decir bosques aleatorios, debido a que este obtuvo mejores resultados con respecto a los otros modelos analizados. Se debe tener en cuenta que este modelo es una colección de múltiples árboles de decisión que trabajan en conjunto para realizar predicciones. Cada árbol se entrena en una submuestra aleatoria de los datos de entrenamiento, y también puede considerar diferentes subconjuntos de características. Esto se conoce como "ensamble".

Para el entrenamiento de este modelo es importante definir dos parámetros, los estimadores que se refiere a los árboles de decisión individuales que componen el bosque y los criterios, dentro de los criterios se tiene [67]:

"mse" (Mean Squared Error): Este es el criterio predeterminado para regresión en Random Forest. Se utiliza para minimizar la varianza dentro de cada hoja del árbol. En cada nodo, se calcula el promedio de los valores de la variable objetivo y se mide la diferencia entre estos valores y el valor predicho por el nodo. La división que minimiza esta diferencia se selecciona.

"gini" y "entropy": Estos son los criterios comunes para clasificación. Se utilizan para medir la impureza de un nodo. En cada nodo, se calcula la impureza antes y después de la división, y la diferencia se utiliza para evaluar la calidad de la división. En clasificación, gini se refiere a la impureza de Gini, mientras que entropy se refiere a la ganancia de información.

Los criterios que se usaron fueron "Gini" y "entropy", no se usó "mse" debido a que este se usa para regresiones y no clasificación.

Para el ajuste del modelo, se empieza con un número de 25 estimadores con respecto al criterio "gini", con este número de estimadores se obtuvo una exactitud de entrenamiento de 0.91 y una exactitud de testeo de 0.90 como se observa en la Figura 53, estos valores son considerados bajos en el ajuste de un modelo.



```
# Crear el modelo para ajustar
from sklearn.ensemble import RandomForestClassifier
bosque = RandomForestClassifier(n_estimators=25,
                               criterion='gini',
                               max_features='sqrt',
                               max_depth=10)

bosque.fit(X_train_std, y_train)

# Precision global de clasificación corecta
print('Train Accuracy : %.5f' % bosque.score(X_train_std, y_train))
print('Test Accuracy : %.5f' % bosque.score(X_test_std, y_test))

Train Accuracy: 0.91224
Test Accuracy: 0.901468
```

Figura 53. Ajuste "gini" Random Forest con 25 estimadores

Respecto al criterio "entropy" con un número de 25 estimadores se obtuvo una exactitud de entrenamiento de 0.90 y una exactitud de testeo de 0.89 como se observa en la Figura 54, como estos valores se consideran bajos, se continua aumentando el número de estimadores.

```
0s # Crear el modelo para ajustar
from sklearn.ensemble import RandomForestClassifier
bosque = RandomForestClassifier(n_estimators=25,
                              criterion='entropy',
                              max_features='sqrt',
                              max_depth=10)

bosque.fit(X_train_std, y_train)

# Precision global de clasificación correcta
print('Train Accuracy : %.5f' % bosque.score(X_train_std, y_train))
print('Test Accuracy : %.5f' % bosque.score(X_test_std, y_test))

Train Accuracy: 0.90142
Test Accuracy: 0.89021
```

Figura 54. Ajuste “entropy” Random Forest con 25 estimadores

Los resultados con 25 estimadores no fueron los ideales, por esa razón se procede a evaluar con un numero de 50 estimadores en base a los criterios de “gini” y “entropy”, con este número de estimadores se obtuvo una exactitud de entrenamiento de 0.94 y 0.92 en el testeo como se observa en la Figura 55.

```
0s # Crear el modelo para ajustar
from sklearn.ensemble import RandomForestClassifier
bosque = RandomForestClassifier(n_estimators=50,
                              criterion='entropy',
                              max_features='sqrt',
                              max_depth=10)

bosque.fit(X_train_std, y_train)

# Precision global de clasificación correcta
print('Train Accuracy : %.5f' % bosque.score(X_train_std, y_train))
print('Test Accuracy : %.5f' % bosque.score(X_test_std, y_test))

Train Accuracy: 0.94193
Test Accuracy: 0.92571
```

Figura 55. Ajuste “entropy” Random Forest con 50 estimadores

Para obtener un mejor ajuste se evaluó con un numero de 60 estimadores, en base a los criterios establecidos se obtuvo una exactitud de entrenamiento de 0.97 y 0.97 de testeo como se observa en la Figura 56 , estos valores son considerados como ideales para establecer el ajuste del modelo de Random Forest. Para establecer que este número de estimadores es el ideal se evaluó con un numero de 75 y 100 estimadores como se observa en la Figura 57 y Figura 58 respectivamente. Se obtuvieron los mismos valores de exactitud que se obtuvo cuando se evaluó con 60 estimadores, es decir que después de los 60 estimadores los valores de entrenamiento y testeo se repiten. Es por esta razón que se elige el uso de 60 estimadores para que no haya un sobreajuste en el modelo seleccionado.

```
0s ✓ # Crear el modelo para ajustar
from sklearn.ensemble import RandomForestClassifier
bosque = RandomForestClassifier(n_estimators=60,
                               criterion='gini',
                               max_features='sqrt',
                               max_depth=10)

bosque.fit(X_train_std, y_train)

# Precision global de clasificación correcta
print('Train Accuracy : %.5f' % bosque.score(X_train_std, y_train))
print('Test Accuracy : %.5f' % bosque.score(X_test_std, y_test))

Train Accuracy: 0.97123
Test Accuracy: 0.97021
```

Figura 56. Ajuste Random Forest con 60 estimadores

```
0s ✓ # Crear el modelo para ajustar
from sklearn.ensemble import RandomForestClassifier
bosque = RandomForestClassifier(n_estimators=75,
                               criterion='gini',
                               max_features='sqrt',
                               max_depth=10)

bosque.fit(X_train_std, y_train)

# Precision global de clasificación correcta
print('Train Accuracy : %.5f' % bosque.score(X_train_std, y_train))
print('Test Accuracy : %.5f' % bosque.score(X_test_std, y_test))

Train Accuracy: 0.97123
Test Accuracy: 0.97021
```

Figura 57. Ajuste Random Forest con 75 estimadores

```
0s ✓ # Crear el modelo para ajustar
from sklearn.ensemble import RandomForestClassifier
bosque = RandomForestClassifier(n_estimators=100,
                               criterion='gini',
                               max_features='sqrt',
                               max_depth=10)

bosque.fit(X_train_std, y_train)

# Precision global de clasificación correcta
print('Train Accuracy : %.5f' % bosque.score(X_train_std, y_train))
print('Test Accuracy : %.5f' % bosque.score(X_test_std, y_test))

Train Accuracy: 0.97123
Test Accuracy: 0.97021
```

Figura 58. Ajuste Random Forest con 100 estimadores

Para evaluar el entrenamiento del modelo se obtiene la región de decisión que se refiere a las áreas del espacio de características donde el modelo asigna una clase específica,

se tienen tres clases, estrés bajo “0”, estrés moderado “1” y estrés alto “2”, en la Figura 59 se observa la división del espacio de características en regiones donde se asigna una etiqueta específica. Cuando se combina un conjunto de árboles en un Random Forest, cada árbol vota por una clase y la clase final predicha por el bosque se determina por votación. La región de decisión del bosque es esencialmente la combinación de las regiones de decisión de todos los árboles individuales.

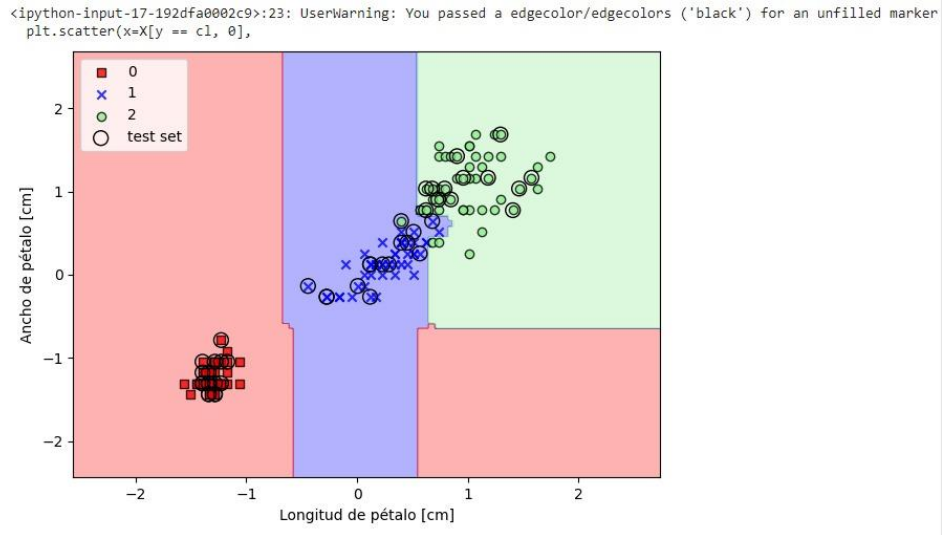


Figura 59. Región de decisión

Para la predicción con el algoritmo de Random Forest, se deben calcular los pesos de cada variable y así obtener la predicción aplicando el algoritmo.

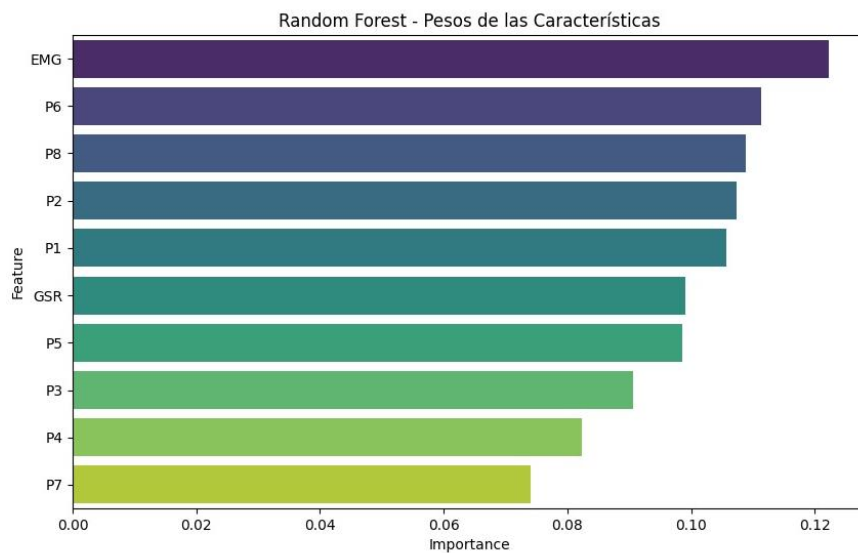


Figura 60. Importancia de los pesos de las variables con Random Forest


```

Random Forest - Pesos de las Características:
Feature Importance
0    EMG    0.122254
7    P6     0.111305
9    P8     0.108859
3    P2     0.107353
2    P1     0.105663
1    GSR    0.099065
6    P5     0.098477
4    P3     0.090649
5    P4     0.082306
8    P7     0.074068

```

Figura 61. Pesos de las variables

- Matriz de correlación.** Para obtener la matriz de correlación se usó el método *'corrcoef'*, que permite obtener la matriz de correlación entre una o más variables. Los valores obtenidos de correlación van entre 0 y 1, siendo 1 el más presente en cada variable, lo que significa que tiene una correlación positiva fuerte, es decir el valor de la primera variable provoca que la segunda aumente.

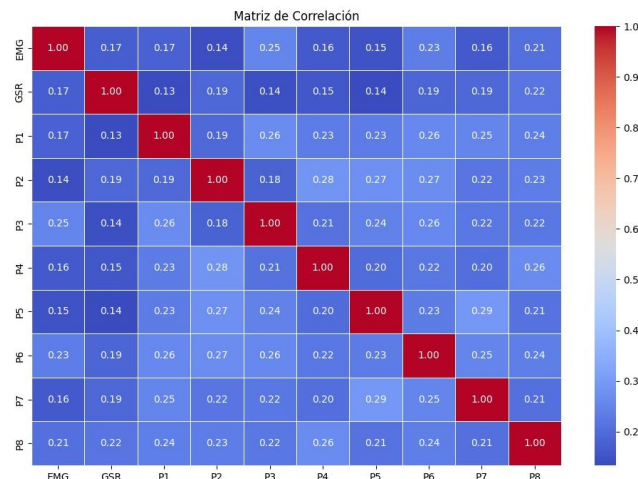


Figura 62. Mapa de calor de la matriz de correlación

```

Matriz de Correlación entre Características:
EMG  GSR  P1  P2  P3  P4  P5  P6  P7  P8
EMG  1.000000  0.168280  0.174873  0.139953  0.249290  0.162541  0.150296  0.232542  0.163795  0.207060
GSR  0.168280  1.000000  0.133424  0.187698  0.143206  0.153960  0.136558  0.187358  0.189510  0.217695
P1  0.174873  0.133424  1.000000  0.193994  0.263665  0.231587  0.229340  0.256878  0.249771  0.238263
P2  0.139953  0.187698  0.193994  1.000000  0.180288  0.283293  0.272343  0.266001  0.217740  0.234199
P3  0.249290  0.143206  0.263665  0.180288  1.000000  0.210536  0.243952  0.257939  0.222459  0.221608
P4  0.162541  0.153960  0.231587  0.283293  0.210536  1.000000  0.199463  0.223463  0.202280  0.264591
P5  0.150296  0.136558  0.229340  0.272343  0.243952  0.199463  1.000000  0.230153  0.290024  0.212330
P6  0.232542  0.187358  0.256878  0.266001  0.257939  0.223463  0.230153  1.000000  0.248720  0.237251
P7  0.163795  0.189510  0.249771  0.217740  0.222459  0.202280  0.290024  0.248720  1.000000  0.207700
P8  0.207060  0.217695  0.238263  0.234199  0.221608  0.264591  0.212330  0.237251  0.207700  1.000000

```

Figura 63. Matriz de correlación

La principal ventaja de Random Forest es que al promediar o votar sobre múltiples árboles, se reduce la tendencia a sobreajustar (overfitting) a datos específicos del conjunto de entrenamiento, y se mejora la generalización del modelo a nuevos datos. La combinación de las regiones de decisión de los árboles individuales contribuye a la robustez y rendimiento del modelo en comparación con un solo árbol de decisión.

i. Almacenamiento del modelo final

A continuación, se carga el modelo seleccionado en este caso el Random Forest desde el archivo en formato pickle (.pkl) y luego imprime o explora el modelo cargado. Primero se importa la biblioteca joblib, que se utiliza para cargar y guardar objetos de Python en formato binario. En este caso, se usará para cargar un modelo de Random Forest desde un archivo. Utilizando la función load de joblib, se carga el modelo desde el archivo llamado 'Random Forest_model.pkl'. El modelo debe haber sido guardado previamente en ese archivo utilizando la función joblib.dump().

j. Carga del modelo final

Para cargar el modelo final previamente entrenado se realizó un API en PHP que se basa en realizar una solicitud al script PHP con la función `response = requests.get(url_php_script)` y se realizó la carga del modelo donde se determinan las predicciones en base a las clases.

```
# URL del script PHP
url_php_script = 'https://stressappshirleybd.000webhostapp.com/model.php'

# Realizar solicitud al script PHP
response = requests.get(url_php_script)
data = response.json()

# Crear un DataFrame con los resultados
df_combined = pd.DataFrame(data, index=[0])
print(df_combined)

# Cargar el modelo previamente entrenado
modelo = joblib.load('Random Forest_model.pkl')

# Realizar predicciones en los datos combinados
predicciones = modelo.predict(df_combined.values.reshape(1, -1))

# Imprimir las predicciones
print("Predicciones:")
print(predicciones)

if np.any(predicciones == 0):
    print("Se detecto un estres bajo")
elif np.any(predicciones == 1):
    print("Se detecto un estres moderado")
else:
    print("Se detecto un estres alto")
```

Figura 64. Aplicación del modelo Random Forest

3.2.7 Desarrollo de la aplicación móvil

a. Instalación de Flutter

Para el desarrollo de la aplicación móvil vamos a usar el entorno de programación Visual Studio Code, y el lenguaje de programación Dart en el framework de Flutter como se analizó en la Tabla 29, para ello se descarga Flutter desde la página oficial <https://docs.flutter.dev/get-started/install> y se elige el sistema operativo que se está usando, en este caso Windows.

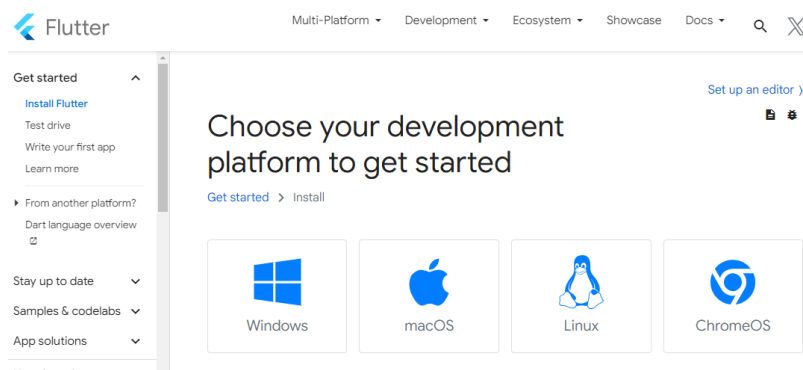


Figura 65. Sistema operativo para la instalación de flutter

Posteriormente elegimos el tipo de aplicación que se va a desarrollar, en este caso una aplicación móvil y descargamos el instalador.

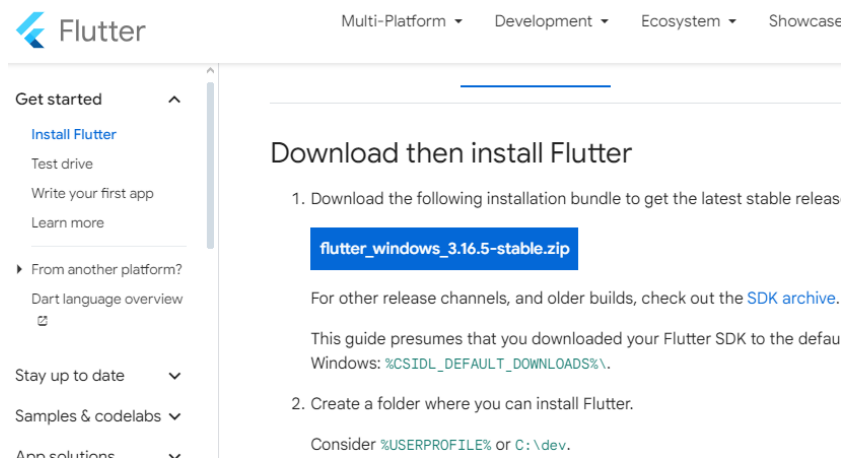


Figura 66. Descarga de Flutter desde el sitio web oficial

Con el archivo zip descargado, se descomprime y se copia en el disco C de la computadora, una vez realizado este procedimiento, se ingresa a Editar variables de entorno, dentro de esta propiedad del sistema seleccionamos la variable Path.

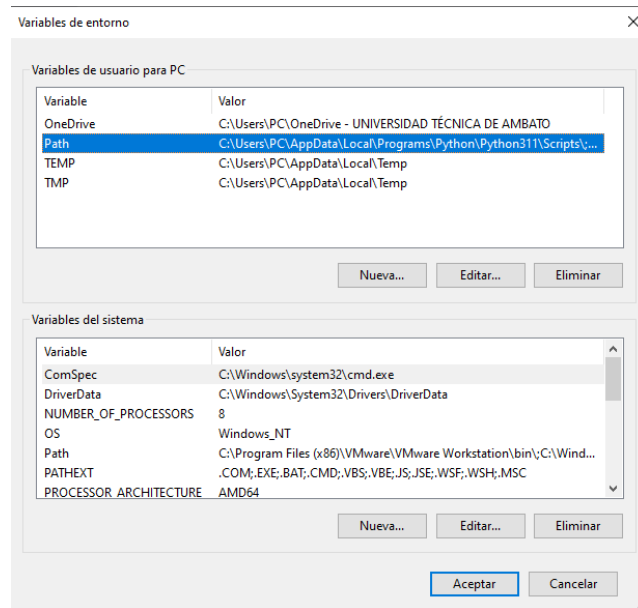


Figura 67. Variables de entorno para la instalación de Flutter

Se edita la variable Path, y se ingresa la ruta del archivo descomprimido en el disco C, dentro de la carpeta flutter y la subcarpeta bin, es decir la siguiente ruta: *C:\flutter\bin*.

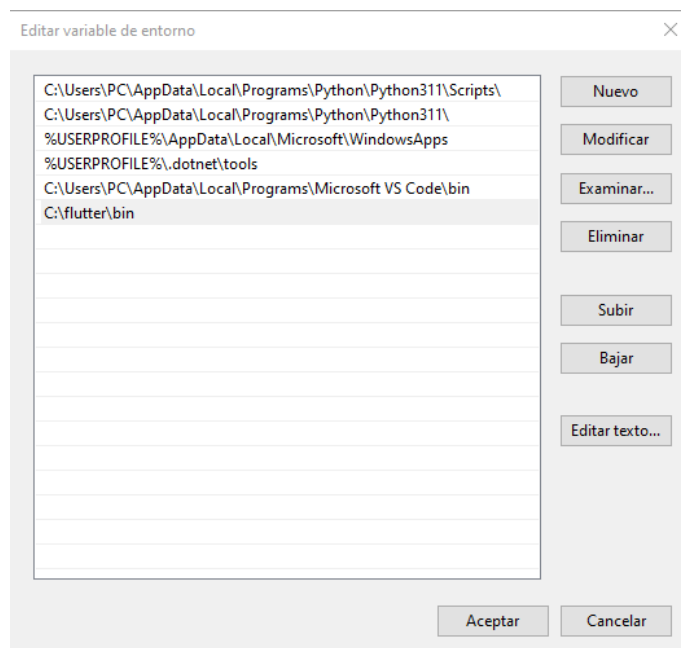
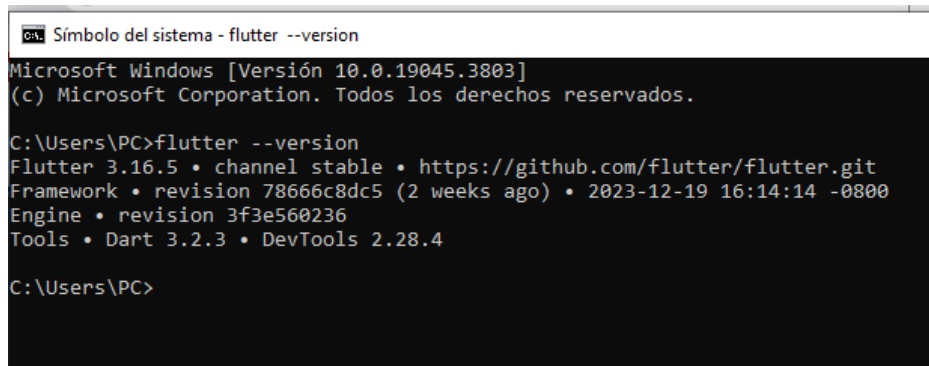


Figura 68. Edición de la variable path para la instalación de Flutter

Con el comando `flutter --version`, dentro del cmd, se verifica la instalación de Flutter.



```
Símbolo del sistema - flutter --version
Microsoft Windows [Versión 10.0.19045.3803]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\PC>flutter --version
Flutter 3.16.5 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 78666c8dc5 (2 weeks ago) • 2023-12-19 16:14:14 -0800
Engine • revision 3f3e560236
Tools • Dart 3.2.3 • DevTools 2.28.4

C:\Users\PC>
```

Figura 69. Verificación de la instalación de Flutter.

Para usar Flutter dentro de Visual Studio Code, se debe instalar su extensión junto con la extensión de Dart.

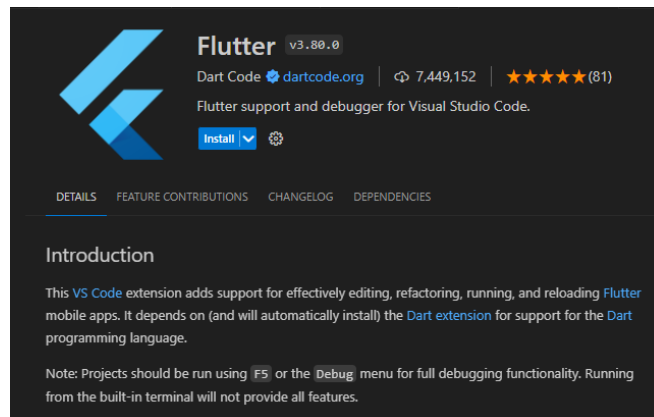


Figura 70. Instalación de la extensión de Flutter

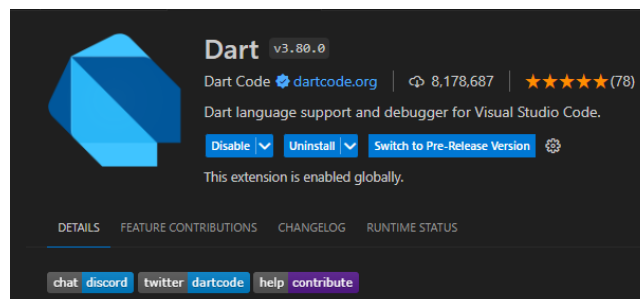


Figura 71. Instalación de la extensión de Dart.

b. Consideraciones para el diseño de la aplicación móvil

Para el desarrollo de las aplicaciones móviles se tomó en cuenta los estándares para el diseño visual e informacional de aplicaciones móviles que se analizó en la sección

1.3.9, donde se establece los aspectos importantes para el diseño. Cumpliendo con la Norma ISO 9241-11. En base a esta información se cumplió con los estándares y el resultado se visualiza en la Tabla 42.

Tabla 42. Cumplimiento de los estándares de diseño para app

Estándar	Cumplimiento	
	SI	NO
Soportar las distintas resoluciones de los dispositivos	X	
El instalador de la aplicación no supera los 10MB.	X	
Genera y provee el manual de uso para el usuario.	X	
La paleta de colores está relacionados al objetivo de la aplicación.	X	
La animación no debe verse excesiva, para que no haya distracciones por parte del usuario.	X	

Para que la aplicación soporte las distintas resoluciones de los dispositivos, se usó el widget ‘LayoutBuilder’, de Flutter, para adaptar el diseño de la app móvil, a cualquier resolución de pantalla del dispositivo final.

El instalador de la aplicación con extensión .apk tiene un tamaño de 9MB.

El manual de usuario de la aplicación se puede observar en el Anexo J.

El diseño de colores de la aplicación se estableció en base a la paleta de colores para dispositivos médicos (Figura 72) ya que esta aplicación está relacionada con la salud mental.



Figura 72. Paleta de colores para aplicación de salud

De esta paleta de colores se eligió, el blanco (#FDFEFE), el verde (#58D68D) y el rojo (#E74C3C).

c. Desarrollo de la ventana de Inicio

Para el desarrollo de la ventana Home, se usó la clase *'HomeScreen'* que representa la pantalla de inicio de la aplicación mediante un widget sin estado, el cuerpo de la pantalla se representa con la clase *'HomeScreenView'*, la que usa *'Padding'* para el texto de la pantalla y la clase *'CustomListTile'* que permite definir los títulos y los textos de la pantalla. El resultado de la pantalla de home se observa en la Figura 73.

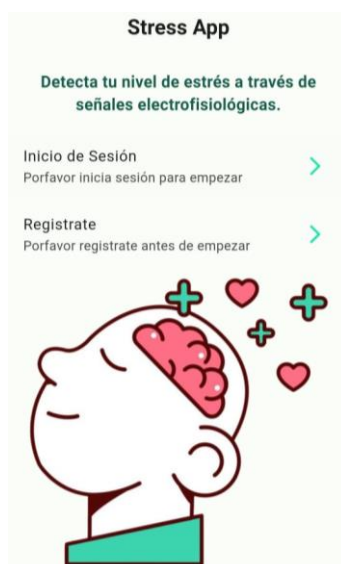


Figura 73. Interfaz gráfica de inicio

d. Desarrollo de la ventana de registro

Para realizar la ventana registro que será la interacción con el usuario, se debe importar las bibliotecas y paquetes necesarios, incluyendo aquellos relacionados con Flutter, Riverpod (para gestión de estado), widgets personalizados, constantes, enrutamiento y comunicación HTTP. Se define una clase llamada *RegisterScreen* que extiende *ConsumerWidget*. Se crean controladores de texto para capturar la entrada del usuario y algunas variables para gestionar la fecha de nacimiento y la aceptación de términos y condiciones.

```

class RegisterScreen extends ConsumerWidget {
  TextEditingController controllerName = TextEditingController();
  TextEditingController controllerLastname = TextEditingController();
  TextEditingController controllerUsername = TextEditingController();
  TextEditingController controllerPassword = TextEditingController();
  String? fechaNacimiento = Constants.age;
  TextEditingController controllerEmail = TextEditingController();
  bool isChecked = true;
  RegisterScreen({super.key});
}

```

Figura 74. Controladores de texto

Se define una función asíncrona llamada *register*, que se encarga de realizar el registro del usuario. La función realiza validaciones básicas y envía una solicitud HTTP POST al servidor.

```

var url = Uri.parse('https://shirleytesisbd.000webhostapp.com/register.php');
var uuid = const Uuid();
String nuevoUuid = uuid.v4();
var response = await http.post(url, body:{
  "id_usuario" : nuevoUuid,
  "username" : controllerName.text,
  "userlastname" : controllerLastname.text,
  "nickname" : controllerUsername.text,
  "password" : controllerPassword.text,
  "userdate" : fechaNacimiento,
  "useremail" : controllerEmail.text,
});

```

Figura 75. Conexión con el servidor

Se utilizan widgets personalizados como *customTextField* y *customDateInput* para facilitar la construcción de la interfaz de usuario. Y se usa la función *ref.read(appRouterProvider).go('/')* para navegar en la pantalla con el botón “regresar”. Los mensajes emergentes informan al usuario sobre el resultado en el proceso de registro y se crean con la función ‘*Fluttertoast.showToast*’.

Figura 76. Interfaz gráfica de la ventana registro

e. Desarrollo de la ventana de login

Para la creación de la ventana de login, se declara la función *'Future login() async'* que se encarga de manejar la lógica de inicio de sesión. Se encarga de realizar validación y envía una solicitud POST al servidor para autenticar al usuario y gestionar la respuesta del servidor.

Se usa el método *'build'* que define la interfaz de usuario de la pantalla de inicio de sesión. Aquí se encuentran los campos de entrada para el nombre de usuario y la contraseña, los botones, los enlaces y otros elementos visuales. La lógica del botón de inicio de sesión invoca la función *'login'* cuando se presiona. Esto inicia el proceso de autenticación del usuario. Después de enviar la solicitud al servidor, la aplicación maneja la respuesta. Si el servidor devuelve "Success", la aplicación navega a la pantalla del dashboard. Si hay un error, se muestra un mensaje de error.

```
var url = Uri.parse('https://shirleytesisbd.000webhostapp.com/login.php');
var response = await http.post(url, body: {
  "nickname" : controllerUsername.text,
  "password" : controllerPassword.text,
});
var data = json.decode(response.body);
if (data == "Success"){
  ref.read(appRouterProvider).go('/dashboard');
} else{
  Fluttertoast.showToast(
    msg: "El usuario o la contraseña esta incorrecta",
    toastLength: Toast.LENGTH_SHORT,
    gravity: ToastGravity.CENTER,
    fontSize: 16.0,
```

Figura 77. Validación de usuario

Se utiliza Riverpod para gestionar el estado y la navegación entre pantallas. La función *ref.read(appRouterProvider).go('/')* navega a la pantalla principal cuando se presiona el botón de regreso. Se utilizan widgets personalizados como *customTextField* para facilitar la construcción de la interfaz de usuario.

Figura 78. Interfaz gráfica de la ventana de login

f. Desarrollo de la ventana de dashboard

Para la creación de la ventana de dashboard se usó la estructura básica del widget *Scaffold* que contiene una barra de aplicación y un cuerpo con varios elementos, incluidos gráficos en tiempo real y botones.

```
return Scaffold(  
  appBar: AppBar(  
    title: const Text('Regresar'),  
    leading: IconButton(  
      icon: const Icon(Icons.arrow_back),  
      onPressed: () {  
        // Lógica para regresar  
        ref.read(appRouterProvider).go('/');  
      },  
    ),  
  ),  
);
```

Figura 79. Creación de los dashboard

El widget *RealTimeChart* es un *StatefulWidget* que se encarga de mostrar un gráfico en tiempo real con datos actualizados desde el servidor. Las funciones asíncronas que obtienen datos desde el servidor PHP para los sensores GSR y EMG y actualizan el gráfico en tiempo real.

```
class _RealTimeChartState extends State<RealTimeChart> {  
  late List<ChartData> chartDataGSR;  
  late List<ChartData> chartDataEMG;  
  late Timer timer;  
  double minValueGSR = 0.0;  
  double maxValueGSR = 0.0;  
  double averageValueGSR = 0.0;  
  double minValueEMG = 0.0;  
  double maxValueEMG = 0.0;  
  double averageValueEMG = 0.0;  
}
```

Figura 80. Actualización de datos en tiempo real

Se utiliza la biblioteca *syncfusion_flutter_charts* para crear un gráfico de área con datos de las listas *chartDataGSR* y *chartDataEMG*. Estos datos se actualizan constantemente desde el servidor.

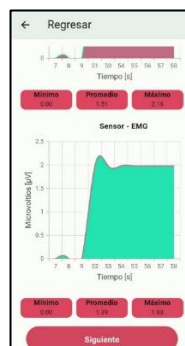


Figura 81. Interfaz gráfica de la ventana de Dashboard

g. Desarrollo de la ventana de formulario

Para realizar la ventana del formulario, primero se declara varias variables que representan el estado del formulario, como `selectedEstadoP1`, `selectedEstadoP2`, ..., `selectedEstadoP8`. Estas variables almacenan las respuestas seleccionadas por el usuario para cada una de las preguntas del formulario.

La función `formulario ()` se encarga de enviar las respuestas del formulario al servidor utilizando una solicitud HTTP POST. Si la respuesta del servidor es "Success", la aplicación navega a la pantalla de resultados (`'/result'`). Si hay un error, se muestra un mensaje utilizando el paquete `fluttertoast`.

```
Future formulario() async{
  var url = Uri.parse('https://shirleytesisbd.000webhostapp.com/formulario.php');
  var response = await http.post(url, body:{
    "pregunta1" : selectedEstadoP1,
    "pregunta2" : selectedEstadoP2,
    "pregunta3" : selectedEstadoP3,
    "pregunta4" : selectedEstadoP4,
    "pregunta5" : selectedEstadoP5,
    "pregunta6" : selectedEstadoP6,
    "pregunta7" : selectedEstadoP7,
    "pregunta8" : selectedEstadoP8,
  });
  var data = json.decode(response.body);
  if (data == "Success"){
    ref.read(appRouterProvider).go('/result');
  } else{
    Fluttertoast.showToast(
      msg: "Error al registrar sus respuestas",
      toastLength: Toast.LENGTH_SHORT,
      gravity: ToastGravity.CENTER,
      fontSize: 16.0,
    );
  }
}
```

Figura 82. Creación de la ventana de dashboard

El código utiliza widgets de Flutter para construir la interfaz de usuario del formulario. Se incluyen preguntas con menús desplegables (`customDropdown`), cada uno representado por un `Row` y un `SizedBox` con un botón para enviar las respuestas.

Dentro de cada pregunta, hay un menú desplegable (`customDropdown`) que utiliza el método `ref.read(formNotifierProvider.notifier).mapEventsToState()` para actualizar el estado utilizando el patrón `Notifier`. Cada respuesta seleccionada se refleja en el estado.

El botón "Obtener Resultados" al final del formulario utiliza el método `formulario ()` para enviar las respuestas al servidor. Antes de enviar, también se utiliza `ref.read(formNotifierProvider.notifier).mapEventsToState(const AddTodo())` para agregar los resultados del formulario al estado utilizando el patrón `Notifier`.

Figura 83. Interfaz gráfica de la ventana de formulario

h. Desarrollo de la ventana de resultado

Se utiliza `ref.watch(resultadoestresProvider)` para obtener el resultado del proveedor `resultadoestresProvider`, que contiene el resultado del estrés. El botón "Mostrar Resultados" tiene un `onPressed` que llama a `ref.read(resulttestresProvider.notifier).update((state) => state + 1)`. Esto sugiere que el resultado del estrés se actualiza aumentando en 1.

```
Widget build(BuildContext context, WidgetRef ref) {
  final resultadoAsync = ref.watch( resultadoestresProvider );

  return Scaffold(
    appBar: AppBar(
      title: const Text('Regresar'),
      leading: IconButton(
        icon: const Icon(Icons.arrow_back),
        onPressed: () {
          ref.read(appRouterProvider).go('/form');
        },
      ), // IconButton
    ), // AppBar
    body: Container(
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(20),
```

Figura 84. Creación de la ventana resultado

La sección que muestra el resultado utiliza `resultadoAsync.when(...)` para manejar diferentes estados del resultado (cargado, error, cargando). Se utiliza un `Container` para resaltar visualmente el resultado.

Se presentan tres botones que representan diferentes niveles de estrés: Bajo, Moderado y Alto. Cada botón está asociado a una función (`screenTratamientoBajo`,

screenTratamientoModerado, *screenTratamientoAlto*) que muestra un cuadro de diálogo con información sobre el tratamiento correspondiente al nivel de estrés.



Figura 85. Interfaz gráfica de la ventana de resultados

i. Desarrollo de la ventana de reporte

Para la creación de reportes se estableció una clase llamada *'ChartDataReport'*, que se encarga de almacenar los datos en un gráfico circular, y se usaron las funciones *'buildChart'* y *'buildReportChart'*, que permite construir el grafico circular con los datos del estrés bajo, moderado y alto y muestra los datos estadísticos de este. Además, se uso un widget llamado *'PaginatedDataTableWidget'* que muestra datos en una tabla paginada y con *'DataTable'* se muestran los detalles sobre las consultas de ID, EMG, GSR y el resultado del nivel de estrés.



Figura 86. Interfaz gráfica de la ventana de reporte

3.2.8 Registro y almacenamiento de datos

Para el registro y almacenamiento de datos, el servidor principal es alojado en 000webhost, donde se ha implementado una estructura de APIs en PHP para gestionar la interacción con una aplicación móvil construida en Flutter. La aplicación móvil actúa como el principal punto de acceso y envío de información hacia la base de datos MySQL. Este enfoque facilita una arquitectura distribuida y eficiente. En la Figura 87 se puede observar la arquitectura general que compone una API.

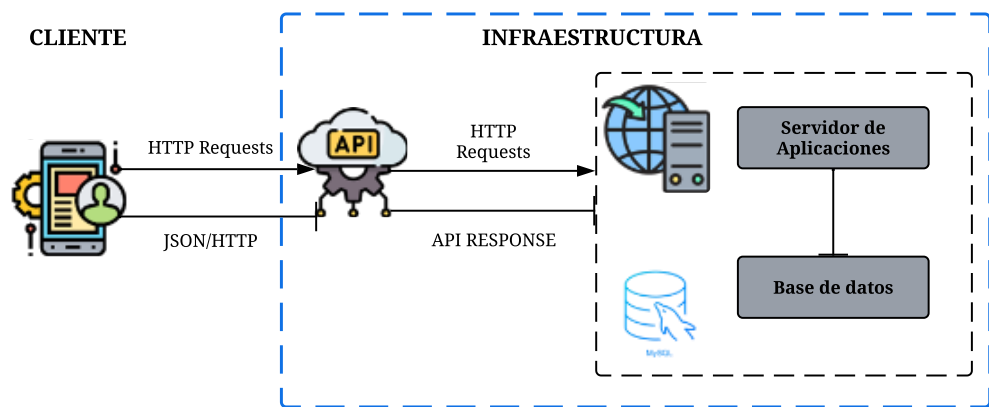


Figura 87. Arquitectura API

El sistema aborda varias funcionalidades clave mediante APIs dedicadas. Se ha implementado una API específica para la validación del inicio de sesión de usuario, proporcionando una capa de seguridad robusta. Además, se han creado APIs para el manejo del dashboard, el registro de usuarios, la captura de datos desde formularios, la generación de resultados como diagnóstico y una para guardar el resultado obtenido. Estas APIs no solo garantizan una comunicación eficiente, sino que también facilitan la integración con la aplicación móvil, su esquema general se observa en la Figura 88.

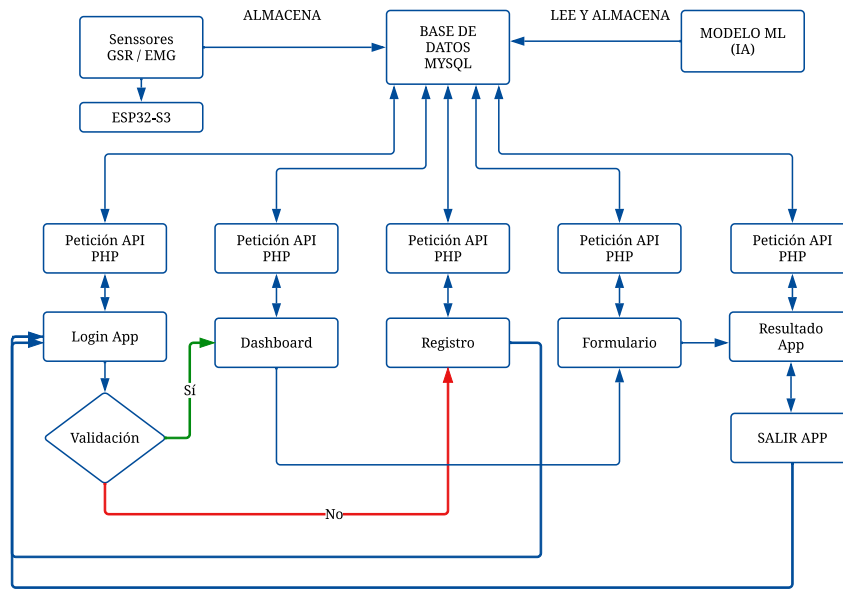


Figura 88. Esquema general del almacenamiento de datos

j. Almacenamiento de datos en MySQL

Los datos que se ingresan para realizar la detección del nivel de estrés se alojan en una base de datos, de acuerdo con la comparación realizada en la Tabla 30, se utiliza MySQL debido a que es de código abierto y puede ser administrada mediante PhpMyAdmin. De acuerdo con el esquema de la Figura 88, se genera una base de datos con cuatro tablas, llamadas “formulario”, “sensores”, “usuarios” y “resultado”. La relación de la base de datos se puede analizar en la Figura 89.

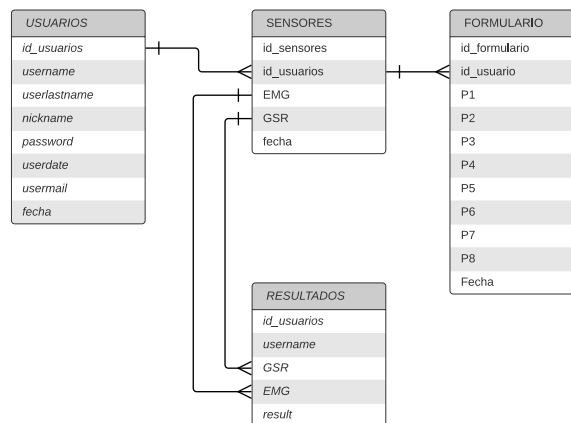


Figura 89. Diagrama de las relaciones de la base de datos

Cada tabla consta de los campos necesarios para alojar la información que se registra en la aplicación móvil.

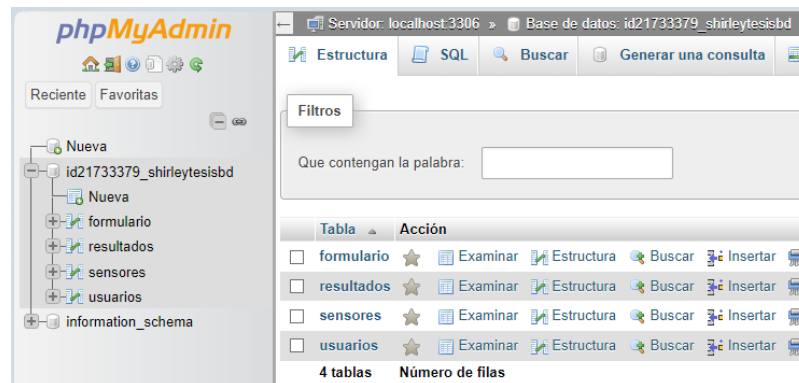


Figura 90. Base de datos para el almacenamiento de información

La tabla llamada “usuarios” es la principal, debido que en esta se almacena la información de registro con la que los estudiantes podrán ingresar a la aplicación, esta tabla consta de ocho campos, divididos en “id_usuario”, que se genera automáticamente y es muy importante ya que es la credencial del usuario, “username” y “userlastname” corresponden a los campos donde los usuarios deben colocar su nombre y apellido con los que desean registrarse, el “nickname” corresponde al nombre de usuario para rápido acceso, “password” a la contraseña, “userdate” a la fecha de nacimiento del usuario, “useremail” al correo electrónico y “fecha” corresponde al campo donde se almacena la fecha en la cual se realiza el registro de usuario.

The screenshot shows the 'Estructura de tabla' view for the 'usuarios' table. The table structure is as follows:

#	Nombre	Tipo	Cotejamiento	At
<input type="checkbox"/> 1	id	int(11)		
<input type="checkbox"/> 2	id_usuario	varchar(150)	utf8mb4_general_ci	
<input type="checkbox"/> 3	username	varchar(150)	utf8mb4_general_ci	
<input type="checkbox"/> 4	userlastname	varchar(150)	utf8mb4_general_ci	
<input type="checkbox"/> 5	nickname	varchar(100)	utf8mb4_general_ci	
<input type="checkbox"/> 6	password	varchar(150)	utf8mb4_general_ci	
<input type="checkbox"/> 7	userdate	varchar(100)	utf8mb4_general_ci	
<input type="checkbox"/> 8	useremail	varchar(200)	utf8mb4_general_ci	
<input type="checkbox"/> 9	fecha	timestamp(6)		

Figura 91. Tabla de registro de usuarios

La tabla llamada “formulario”, consta de diez campos que hacen referencia a la id_formulario y a las 8 preguntas planteadas en la Tabla 35, que corresponden al formulario SISCO aplicado en este proyecto, donde se registran las respuestas definidas por los usuarios mediante la aplicación móvil y la “fecha” de cuando se realizó el registro.

#	Nombre	Tipo	Cotejamiento	Atributos
<input type="checkbox"/>	1 id_formulario	int(11)		
<input type="checkbox"/>	2 id_usuario	int(11)		
<input type="checkbox"/>	3 p1	int(11)		
<input type="checkbox"/>	4 p2	int(11)		
<input type="checkbox"/>	5 p3	int(11)		
<input type="checkbox"/>	6 p4	int(11)		
<input type="checkbox"/>	7 p5	int(11)		
<input type="checkbox"/>	8 p6	int(11)		
<input type="checkbox"/>	9 p7	int(11)		
<input type="checkbox"/>	10 p8	int(11)		
<input type="checkbox"/>	11 fecha	int(11)		

Figura 92. Tabla del registro del formulario

Y la tabla de sensores consta de cinco campos donde se registran los datos obtenidos de acuerdo con cada variable fisiológica en este caso “EMG” y “GSR”, estos se relacionan con el campo “id_usuarios”. Consta de un campo llamado “id_sensores” y “fecha”, que se generan de forma automática.

#	Nombre	Tipo	Cotejamiento	Atributos
<input type="checkbox"/>	1 id_sensores	int(11)		
<input type="checkbox"/>	2 id_usuario	varchar(150)	utf8mb4_general_ci	
<input type="checkbox"/>	3 EMG	float		
<input type="checkbox"/>	4 GSR	float		
<input type="checkbox"/>	5 fecha	timestamp(6)		

Figura 93. Tabla de registro de los sensores.

En la tabla denominada “resultados”, se almacenan los datos obtenidos por los sensores en relación con el usuario en tiempo real, esta tabla contiene seis campos que hacen referencia a “id_result”, “id_usuario”, “username”, “emg”, “gsr” y “result”, esta tabla es la responsable de la generación de reportes del sistema.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
1	id_result	int(11)			No	Ninguna		AUTO_INCREMENT
2	id_usuario	varchar(150)	utf8_unicode_ci		No	Ninguna		
3	username	varchar(100)	utf8_unicode_ci		No	Ninguna		
4	emg	float			No	Ninguna		
5	gsr	float			No	Ninguna		
6	result	varchar(50)	utf8_unicode_ci		No	Ninguna		

Figura 94. Tabla de resultados del sistema

k. Desarrollo de las peticiones API

Para la conexión entre la aplicación móvil y la base de datos se han realizado varias peticiones API que corresponden a los datos que se almacenan en las tablas de la base de datos creada en MySQL.

- ***Petición API para el login de la aplicación***

Esta API verifica la autenticación de un usuario mediante un sistema de nombre de usuario (nickname) y contraseña. Primero se establece una conexión a la base de datos MySQL utilizando `mysqli_connect`. Se proporcionan los detalles del servidor (localhost), el nombre de usuario, la contraseña y el nombre de la base de datos, se verifica si la conexión a la base de datos fue exitosa y se imprime el mensaje "Conexión exitosa".

```
$db = mysqli_connect('localhost', 'id21733379_shirley', '12345678.Qa/',
'id21733379_shirleytesisbd');
if (!$db) {
    echo 'Conexión exitosa';
}
```

Se recuperan el nombre de usuario (nickname) y la contraseña del formulario con una solicitud POST y se crea una consulta SQL para seleccionar el hash almacenado en la base de datos correspondiente al nombre de usuario proporcionado.

```

$nickname = $_POST['nickname'];
$password = $_POST['password'];

// Obtener el hash almacenado en la base de datos para el usuario con el
nickname proporcionado
$sql = "SELECT password FROM usuarios WHERE nickname = '$nickname'";
$result = mysqli_query($db, $sql);

```

Si la consulta SQL se ejecuta correctamente, se obtiene el hash almacenado desde la base de datos. Luego, se utiliza la función `password_verify` para comparar la contraseña proporcionada con el hash almacenado. Si coinciden, se imprime "Success"; de lo contrario, se imprime "Error". Si hay un error en la consulta SQL, se imprime "Error en la consulta SQL" y se cierra la conexión a la base de datos después de realizar la autenticación.

```

if ($result) {
    $row = mysqli_fetch_assoc($result);
    $hash_almacenado = $row['password'];

    // Verificar si la contraseña ingresada coincide con el hash
    almacenado utilizando password_verify
    if (password_verify($password, $hash_almacenado)) {
        // Contraseña válida
        echo json_encode("Success");
    } else {
        // Contraseña incorrecta
        echo json_encode("Error");
    }
} else {
    // Error en la consulta SQL
    echo json_encode("Error en la consulta SQL");
}

mysqli_close($db);
?>

```

- ***Petición API para los datos del registro***

Para el registro de un nuevo usuario, se establece una conexión a la base de datos MySQL utilizando `mysqli_connect`. Se proporcionan los detalles del servidor (localhost), el nombre de usuario, la contraseña y el nombre de la base de datos, y se verifica si la conexión a la base de datos fue exitosa.

```

?php
$db = mysqli_connect('localhost', 'id21733379_shirley', '12345678.Qa/',
'id21733379_shirleytesisbd');

```

```
if (!$db) {
    echo 'Conexión exitosa';
```

Se recuperan los datos mediante una solicitud POST, estos datos representan la información del nuevo usuario que se desea registrar.

```
}
$id_usuario = $_POST['id_usuario'];
$username = $_POST['username'];
$userlastname = $_POST['userlastname'];
$nickname = $_POST['nickname'];
$password = $_POST['password'];
$userdate = $_POST['userdate'];
$useremail = $_POST['useremail'];
```

La contraseña proporcionada se cifra utilizando la función 'password_hash'. Esta función permite almacenar de forma segura las contraseñas en la base de datos. Y se crea una consulta SQL para insertar los datos del nuevo usuario en la tabla 'usuarios'. Finalmente se ejecuta la consulta utilizando 'mysqli_query'. Dependiendo del resultado de la ejecución, se imprime "Success" o "Error" en formato JSON, y se cierra la conexión a la base de datos después de realizar la inserción del nuevo usuario.

```
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);

$sql = "INSERT INTO usuarios (id_usuario, username, userlastname,
nickname, password, userdate, useremail)
VALUES ('$id_usuario', '$username', '$userlastname', '$nickname',
'$hashedPassword', '$userdate', '$useremail')";

$result = mysqli_query($db,$sql);

if($result){
    echo json_encode("Success");
}
else{
    echo json_encode("Error");
}
mysqli_close($db);
?>
```

- **Petición API para los datos del formulario**

Primero se establece la conexión con la base de datos MySQL. Después se recuperan los datos del formulario con una solicitud POST y se crea una consulta SQL para insertar las respuestas del formulario en la tabla 'formulario'. Se ejecuta la consulta utilizando 'mysqli_query'. Dependiendo del resultado de la ejecución, se imprime

"Success" o "Error" en formato JSON. Finalmente se cierra la conexión a la base de datos después de realizar la inserción de datos del formulario.

```
<?php
$db = mysqli_connect('localhost', 'id21733379_shirley', '12345678.Qa/',
'id21733379_shirleytesisbd');

if (!$db) {
    echo 'Conexión exitosa';
}

$pregunta1 = $_POST['pregunta1'];
$pregunta2 = $_POST['pregunta2'];
$pregunta3 = $_POST['pregunta3'];
$pregunta4 = $_POST['pregunta4'];
$pregunta5 = $_POST['pregunta5'];
$pregunta6 = $_POST['pregunta6'];
$pregunta7 = $_POST['pregunta7'];
$pregunta8 = $_POST['pregunta8'];

$sql = "INSERT INTO formulario (p1, p2, p3, p4, p5, p6, p7, p8)
VALUES ('$pregunta1', '$pregunta2', '$pregunta3', '$pregunta4',
'$pregunta5', '$pregunta6', '$pregunta7', '$pregunta8')";

$result = mysqli_query($db,$sql);

if($result){
    echo json_encode("Success");
}
else{
    echo json_encode("Error");
}
mysqli_close($db);
?>
```

- ***Petición API para los datos de los sensores***

Esta API se encarga de recibir los datos del microcontrolador y los inserta en la base de datos MySQL, para ello se definen las credenciales y la ubicación de la base de datos MySQL a la que se va a conectar y se crea una nueva instancia de la clase 'mysqli' para establecer una conexión con la base de datos. Y se verifica si la conexión a la base de datos fue exitosa.

```
<?php
    $host = "localhost";
    $dbname = "id21733379_shirleytesisbd";
    $username = "id21733379_shirley";
    $password = "12345678.Qa/";

$conn = new mysqli($host, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
```

```

}
else { echo "Connected to mysql database. "; }

```

Se verifica si los datos recibidos desde el microcontrolador ('emg' y 'gsr') no están vacíos. Si ambos valores existen, se construye una consulta SQL para insertar estos valores en una tabla llamada sensores junto con un identificador de usuario (id_usuario). Si la inserción es exitosa, se imprime un mensaje indicando que los valores se han insertado en la tabla; de lo contrario, se muestra un mensaje de error.

```

if(!empty($_POST['emg']) && !empty($_POST['gsr']) )
{
    $val1 = $_POST['emg'];
    $val2 = $_POST['gsr'];
    $id2 = "3204cb8e-80e9-4c13-9a52-7ad54ddea139";
    $sql = "INSERT INTO sensores (id_usuario, EMG, GSR) VALUES
('".$id2."', '".$val1."', '".$val2."')";

    if ($conn->query($sql) === TRUE) {
        echo "Values inserted in MySQL database table.";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}

// Close MySQL connection
$conn->close();

?>

```

1. Almacenamiento en la nube

Para el desarrollo de este proyecto su uso el proveedor de servicios de hosting llamado 000Webhost, que es una plataforma accesible y gratuita que ofrece servicios como el espacio de almacenamiento, ancho de banda y opciones de gestión de archivos

Para beneficiarse de los servicios de 000WebHost hay que registrarse con un correo personal, el alojamiento gratuito permite un sitio web, 300MB de memoria SSD y 3GB de banda ancha sin límite de tiempo, estas características son suficientes en base a los requerimientos del sistema.

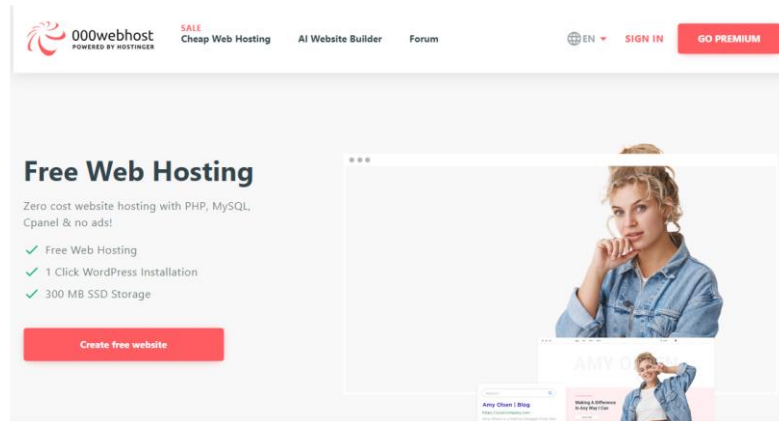


Figura 95. Registro en 000WebHost

m. Creación del servidor

La creación del servidor en 000webhost es sumamente sencilla, simplemente hay que presionar en crear un nuevo sitio web y llenarla con los datos que nos solicita y así podremos acceder a los servicios que dispone dentro del plan gratuito, una vez creado el sitio de alojamiento se puede hacer uso de los siguientes servicios, alojamiento para: sitios web, archivos, dominios, emails, base de datos, protección de contraseñas.

The image shows a 'Create website' form with a close button (X) in the top right corner. The form has two input fields for 'Website name (auto-generated if em...)' containing 'shirleytesisbd' and '.000webhostapp.com'. Below is a 'Password' field with a masked password '.....' and a visibility toggle. A list of requirements is shown with green checkmarks: 'One number', 'One symbol', 'One lowercase letter', 'One uppercase letter', 'Use 8-50 characters', and 'Only Latin letters'. Below that is a 'Repeat password' field with a masked password '.....' and a visibility toggle. A final checkmark indicates 'Passwords match'. At the bottom center is a red 'Create' button.

Figura 96. Creación del sitio de alojamiento web

n. Sincronización entre la base de datos y la nube

La base de datos MySQL se ha alojado en el servidor 000webhost, y para su sincronización se han realizado varias APIs como se detalla en el esquema de la Figura 88, las cuales permiten la conexión entre la base de datos y la aplicación móvil.

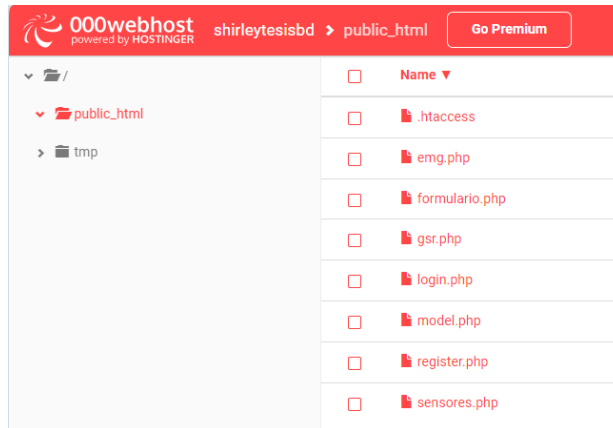


Figura 97. Sincronización del servidor con la base de datos

3.2.9 Pruebas de funcionamiento

Se llevaron a cabo pruebas de funcionamiento del sistema propuesto con el propósito de evaluar su confiabilidad. Estas pruebas se realizaron a un total de 10 estudiantes seleccionados aleatoriamente. Antes de realizar las pruebas se obtuvo su autorización y se les informó que el dispositivo no afectaría a su salud mental o les generaría incomodidad.

a. *Proceso para realizar las pruebas*

- Este proceso se puede observar en el manual de usuario en el Anexo J.
- Se debe colocar el prototipo en la muñeca de los usuarios con ayuda de los tirantes de velcro.
- Se enciende el dispositivo mediante el interruptor y se establece comunicación con la aplicación móvil.
- La toma de medidas requiere un período de 5 minutos para su finalización, durante los cuales se recopilan los datos. Los dos primeros minutos se asignan para permitir que el paciente se calme y para que los sensores se ajusten mediante software, dado que este tipo de análisis demanda que el paciente se encuentre en un estado lo más relajado posible. Los tres minutos siguientes constituyen el tiempo efectivo de medición, a partir del cual se debe realizar el análisis de los datos mediante la aplicación.

- Después de capturar las señales electrofisiológicas, se debe llenar el formulario para conocer su estado de estrés psicológicamente en la actualidad y se procede a visualizar los resultados dentro de la aplicación móvil

Estas pruebas se realizaron en base a los estresores académicos más comunes dentro de la comunidad universitaria como se detalló en la sección 1.3.2, la prueba se realizó a estudiantes bajo el factor estresante de procesos de evaluación, la primera etapa corresponde a la realización de la prueba a estudiantes que no se encuentran en época de evaluación y la segunda prueba se realiza a estudiantes que se encuentran en época de evaluaciones.

b. Prueba fuera de la época de evaluaciones

Esta prueba se realizó con estudiantes de semestres inferiores, los cuales no tenían que rendir una evaluación, pero si tenían trabajos que realizar, a estos sujetos de prueba se les conecto los sensores de manera indicada como se observa en la Figura 98 y usaron la aplicación para responder al formulario como se muestra en la Figura 99.



Figura 98. Sujeto 1 colocación de sensores



Figura 99. Prueba de funcionamiento



Figura 100. Lectura de GSR y EMG del sujeto 1

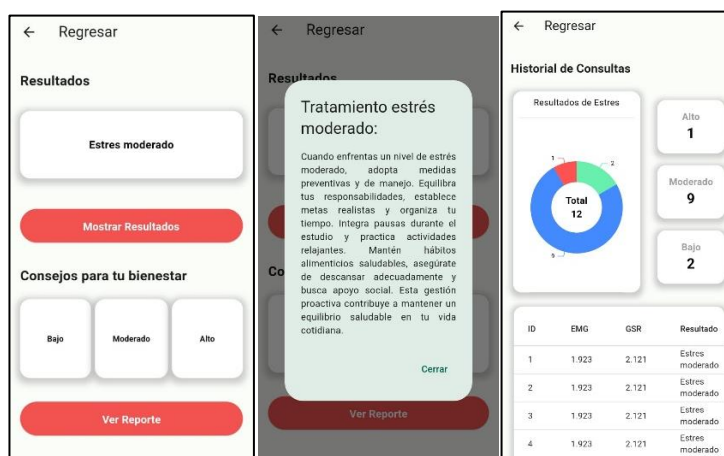


Figura 101. Resultados de la aplicación móvil

Los resultados se obtuvieron de la ventana de reportes de la aplicación donde se realiza un historial de la detección de nivel de estrés de los usuarios bajo diferentes estresores y diferentes tiempos.

Tabla 43. Prueba fuera de la época de evaluaciones

Usuario	GSR	EMG	P1	P2	P3	P4	P5	P6	P7	P8	NIVEL
1	1.92	2.12	2	2	3	3	3	2	2	3	Estrés moderado
2	0.48	0.6	2	1	1	1	2	1	1	1	Estrés bajo
3	1.10	2.29	3	2	2	1	3	4	1	2	Estrés moderado
4	1.80	2.05	2	2	3	3	2	2	3	2	Estrés moderado
5	0.55	0.68	2	1	1	1	2	1	1	1	Estrés bajo
6	1.05	2.20	3	2	2	1	3	4	1	2	Estrés moderado
7	1.60	1.85	2	3	3	2	2	3	3	2	Estrés moderado
8	0.72	0.90	1	1	2	1	1	1	1	2	Estrés bajo
9	1.20	2.05	2	2	1	3	4	1	2	2	Estrés moderado
10	2.10	2.40	3	3	3	2	2	3	3	2	Estrés moderado

- **Análisis de resultados**

Los resultados obtenidos se encuentran dentro de los rangos establecidos en la Tabla 33, y dentro de los rangos validados por el Psicólogo clínico (Anexo I), en este entorno de prueba se obtuvo siete usuarios que muestran un nivel de estrés moderado y tres usuarios con un nivel bajo de estrés. Los resultados reflejan la capacidad del método de detección y ofrecen información valiosa sobre la distribución de los niveles de estrés en el grupo evaluado.

c. Pruebas en épocas de evaluaciones

Esta prueba fue aplicada a estudiantes durante la semana de evaluaciones y entrega de proyectos finales, para estimar su nivel de estrés se realizó el procedimiento anterior, donde ellos interactuaron con la aplicación y se obtuvo los resultados descritos en la Tabla 44.

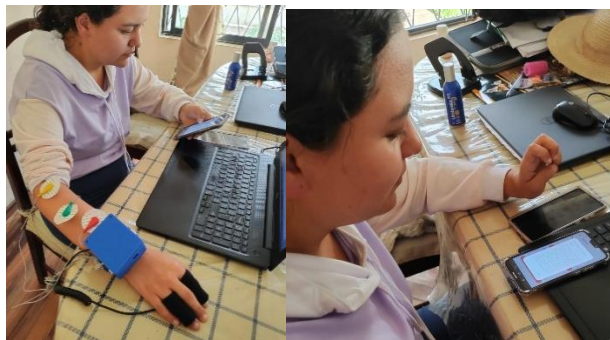


Figura 102. Prueba de funcionamiento del sujeto 2

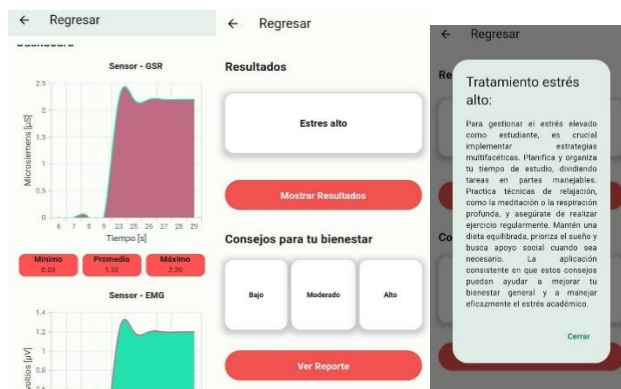


Figura 103. Resultados bajo estresores académicos

En base a los resultados obtenidos bajo estresores académicos se obtuvieron los datos detallados en la Tabla 44, donde se realizó las pruebas a 10 sujetos, y se analizó el reporte dado por el sistema.

Tabla 44. Prueba bajo estresores académicos

Usuario	GSR	EMG	P1	P2	P3	P4	P5	P6	P7	P8	NIVEL
1	3.97	2.8	4	2	3	3	4	3	4	3	Estrés alto
2	4.15	3.1	5	3	4	4	3	3	2	2	Estrés alto
3	2.80	2.5	2	1	3	3	2	2	3	3	Estrés moderado
4	1.80	0.95	2	3	3	3	2	4	3	4	Estrés moderado
5	4.05	3.1	4	4	3	5	4	3	4	5	Estrés alto
6	3.9	2.35	3	3	4	4	4	4	2	3	Estrés alto
7	3.27	3.1	4	4	3	3	2	3	3	4	Estrés moderado
8	2.37	2.28	2	3	2	2	3	2	3	3	Estrés moderado
9	4.38	3.2	3	1	4	4	4	4	2	2	Estrés alto
10	3.92	2.40	4	2	3	5	4	3	2	5	Estrés moderado

- **Análisis de resultados**

Los datos obtenidos por los sensores se encuentran dentro del rango establecido en la Tabla 33, con relación a las variables GSR y EMG, en este entorno de prueba se obtuvo como resultado cinco niveles denominados estrés alto y cinco con estrés moderado debido a las exigencias académicas que conllevan los estudiantes en épocas de evaluación.

d. Errores y confiabilidad de los sensores del sistema

Para obtener el error entre los valores obtenidos de cada sensor usado en el prototipo, es decir el Muscle sensor V3 y el grove sensor GSR, es necesario tener presente los valores teóricos de cada variable fisiológica, debido a que en el mercado ecuatoriano a un no hay en existencia un dispositivo que mida la actividad electrodérmica y la actividad muscular.

El cálculo del error absoluto para cada prueba se realiza a través de:

$$Error\ absoluto = valor\ teorico - valor\ medido$$

El cálculo del error relativo (ER) se calcula con la siguiente formula:

$$ER = \left| \frac{Valor\ teórico - valor\ medido}{valor\ teórico} \right| \times 100$$

El valor teórico del sensor Grove GSR y del sensor EMG, se obtuvo de la base de datos usada para el entrenamiento del modelo de ML, que se encuentra bajo las mismas circunstancias en las que se realizaron las mediciones.

Tabla 45. Cálculo de error del sensor GSR

Nº medición	GSR teórico [uS]	GSR medido [uS]	Error absoluto	Error relativo	Error relativo porcentual
1	4,25	3,97	0,28	0,07	6,59
2	4,35	4,15	0,20	0,05	4,60
3	3,10	2,80	0,30	0,10	9,68
4	1,97	1,80	0,17	0,09	8,63
5	4,16	4,05	0,11	0,03	2,64
6	4,05	3,90	0,15	0,04	3,70
7	3,56	3,27	0,29	0,08	8,15
8	2,51	2,37	0,14	0,06	5,58
9	4,42	4,38	0,04	0,01	0,90
10	4,20	3,92	0,28	0,07	6,67
PROMEDIO					5,71

Tabla 46. Cálculo de error del sensor EMG

Nº medición	EMG teórico [mV]	EMG medido [mV]	Error absoluto	Error relativo	Error relativo porcentual
1	3,10	2,8	0,30	0,10	9,68
2	3,25	3,1	0,15	0,05	4,62
3	2,84	2,5	0,34	0,12	11,97
4	1,05	0,95	0,10	0,10	9,52
5	3,24	3,1	0,14	0,04	4,32
6	2,48	2,35	0,13	0,05	5,24
7	3,19	3,1	0,09	0,03	2,82
8	2,41	2,28	0,13	0,05	5,39
9	3,48	3,2	0,28	0,08	8,05
10	2,60	2,4	0,20	0,08	7,69
PROMEDIO					6,93

La confiabilidad del sistema se obtiene con el cálculo del promedio ponderado de los errores porcentuales individuales.

$$\text{Confiabilidad del sistema} = 100\% - \text{error promedio porcentual del sistema}$$

$$\text{Confiabilidad del sistema} = 100\% - 6.32\% = 93.68\%$$

e. Prueba de confianza y consistencia de datos

Para determinar la confiabilidad en los datos de los sensores se realizó tres mediciones en un determinado intervalo de tiempo, esto se realizó con tres usuarios donde se calculó la varianza y la desviación estándar de cada sensor.

- *Sensor GSR*

Tabla 47. Resultado prueba de consistencia de datos GSR

USUARIO	PRUEBA 1	PRUEBA 2	PRUEBA 3	PROMEDIO	DESVIACIÓN ESTÁNDAR	VARIANZA
1	2,18	2,16	2,14	2,16	0,0200	0,0004
2	1,59	1,61	1,62	1,61	0,0153	0,0002
3	3,9	4,1	4,05	4,02	0,1041	0,0108

- *Sensor EMG*

Tabla 48. Resultado prueba de consistencia de datos EMG

USUARIO	PRUEBA 1	PRUEBA 2	PRUEBA 3	PROMEDIO	DESVIACIÓN ESTÁNDAR	VARIANZA
1	1,31	1,24	1,27	1,27	0,0351	0,0012
2	1,99	2,03	1,95	1,99	0,0400	0,0016
3	2,16	2,11	2,08	2,12	0,0404	0,0016

- *Análisis de resultados*

Los valores del conjunto de datos son bastante consistentes y cercanos entre sí. La variabilidad en torno a la media es mínima, ya que tanto la desviación estándar como la varianza son muy bajas. En conclusión, se tiene un conjunto de datos bastante homogéneo y estable, al menos en los tres valores proporcionados. Estas medidas estadísticas indican que los datos son bastante coherentes y no hay una dispersión significativa. Con esto se determina la eficiencia de la aplicación para su uso en estudiantes universitarios o en profesionales encargados de la salud mental para monitorear el nivel de estrés en cada sujeto, obtener recomendaciones necesarias y brindar un diagnóstico preciso.

3.2.10 Autonomía del sistema

Calcular la autonomía del sistema requiere evaluar el consumo de potencia del prototipo, donde se especifica la corriente y el voltaje de operación de cada componente del sistema de acuerdo con la información proporcionada en los respectivos Datasheet del fabricante.

En la Tabla 49, se analiza la potencia de la placa con el sensor GSR, mientras que en la Tabla 50 se analiza la potencia del sensor EMG, debido a que este sensor necesita dos baterías de 9V independientemente de la batería que alimenta la placa.

Tabla 49. Potencia de la placa

Elemento	Consummo de corriente	Cantidad	Consumo [mA]	Voltaje de operacion [V]	Potencia [W]
ESP32	100	1	100	3	0,33
Sensor GSR	0,72	1	20	3,3	0,066
Total					0.396

Tabla 50. Potencia del sensor EMG

Elemento	Consummo de corriente	Cantidad	Consumo [mA]	Voltaje de operacion [V]	Potencia [W]
Sensor EMG	15	2	30	9	0,54
Total					0.54

Una vez que se ha establecido el consumo de potencia promedio del prototipo, se procede a calcular la energía suministrada por la batería. Este cálculo se basa en el voltaje nominal y la capacidad de amperaje de las baterías, que se puede observar en la Tabla 51.

Tabla 51. Potencia de las baterías

Bateria	Corriente [mAh]	Voltaje [V]
Nano Tech Lipo	600	3.7
Li - poly	650	9

$$Energia_{bateria1} = V.I = 3.7V . 0.6Ah$$

$$Energia_{bateria1} = 2.22 [Wh]$$

$$Energia_{bateria2} = V.I = 9V . 0.65Ah = 5.85 [Wh]$$

Al aplicar la ecuación de autonomía del sistema, se calcula la duración autónoma del sistema considerando el consumo medio del circuito y la energía provista por la batería. Además, se establece la autonomía del sensor EMG, debido a que necesita energía de baterías adicionales.

$$Autonomia_{sistema} = \frac{Energia\ bateria\ total}{Potencia\ placa} \quad (4)$$

$$Autonomia_{placa} = \frac{2.22Wh}{0.936W} = 2.37 [h]$$

$$Autonomia_{EMG} = \frac{5.85Wh}{0.54W} = 10.8 [h]$$

La autonomía del sistema se realizó, mediante el análisis de la autonomía de la placa con la autonomía del sensor EMG. Este análisis concluye que el sistema mantiene su operatividad mediante la alimentación independiente durante un período de 6.30 horas.

$$Autonomia_{sistema} = 6.30 [h]$$

3.2.11 Presupuesto

Para determinar el presupuesto de implementación del proyecto se consideró el costo del hardware y de construcción. Para determinar el presupuesto del hardware se consideró todos los gastos que incluyen materiales y construcción.

Tabla 52. Presupuesto de implementación del prototipo

ITEM	DETALLE	CANT	VALOR UNI	VALOR TOTAL
1	ESP32 Tiny S3	1	\$30.00	\$30.00
2	Sensor Grove GSR	1	\$32.00	\$32.00
3	Sensor Muscle V3	1	\$50.00	\$50.00
4	Módulo TP4056	1	\$2.00	\$2.00
5	Baterías recargables de 9V	2	\$15.00	\$30.00
6	Bateria recargable Lipo de 3.7V	1	\$12	\$12
7	Interruptor	1	\$2	\$2
8	Impresión PCB CNC	1	\$8	\$8
9	Impresión 3D caja contenedora	1	\$20	\$20
10	Velcro	1	\$2	\$2
11	Elementos electrónicos varios	1	\$10	\$10
			Subtotal	\$176,79
			IVA (12%)	\$21.21
			TOTAL	\$198.00

Al determinar el presupuesto de la construcción, es esencial considerar el tiempo invertido en la creación del prototipo. Este tiempo se calcula en función de la tarifa por hora, derivada del salario promedio de un Ingeniero en Telecomunicaciones, establecido por el Ministerio de Trabajo en \$858 dólares mensuales. Se basa en un horario laboral de 8 horas diarias y una duración de 21 días al mes, resultando en:

$$remuneracion_{diaria} = \frac{remuneración_{mensual}}{dias\ laborales}$$

$$remuneracion_{diaria} = \frac{858}{21} = \$40.86$$

$$remuneracion_{hora} = \frac{40.86}{8} = \$5.11$$

La estimación para el tiempo de diseño, desarrollo, corrección de errores y pruebas de funcionamiento del proyecto se calculó en un total de 140 horas, generando un presupuesto de mano de obra equivalente a 715.40 dólares.

$$remuneracion_{diaria} = Horas_{mano\ de\ obra} \times remuneracion_{hora}$$

$$remuneracion_{diaria} = 140 \times 5.11 = \$715.40$$

Con la siguiente ecuación se obtiene el presupuesto total.

$$presupuesto_{total} = presupuesto_{hardware} \times presupuesto_{construcción}$$

$$presupuesto_{total} = \$198.00 + \$715.40$$

$$presupuesto_{total} = \$198.00 + \$715.40 = \$913.40$$

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Con base en la información recopilada sobre las variables electrofisiológicas vinculadas al estrés en estudiantes universitarios, se identificó la actividad electrodérmica (EDA/GSR); la detección del estrés a través de esta variable implica monitorear las fluctuaciones en la conductividad eléctrica de la piel. Esta medición se expresa en Siemens [S], donde el rango adecuado para un nivel bajo de estrés es inferior a 0.7 [uS], el nivel moderado oscila entre 0.8 y 3.8 [uS], mientras que el nivel alto se manifiesta por valores superiores a 3.9 [uS]. Asimismo, la electromiografía (EMG), es la encargada de medir la actividad eléctrica de los músculos, revela que, en situaciones de estrés, el sistema nervioso simpático induce la activación de determinados músculos, reflejándose en un aumento de la actividad eléctrica. Esta variable se cuantifica en voltios [V], y el nivel bajo de estrés se caracteriza por un valor inferior a 0.5 [mV], el nivel moderado se sitúa entre 0.6 y 2.8 [mV], mientras que el nivel alto supera los 2.9 [mV].
- Se llevó a cabo un exhaustivo análisis bibliográfico de los algoritmos de Machine Learning, destacándose especialmente los algoritmos de aprendizaje supervisado por su eficacia para abordar la complejidad de los datos electrofisiológicos vinculados al estrés. Entre los modelos evaluados, el Random Forest demostró una mayor eficiencia, evidenciada por una "Precision" y "Recall" de 0.95 para la clase "Nivel alto". Asimismo, para la clase "Nivel moderado", se lograron valores de 0.93 y 0.96 respectivamente en "Precision" y "Recall", mientras que para la clase "Nivel bajo" se obtuvo 1 y 0.35 respectivamente en ambas métricas. Este algoritmo exhibió un rendimiento destacado con un 97.10% de precisión en la etapa de entrenamiento, un 97.02% en el testeo y un 90.9% en la fase de prueba, resultados que respaldan de manera sólida la elección y aplicación de este modelo.

- En la implementación del sistema electrónico destinado a la adquisición y monitoreo de señales electrofisiológicas mediante Machine Learning, se llevó a cabo un análisis técnico exhaustivo de los requisitos tanto del sistema electrónico como de la aplicación móvil asociada. Este proceso culminó en la selección cuidadosa de sensores apropiados para captar las señales electrofisiológicas y la elección estratégica del microcontrolador ESP32 Tiny S3. La elección de la ESP32 Tiny S3 no solo se basó en su sólido rendimiento en el procesamiento de las señales, sino también en su capacidad para integrarse de manera fluida con los demás componentes del sistema.
- El sistema inalámbrico desarrollado para la detección de estrés académico se ha demostrado eficaz y confiable. Se logró un índice de desviación estándar de 0.02 en repetidas mediciones de la variable GSR y un índice de 0.03 en la variable EMG. Estos valores, siendo relativamente bajos, respaldan la contundencia y estabilidad de las mediciones del dispositivo, lo que permite clasificar al dispositivo como funcional.
- El sistema demostró una confiabilidad del 93,94%, consolidándose como un prototipo funcional capaz de evaluar el nivel de estrés académico en estudiantes universitarios. Esta capacidad contribuye significativamente a prevenir la sobrecarga de estrés en el organismo y a evitar posibles consecuencias graves derivadas de la falta de conciencia sobre su estado mental. Al proporcionar a los usuarios información sobre su nivel de estrés, la aplicación móvil asociada facilita la entrega de consejos personalizados para gestionar el estrés y fomentar la relajación.

4.2 Recomendaciones

- Para el entrenamiento de algoritmos de machine Learning se recomienda tener un Dataset con un número de datos superior a 3000, debido a que el proceso de machine Learning será más efectivo y tendrá más control.
- Se sugiere llevar a cabo una evaluación exhaustiva del modelo de machine Learning mediante el análisis de al menos 10 modelos diferentes. Este enfoque

permitirá identificar el modelo óptimo, reduciendo la posibilidad de sobreajuste durante el proceso de entrenamiento.

- Para la adquisición de sensores o componentes electrónicos se recomienda analizar el Datasheet de cada uno, y en conjunto se debe analizar los requerimientos en función de voltaje y corriente para analizar los requisitos del sistema electrónico en conjunto.
- Se recomienda llevar a cabo una exhaustiva investigación que involucre varios estudios y artículos académicos sobre las variables electrofisiológicas que se van a estudiar. Contar con información relevante y confiable facilitará la calibración de los sensores seleccionados para los propósitos previstos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] OMS, «Salud Mental: Estrés», Organización Mundial de la Salud, Informativo, 2022. [En línea]. Disponible en: <https://apps.who.int/iris/rest/bitstreams/1433519/retrieve>
- [2] ONU, «Salud mental», Organización de las Naciones Unidas, Informativo, 2022. [En línea]. Disponible en: <https://news.un.org/es/tags/salud-mental>
- [3] C. A. Román, F. Ortiz, y Y. Hernández, «El estrés académico en estudiantes latinoamericanos de la carrera de Medicina», *Rev. Iberoam. Educ.*, vol. 46, n.º 7, Art. n.º 7, jul. 2008, doi: 10.35362/rie4671911.
- [4] ENSANUT, «Encuesta Nacional de Salud y nutrición», Informativo, 2022. [En línea]. Disponible en: <https://www.salud.gob.ec/encuesta-nacional-de-salud-y-nutricion-ensanut/>
- [5] A. Barraza, «Características del estrés académico», Colombia, 2019.
- [6] V. Tomalá, «Sistema electrónico de monitoreo de niveles de estrés académico», Tesis, Universidad Técnica de Ambato, Ambato, 2018.
- [7] C. N. Cantos, «Detección de los niveles de estrés y ansiedad en pilotos aplicando técnicas de Machine Learning».
- [8] D. Sandoval, «Diseño e implementación de un sistema para la detección del estrés mediante redes neuronales convolucionales a partir de imágenes térmicas», Tesis, Universidad Politécnica de Madrid, España, 2019.
- [9] C. Navarro, «Desarrollo de un sistema wearable para la detección de estrés mediante aprendizaje automático basado en ritmo cardíaco», Tesis, Universidad de Concepción, Chile, 2021.
- [10] S. Tutillo, «Sistema Biofeedback de sensores de detección de estrés mediante algoritmos de aprendizaje de máquinas en estudiantes universitarios», Tesis, Universidad Técnica del Norte, Ibarra, 2021.

- [11] S. Cohen, R. Kessler, y U. Gordon, *Measuring Stress*. Oxford University Press, 1995.
- [12] C. A. Martínez, *ESTRES. Aspectos médicos*, Segunda edición. Madrid: Instituto Nacional de Seguridad e Higiene en el Trabajo.
- [13] S. Nogareda, «Fisiología del estrés», Ministerio del Trabajo España, España, Informativo, 2016.
- [14] F. Duval, F. Gonzales, y H. Rabia, «Neurobiología del estrés», *SONEPSYN*, pp. 307-318, 2010.
- [15] W. van den Broek, «Neurobiology of Psychosocial Stress and Depression». [En línea]. Disponible en: <https://www.shockmd.com/wp-content/dorsolateral-prefrontal-cortex3.jpg>
- [16] «Trastorno por estrés postraumático: Cuando el cerebro sufre un trauma - NeuroClass». Accedido: 26 de diciembre de 2023. [En línea]. Disponible en: <https://neuro-class.com/estres-postraumatico-cuando-el-cerebro-sufre-un-trauma/>
- [17] N. E. Zárate-Depraect, M. G. Soto-Decuir, E. G. Martínez-Aguirre, M. L. Castro-Castro, R. A. García-Jau, y N. M. López-Leyva, «Hábitos de estudio y estrés en estudiantes del área de la salud», *Rev. Fund. Educ. Médica*, vol. 21, n.º 3, p. 153, 2018, doi: 10.33588/fem.213.948.
- [18] A. Barraza, «El estrés académico en los alumnos de postgrado», *Psicología científica*, 2010.
- [19] A. Barraza, «Estrés académico: un estado de la cuestión.», *Psicología científica*, 2009.
- [20] L. O. Olivas Ugarte, S. F. Morales-Hernández, y M. K. Solano-Jáuregui, «Evidencias psicométricas de Inventario SISCO SV-21 para el estudio del estrés académico en universitarios peruanos», *Propósitos Represent.*, vol. 9, n.º 2, 2021, doi: 10.20511/pyr2021.v9n2.647.

- [21] L. A. Alvarez Silva, R. M. Gallegos Luna, y P. S. Herrera López, «Estrés académico en estudiantes de tecnología superior», *Universitas*, n.º 28, pp. 193-209, feb. 2018, doi: 10.17163/uni.n28.2018.10.
- [22] A. Barraza, *INVENTARIO SISCO SV-21 Inventario SISTémico COgnoscitivista para el estudio del estrés académico. Segunda versión de 21 ítems*, Primera edición. México: Sello Editorial ECORFAN: 607-8534, 2018.
- [23] NIH, «I'm So Stressed Out!», *Natl. Inst. Ment. Health*, n.º NIH Publication No. 20-MH-8125.
- [24] B. E. Griffith y C. S. Peskin, «Electrophysiology», *Commun. Pure Appl. Math.*, vol. 66, n.º 12, doi: <https://doi.org/10.1002/cpa.21484>.
- [25] M. A. Cavalcanti Garcia y T. M. Vieira, «Surface electromyography: Why, when and how to use it», *I*, vol. 4, pp. 17-28, 2011.
- [26] D. C. Fowles, M. J. Christie, R. Edelberg, W. W. GRINGS, y D. T. Lykken, «Publication Recommendations for Electrodermal Measurements», *Psychophysiology*, vol. 18, n.º 3, pp. 232-239, 1981.
- [27] M. E. Moncada y J. de la Cruz, «La actividad electrodérmica - Revisión», *Ing. E Investig.*, vol. Vol 31, n.º No.2, pp. 143-151, ago. 2011.
- [28] M. V. Villarejo, B. G. Zapirain, y A. M. Zorrilla, «A Stress Sensor Based on Galvanic Skin Response (GSR) Controlled by ZigBee», *Sensors*, vol. 12, n.º 5, Art. n.º 5, may 2012, doi: 10.3390/s120506075.
- [29] G. Wolterink *et al.*, «Development of Soft sEMG Sensing Structures Using 3D-Printing Technologies», *Sensors*, vol. 20, n.º 15, 2020, doi: 10.3390/s20154292.
- [30] R. Merletti, A. Botter, A. Troiano, E. Merlo, y M. A. Minetto, «Technology and instrumentation for detection and conditioning of the surface electromyographic signal: State of the art», *Clin. Biomech.*, vol. 24, n.º 2, pp. 122-134, feb. 2009, doi: 10.1016/j.clinbiomech.2008.08.006.

- [31] P. Salvo, R. Raedt, E. Carrette, D. Schaubroeck, J. Vanfleteren, y L. Cardon, «A 3D printed dry electrode for ECG/EEG recording», *Sens. Actuators Phys.*, vol. 174, pp. 96-102, feb. 2012, doi: 10.1016/j.sna.2011.12.017.
- [32] M. B. I. Reaz, M. Hussain, y F. Mohd-Yasin, «Techniques of EMG signal analysis: detection, processing, classification and applications (Correction)», *Biol. Proced. Online*, vol. 8, p. 163, oct. 2006, doi: 10.1251/bpo124.
- [33] L. Guasch, *Electrotecnia*. Madrid: McGraw-Hill Interamericana de España S.L, 2006.
- [34] W. Tomasi, *Sistemas de comunicaciones electrónicas [recurso electrónico]*, 4th ed. México: Pearson Educación, 2003.
- [35] E. Nautiques, «La integración de sistemas de información», Barcelona, 2016.
- [36] R. Pallas, *Sensores y Acondicionadores*, 4.^a ed. Marcombo. [En línea]. Disponible en: <https://web.instipp.edu.ec/Libreria/libro/Sensores%20y%20Acondicionadores.pdf>
- [37] A. Galipenso, *Inteligencia artificial: modelos, técnicas y áreas de aplicación*. Madrid: Paraninfo, 2016.
- [38] P. Jackson, «Introduction to artificial intelligence», 2019.
- [39] C. Manning y H. Schütze, *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.
- [40] S. Russell y P. Norvig, *Artificial Intelligence: A Modern Approach*, Tercera. Pearson, 2015.
- [41] A. Franco, «Deep Learning», Universidad de Sevilla, España, 2019. [En línea]. Disponible en: <https://idus.us.es/bitstream/handle/11441/90004/Centeno%20Franco%20Alba%20TFG.pdf?sequence=1&isAllowed=y>
- [42] G. Bonaccorso, «Machine learning algorithms», *Packt Publ. Ltd*, 2017.

- [43] D. Ramirez, «EL MACHINE LEARNING A TRAVÉS DE LOS TIEMPOS, Y LOS APORTES», Colombia, 2018. [En línea]. Disponible en: <https://repository.unilibre.edu.co/bitstream/handle/10901/17289/EL%20MACHINE%20LEARNING.pdf?sequence=1&isAllowed=y>
- [44] A. Mishra, «Metrics to Evaluate your Machine Learning Algorithm». Accedido: 9 de enero de 2024. [En línea]. Disponible en: <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>
- [45] M. Batta, «Machine Learning Algorithms - A Review», » *International Journal of Science and Research (IJSR)*, vol. 9, pp. 381-386, 2019.
- [46] S. Prasanna, «Machine Learning with Python».
- [47] «Mapa de Calor de Correlaciones en Python». Accedido: 11 de enero de 2024. [En línea]. Disponible en: <https://www.codetodevs.com/como-crear-mapa-calor-correlaciones-python-corrcoef-heatmap/>
- [48] M. C. Gasca Mantilla, L. L. Camargo Ariza, y B. Medina Delgado, «Metodología para el desarrollo de aplicaciones móviles», *Rev. Tecnura*, vol. 18, n.º 40, p. 20, may 2014, doi: 10.14483/udistrital.jour.tecnura.2014.2.a02.
- [49] R. Canto, «STANDARDS FOR THE VISUAL AND INFORMATIONAL DESIGN OF MOBILE APPLICATIONS», vol. 5, n.º 2, 2020.
- [50] UNE, «UNE-EN ISO 9241-11:2018 (Ratificada) Ergonomía de la interacci...» Accedido: 15 de enero de 2024. [En línea]. Disponible en: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0060329>
- [51] J. Molina, *Database Systems:*
- [52] C. Peng y Z. Jiang, «Building a Cloud Storage Service System», *2011 3rd Int. Conf. Environ. Sci. Inf. Appl. Technol. ESIAT 2011*, vol. 10, pp. 691-696, ene. 2011, doi: 10.1016/j.proenv.2011.09.111.
- [53] N. Canal, «Técnicas de muestreo. Sesgos más frecuentes».

- [54] A. Technologies, «Three-lead Differential Muscle / Electromyography Sensor for Microcontroller Applications», Datasheet, 2013.
- [55] A. Technologies, «MyoWare™ Muscle Sensor (AT-04-001)», Datasheet, 2015. [En línea]. Disponible en: <https://www.digikey.com/htmldatasheets/production/1897318/0/0/1/MyoWare-Muscle-Sensor-AT-04-001-.pdf>
- [56] DFROBOT, «Gravity: Analog EMG Sensor by OYMotion», Datasheet. [En línea]. Disponible en: <https://www.farnell.com/datasheets/3176112.pdf>
- [57] S. Grove, «Grove - GSR Sensor», Datasheet. [En línea]. Disponible en: https://files.seeedstudio.com/wiki/Grove-GSR_Sensor/res/Grove-GSR_Sensor_WiKi.pdf
- [58] «CJMCU-6701 GSR Skin Sensor Module Analog SPI 3.3V/5V – Robocraze». Accedido: 27 de diciembre de 2023. [En línea]. Disponible en: <https://robocraze.com/products/cjmcu-6701-gsr-skin-sensor-module-analog-spi-3-3v-5v>
- [59] Thechtonics, «GSR Module Current Sensor Kit». Accedido: 28 de diciembre de 2023. [En línea]. Disponible en: <https://www.techtonics.in/gsr-galvanic-skin-response-module-current-sensor-kit-v2-0>
- [60] T. Handson, «ESP8266 NodeMCU WiFi Development Board», Datasheet. [En línea]. Disponible en: <https://handsontec.com/dataspecs/module/esp8266-V13.pdf>
- [61] M. Unexpected, «Introducing the TinyS3 by Unexpected Maker». Accedido: 28 de diciembre de 2023. [En línea]. Disponible en: <https://esp32s3.com/tinys3.html>
- [62] A. Bruno, «ESP32 NODE MCU», Datasheet, 2019. [En línea]. Disponible en: https://www.microelectronicash.com/downloads/ESP32_MANUAL.pdf
- [63] J. Healey, «Wearable and Automotive Systems for Affect Recognition from Physiology», MASSACHUSETTS INSTITUTE OF TECHNOLOGY, MASSACHUSETTS, USA, 2000.

[64] P. Shruti, «Explainable Artificial Intelligence for Intrusion Detection System», *Electronics* 2022, vol. 11, n.º 3079. Accedido: 5 de enero de 2024. [En línea]. Disponible en: <https://doi.org/10.3390/electronics111930790.3390/>

[65] M. Galarnyk, «Python_Tutorials/Sklearn/Train_Test_Split/TrainTestSplitScikitLearn.ipynb at master · mGalarnyk/Python_Tutorials · GitHub». Accedido: 6 de enero de 2024. [En línea]. Disponible en: https://github.com/mGalarnyk/Python_Tutorials/blob/master/Sklearn/Train_Test_Split/TrainTestSplitScikitLearn.ipynb

[66] «Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap - ScienceDirect». Accedido: 5 de enero de 2024. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/abs/pii/S0167947309001601?via%3Dihub>

[67] scikit-learn, «sklearn.ensemble.RandomForestClassifier — documentación de scikit-learn - 0.24.1». Accedido: 17 de enero de 2024. [En línea]. Disponible en: <https://qu4nt.github.io/sklearn-doc-es/modules/generated/sklearn.ensemble.RandomForestClassifier.html?highlight=random#sklearn.ensemble.RandomForestClassifier>

ANEXOS

Anexo A. Instrumento del Inventario SISCO SV-21

1.- Durante el transcurso de este semestre ¿has tenido momentos de preocupación o nerviosismo (estrés)?

- Si
 No

2.- Con la idea de obtener mayor precisión y utilizando una escala del 1 al 5 señala tu nivel de estrés, donde (1) es poco y (5) mucho.

1	2	3	4	5

3.-Dimensión estresores

Instrucciones: A continuación, se presentan una serie de aspectos que, en mayor o menor medida, suelen estresar a algunos alumnos. Responde, señalando con una X, ¿con que frecuencia cada uno de esos aspectos te estresa? tomando en consideración la siguiente escala de valores:

Nunca	Casi nunca	Rara vez	Algunas veces	Casi siempre	Siempre
N	CN	RV	AV	CS	S

¿Con qué frecuencia te estresa:

Estresores	N	CN	RV	AV	CS	S
La sobrecarga de tareas y trabajos escolares que tengo que realizar todos los días						
La personalidad y el carácter de los/as profesores/as que me imparten clases						
La forma de evaluación de mis profesores/as (a través de ensayos, trabajos de investigación, búsquedas en Internet, etc.)						
El nivel de exigencia de mis profesores/as						
El tipo de trabajo que me piden los profesores (consulta de temas, fichas de trabajo, ensayos, mapas conceptuales, etc.)						
Tener tiempo limitado para hacer el trabajo que me encargan los/as profesores/as						
La poca claridad que tengo sobre lo que quieren los/as profesores/as						

4.- Dimensión síntomas (reacciones)

Instrucciones: A continuación, se presentan una serie de reacciones que, en mayor o menor medida, suelen presentarse en algunos alumnos cuando están estresados. Responde, señalando con una X, ¿con que frecuencia se te presentan cada una de estas reacciones cuando estás estresado? tomando en consideración la misma escala de valores del apartado anterior.

Con qué frecuencia se te presentan las siguientes reacciones cuando estás estresado:

Síntomas	N	CN	RV	AV	CS	S
Fatiga crónica (cansancio permanente)						
Sentimientos de depresión y tristeza (decaído)						
Ansiedad, angustia o desesperación						
El nivel de exigencia de mis profesores/as						
Problemas de concentración						
Sentimiento de agresividad o aumento de irritabilidad						
Conflictos o tendencia para discutir						
Desgano para realizar las labores escolares						

5.- Dimensión estrategias de afrontamiento

Instrucciones: A continuación, se presentan una serie de acciones que, en mayor o menor medida, suelen utilizar algunos alumnos para enfrentar su estrés. Responde, encerrando en un círculo, ¿con qué frecuencia utilizas cada una de estas acciones para enfrentar tu estrés? tomando en consideración la misma escala de valores del apartado anterior.

¿Con qué frecuencia utilizas cada una de estas acciones para enfrentar tu estrés:

Estrategias	N	CN	RV	AV	CS	S
Concentrarse en resolver la situación que me preocupa						
Establecer soluciones concretas para resolver la situación que me preocupa						
Analizar lo positivo y negativo de las soluciones pensadas para solucionar la situación que me preocupa						
Mantener el control sobre mis emociones para que no me afecte lo que me estresa						

Anexo B. Certificado de validación del instrumento

CERTIFICADO

Ambato, 02 de enero de 2024

Ingeniera, Mg.

Pilar Urrutia

DECANA


Facultad de Ingeniería en Sistemas, Electrónica e Industrial

Presente. -

Yo, Marco Vinicio Ramírez Enríquez, con cedula de ciudadanía N°1802475473, con registro del Senescyt N°1027-12-1176419, de especialidad Psicología Clínica, certifico que he revisado el instrumento de evaluación denominado: Inventario SISTémico COgnoscitivista para el estudio del estrés académico, para la recolección de información para el proyecto **“DETECCIÓN DE NIVELES DE ESTRÉS EN ESTUDIANTES UNIVERSITARIOS A PARTIR DE SEÑALES ELECTROFISIOLÓGICAS MEDIANTE TÉCNICAS DE MACHINE LEARNING”**, el mismo que fue desarrollado por la Srta. Shirley Lizbeth Pazmiño de la Cruz, con cedula de identidad N° 1850034057, estudiante de la carrera de Telecomunicaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Tecnica de Ambato.

Es así como después de haber realizado las revisiones necesarias me permito validar el instrumento de evaluación para su aplicación a estudiantes universitarios.

Saludos cordiales,



Vinicio Ramírez
PSICÓLOGO CLÍNICO
Cod. MSP. 1802475473

Psc. Clin. Marco Vinicio Ramírez Enríquez

Psicología Clínica-Magíster en Intervención Psicológica

CI: 1802475473

CEL: 0987997813

Anexo C. Procesamiento del instrumento para estrés académico

P1. Nivel de estrés	P2. Sobrecarga de tareas y trabajos	P3. Forma de evaluación	P4. Tipo de trabajo	P5. Tiempo limitado	P6. Fatiga crónica	P7. Problemas de concentración	P8. Desgano	TOTAL (%)	NIVEL DE ESTRÉS
4	4	4	4	4	2	2	2	65	moderado
4	2	2	2	4	2	1	1	45	moderado
5	3	3	3	3	3	3	3	65	moderado
5	1	3	1	3	3	3	1	50	moderado
5	5	4	5	4	5	0	5	82,5	alto
5	5	4	4	2	5	4	5	85	alto
4	4	2	2	5	4	4	5	75	alto
4	5	5	5	5	1	1	5	77,5	alto
5	5	2	4	4	5	5	5	87,5	alto
4	5	5	5	5	5	5	5	97,5	alto
3	2	2	2	4	0	2	2	42,5	moderado
4	3	3	2	3	4	2	3	60	moderado
4	3	4	3	4	3	2	3	65	moderado
5	3	1	1	3	3	3	1	50	moderado
5	4	4	2	4	3	2	3	67,5	alto
4	5	3	1	5	5	4	5	80	alto
3	3	5	3	1	4	3	2	60	moderado
4	2	4	3	4	4	4	3	70	alto
3	3	3	2	4	4	3	2	60	moderado
4	4	5	5	5	4	4	4	87,5	alto
2	3	4	4	3	0	4	0	50	moderado
3	1	2	1	2	3	2	2	40	moderado
4	3	3	2	3	3	1	3	55	moderado
4	4	4	2	2	3	4	1	60	moderado
4	5	5	5	4	2	5	2	80	alto
5	3	3	3	1	4	4	4	67,5	alto
1	5	4	2	2	3	5	2	60	moderado
5	5	5	4	4	5	4	4	90	alto
4	4	5	5	5	4	5	5	92,5	alto
3	3	3	3	3	3	3	3	60	moderado
5	5	5	5	5	5	5	5	100	alto
4	3	3	3	3	1	3	1	52,5	moderado
5	5	4	5	5	3	5	5	92,5	alto
5	5	2	5	5	4	4	0	75	alto
4	5	5	5	4	4	5	1	82,5	alto
4	4	4	2	4	3	4	4	72,5	alto
4	5	5	5	5	2	2	2	75	alto
5	4	2	4	4	4	4	4	77,5	alto
4	4	4	4	4	2	2	2	65	moderado

P1. Nivel de estrés	P2. Sobrecarga de tareas y trabajos	P3. Forma de evaluacion	P4. Tipo de trabajo	P5. Tiempo limitado	P6. Fatiga crónica	P7. Problemas de concentración	P8. Desgano	TOTAL (%)	NIVEL DE ESTRÉS
5	1	1	1	1	3	3	3	45	moderado
4	4	4	3	3	3	2	2	62,5	moderado
4	4	4	5	3	4	5	4	82,5	alto
3	2	3	3	3	1	1	1	42,5	moderado
4	4	5	5	4	3	3	1	72,5	alto
4	3	2	2	4	1	0	1	42,5	moderado
4	2	2	3	3	4	2	2	55	moderado
5	4	2	4	5	2	5	4	77,5	alto
5	4	4	4	4	4	4	4	82,5	alto
5	4	5	2	4	4	2	4	75	alto
4	4	5	5	4	4	4	4	85	alto
3	3	3	3	3	1	3	3	55	moderado
4	3	4	5	4	1	4	3	70	alto
5	4	4	5	5	4	4	2	82,5	alto
4	3	3	3	3	3	3	3	62,5	moderado
5	5	5	5	5	0	4	5	85	alto
4	3	3	3	3	3	3	3	62,5	moderado
5	3	3	1	1	3	3	1	50	moderado
4	4	4	4	4	2	4	4	75	alto
4	4	2	5	5	5	5	5	87,5	alto
5	4	2	4	4	1	2	3	62,5	moderado
2	1	1	1	3	1	3	1	32,5	bajo
5	4	4	4	4	4	4	4	82,5	alto
5	4	5	4	4	4	4	4	85	alto
5	4	4	4	4	2	3	0	65	moderado
5	3	3	3	3	3	1	1	55	moderado
2	3	3	2	3	3	4	3	57,5	moderado
3	3	1	1	3	1	3	1	40	moderado
3	5	4	2	4	0	1	4	57,5	moderado
3	3	1	3	2	1	2	2	42,5	moderado
4	2	3	3	2	4	3	3	60	moderado
5	4	4	4	5	4	3	3	80	alto
5	3	3	3	3	3	3	2	62,5	moderado
5	4	5	5	5	2	2	4	80	alto
5	5	5	5	2	5	5	5	92,5	alto
5	4	4	4	5	4	3	3	80	alto
3	3	4	4	2	4	3	4	67,5	alto
4	3	5	5	5	2	4	4	80	alto
3	1	1	1	1	1	1	1	25	bajo
3	4	4	4	5	1	3	3	67,5	alto
5	4	4	4	4	2	2	2	67,5	alto

P1. Nivel de estrés	P2. Sobrecarga de tareas y trabajos	P3. Forma de evaluacion	P4. Tipo de trabajo	P5. Tiempo limitado	P6. Fatiga crónica	P7. Problemas de concentración	P8. Desgano	TOTAL (%)	NIVEL DE ESTRÉS
5	4	4	5	5	2	1	5	77,5	alto
3	3	1	1	1	1	1	1	30	bajo
5	4	2	5	2	2	4	3	67,5	alto
4	1	3	3	1	1	1	1	37,5	moderado
3	3	4	3	4	3	3	2	62,5	moderado
5	3	3	3	3	1	3	1	55	moderado
5	3	4	4	4	3	3	2	70	alto
4	4	4	2	4	4	2	5	72,5	alto
4	3	1	4	4	0	3	0	47,5	moderado
5	4	3	4	4	4	2	3	72,5	alto
4	4	1	4	3	2	1	3	55	moderado
4	4	3	3	1	3	3	3	60	moderado
4	4	5	2	4	4	2	3	70	alto
5	5	5	5	5	4	4	5	95	alto
4	4	3	4	5	5	4	4	82,5	alto
4	3	3	3	3	3	3	3	62,5	moderado
4	5	5	5	5	5	5	5	97,5	alto
4	4	4	5	5	2	4	4	80	alto
5	5	5	5	4	4	5	4	92,5	alto
4	3	3	3	3	3	3	3	62,5	moderado
4	3	3	1	3	1	1	3	47,5	moderado
4	4	4	4	4	4	4	4	80	alto
4	5	5	4	5	5	5	5	95	alto
4	4	4	2	4	4	4	4	75	alto
3	1	1	1	3	3	3	1	40	moderado
5	5	5	5	5	5	5	3	95	alto
3	1	4	2	2	2	2	3	47,5	moderado
1	1	1	1	1	1	1	0	17,5	bajo
4	4	4	4	4	2	4	2	70	alto
4	4	1	1	2	2	2	2	45	moderado
4	4	2	3	3	2	3	3	60	moderado
2	3	3	1	3	0	1	0	32,5	bajo
3	3	3	1	3	2	2	2	47,5	moderado
3	3	3	3	3	3	3	3	60	moderado
4	4	4	4	5	2	4	4	77,5	alto
5	3	3	3	3	3	3	1	60	moderado
5	5	5	4	5	4	2	4	85	alto
3	4	2	1	2	3	2	2	47,5	moderado
3	3	3	4	4	2	3	3	62,5	moderado
3	1	1	1	3	1	1	1	30	bajo
3	2	2	2	2	1	4	2	45	moderado

Anexo D. Datasheet ESP32 Tiny S3



Features

- Dual 32bit Xtensa LX7 cores @ up to 240Mhz
- RISC-V Ultra Low Power Co-processor
- 2.4GHz Wifi - 802.11b/g/n
- Bluetooth 5, BLE + Mesh
- 8MB QSPI Flash
- 8MB of extra QSPI PSRAM
- 700mA 3.3V LDO Regulator
- Low power RGB LED
- **ULTRA LOW Deep Sleep Current**
- USB-C Connector with back-feed protection
- Native USB + USB Serial JTAG
- LiPo Battery Charging
- JST pads on the back supports PH connector
- VBAT and 5V Sense Pins

Anexo E. Datasheet sensor GSR

Technical Details

Dimensions	24mm x 20mm x 9.80mm
Weight	G.W 29g
Battery	Exclude
Input Voltage	5V/3.3V
Sensitivity adjustable via a potentiometer	

Part List

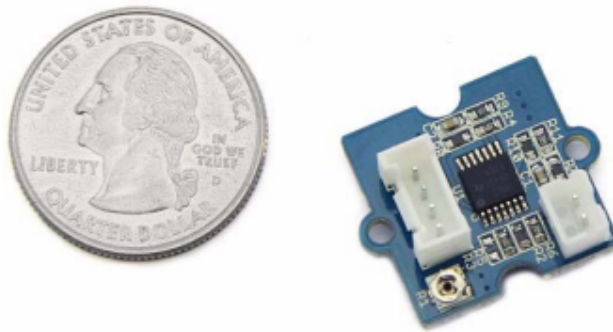
Grove - GSR Sensor	1
--------------------	---

ECCN/HTS

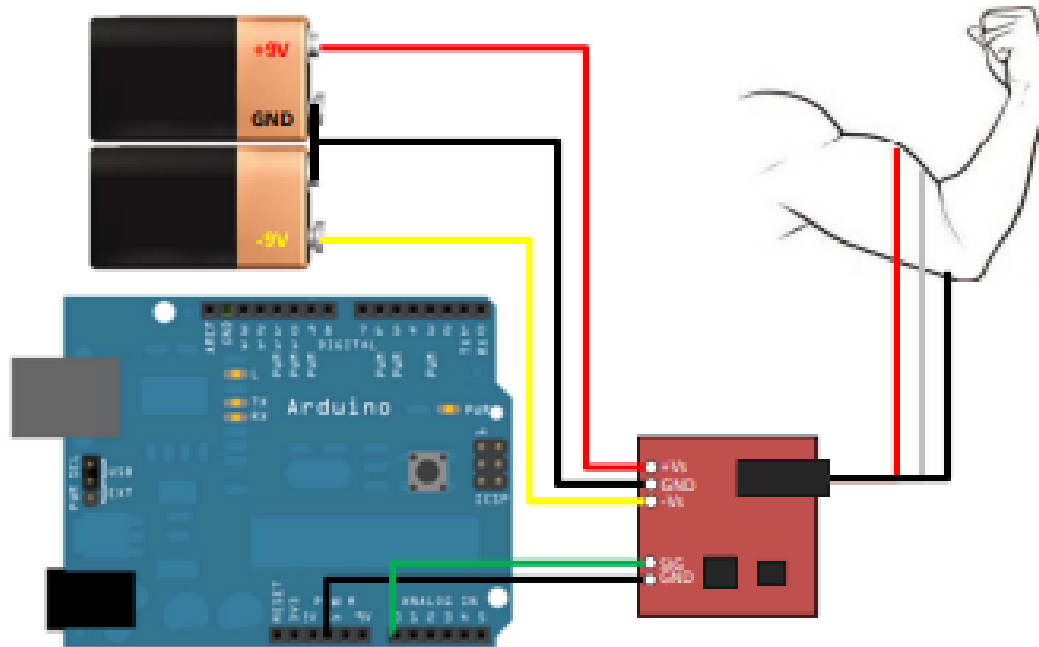
ECCN	EAR99
HSCODE	9031900090

Certification

RoHS	Compliant
------	-----------



Getting Started Using Two 9V Batteries



- 1) Connect the power supply (two 9V batteries)
 - a. Connect the positive terminal of the first 9V battery to the +Vs pin on your sensor.
 - b. Connect the negative terminal of the first 9V battery to the positive terminal of the second 9V battery. Then connect to the GND pin on your sensor.
 - c. Connect the negative terminal of the second 9V battery to the -Vs pin of your sensor.
- 2) Connect the electrodes
 - a. After determining which muscle group you want to target (e.g. bicep, forearm, calf), clean the skin thoroughly.
 - b. Place one electrode in the middle of the muscle body, connect this electrode to the RED Cable's snap connector.
 - c. Place a second electrode at one end of the muscle body, connect this electrode to the Blue Cable's snap connector.
 - d. Place a third electrode on a bony or non-muscular part of your body near the targeted muscle, connect this electrode to the Black Cable's snap connector.
- 3) Connect to a Microcontroller (e.g. Arduino)
 - a. Connect the SIG pin of your sensor to an analog pin on the Arduino (e.g. A0)
 - b. Connect the GND pin of your sensor to a GND pin on the Arduino.

Anexo G. Muestra del Dataset usado para el entrenamiento

EMG	GSR	P1	P2	P3	P4	P5	P6	P7	P8	NIVEL
2.22	6.36	4	4	4	4	4	2	2	2	2
1.89	6.99	4	2	2	2	4	2	1	1	1
0.51	0.27	5	3	3	3	3	3	3	3	1
4.8	1.02	5	1	3	1	3	3	3	1	1
1.29	3.36	5	5	4	5	4	5	0	5	2
3.03	6.24	5	5	4	4	2	5	4	5	2
4.9	6.57	4	4	2	2	5	4	4	5	2
1.9	1.08	4	5	5	5	5	1	1	5	2
2	1.59	5	5	2	4	4	5	5	5	2
1.55	6.63	4	5	5	5	5	5	5	5	2
3.37	1.2	3	2	2	2	4	0	2	2	1
4.66	3.67	4	3	3	2	3	4	2	3	2
1.21	0.4	4	3	4	3	4	3	2	3	1
3.68	1.8	5	3	1	1	3	3	3	1	1
0.44	1.96	5	4	4	2	4	3	2	3	1
4.57	5.75	4	5	3	1	5	5	4	5	2
4.63	4.73	3	3	5	3	1	4	3	2	2
4.5	4.82	4	2	4	3	4	4	4	3	2
2.44	6.31	3	3	3	2	4	4	3	2	1
3.8	6	4	4	5	5	5	4	4	4	2
3.2	0.54	2	3	4	4	3	0	4	0	1
2.99	2.03	3	1	2	1	2	3	2	2	1
3.49	0.78	4	3	3	2	3	3	1	3	1
2.44	2.6	4	4	4	2	2	3	4	1	1

Anexo H. Código de la conexión del microcontrolador

```
#include <WiFi.h>
#include <HTTPClient.h>

#define HOST "shirleytesisbd.000webhostapp.com"
#define WIFI_SSID ""
#define WIFI_PASSWORD ""

const int emgPin = 2;
const int sensorPin = 1;
const float VREF_EMG = 5.0;
const float ADC_RESOLUTION_EMG = 1023.0;
const float SCALING_FACTOR_EMG = 3.0 / VREF_EMG;
const float extra = 1.2;

float VREF_GSR = 3.3;
int R_CALIBRACION = 47000;
int GSR_CONSTANT = 100000;

float val;
float val2;

String sendval, sendval2, postData;

void setup() {
  Serial.begin(115200);
  Serial.println("Communication Started\n\n");
  delay(1000);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is: ");
  Serial.println(WiFi.localIP());
  delay(30);
}

void loop() {
```

```

float val = readEMGSensor();
float val2 = readGSRSensor();
HTTPClient http;
WiFiClient client;

sendval = String(val);
sendval2 = String(val2);

postData = "emg=" + sendval + "&gsr=" + sendval2;

http.begin(client,
"http://shirleytesisbd.000webhostapp.com/sensores.php");
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
delay (4000);

int httpCode = http.POST(postData);
Serial.println("Values are, sendval = " + sendval + " and sendval2 = "
+ sendval2);

if (httpCode == HTTP_CODE_OK) {
    Serial.println("Values uploaded successfully.");
    String webpage = http.getString();
    Serial.println(webpage + "\n");
} else {
    Serial.println("Failed to upload values.");
    Serial.println(httpCode);
}

http.end();
}

float readEMGSensor() {
    int emgValue = analogRead(emgPin);
    float scaledValue = ((emgValue / ADC_RESOLUTION_EMG) * VREF_EMG *
SCALING_FACTOR_EMG) + extra;
    Serial.print("Valor del sensor EMG: ");
    Serial.println(scaledValue, 2);

    return scaledValue;
}

float readGSRSensor() {
    int sensorValue = analogRead(sensorPin);
    float voltage = (sensorValue / 4095.0) * VREF_GSR;
    float conductance = (voltage / R_CALIBRACION) / (VREF_GSR - voltage);
    float microsiemens = conductance * GSR_CONSTANT;

```

```
Serial.print("Valor del sensor GSR ( $\mu$ S): ");  
Serial.println(microsiemens);  
  
return microsiemens;  
}
```


Anexo I. Certificado de validación del dispositivo

CERTIFICADO

Ambato, 08 de enero de 2024

Ingeniera, Mg.

Pilar Urrutia

DECANA

Facultad de Ingeniería en Sistemas, Electrónica e Industrial

Presente. -

Yo, Marco Vinicio Ramírez Enríquez, con cedula de ciudadanía N°1802475473, con registro del Senescyt N°1027-12-1176419, de especialidad Psicología Clínica, certifico que he revisado la funcionalidad del proyecto llamado **“DETECCIÓN DE NIVELES DE ESTRÉS EN ESTUDIANTES UNIVERSITARIOS A PARTIR DE SEÑALES ELECTROFISIOLÓGICAS MEDIANTE TÉCNICAS DE MACHINE LEARNING”**, el mismo que fue desarrollado por la Srta. Shirley Lizbeth Pazmiño de la Cruz, con cedula de identidad N° 1850034057, estudiante de la carrera de Telecomunicaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Tecnica de Ambato.

Es así como después de haber realizado las pruebas necesarias me permito verificar la funcionalidad del dispositivo y puedo validar que los resultados obtenidos son aceptables en cuanto a la detección de estrés, además indico que las medidas tomadas por los sensores se encuentran dentro de los rangos establecidos.

Saludos cordiales,



Psc. Clin. Marco Vinicio Ramírez Enríquez

Psicología Clínica-Magister en Intervención Psicológica

CI: 1802475473

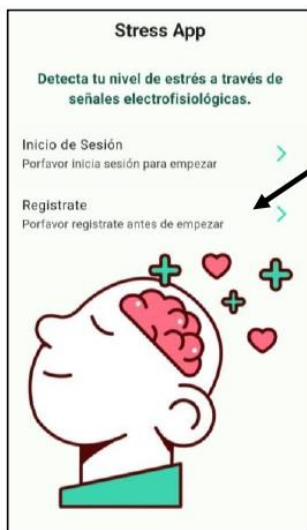
CEL: 0987997813

Manual de Usuario

Stress App



1. Abre la aplicación Stress App y crea un nuevo usuario.



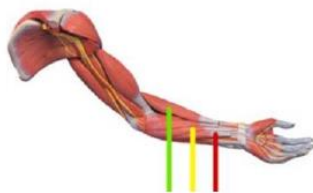
Presiona en el apartado llamado "Regístrate"



Llena el registro con tus datos

2. Coloca los electrodos en las zonas de tu brazo derecho según corresponda.

Antes de colocar los electrodos se debe limpiar la zona, con un paño húmedo para limpiar cualquier factor que puede interferir en la conductancia eléctrica.



Electrodo verde: Se debe colocar en el centro del musculo deseado.

Electrodo amarillo: Se coloca en el extremo del musculo deseado.

Electrodo rojo: Se coloca en una zona no muscular cerca del musculo deseado.



Los electrodos para los dedos se deben colocar en la parte inferior, de preferencia en el dedo indice y en el anular

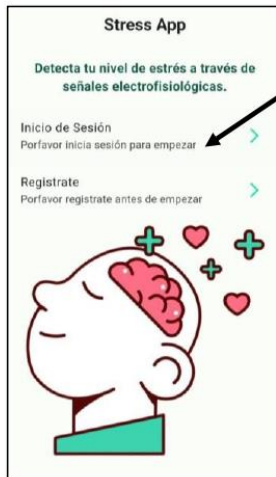
3. Enciende el dispositivo de medición.

En la parte lateral del dispositivo se encuentra un interruptor, el cual debe ser activado hacia la derecha.



Enciende el dispositivo con el interruptor.

4. Inicia sesión en la aplicación.

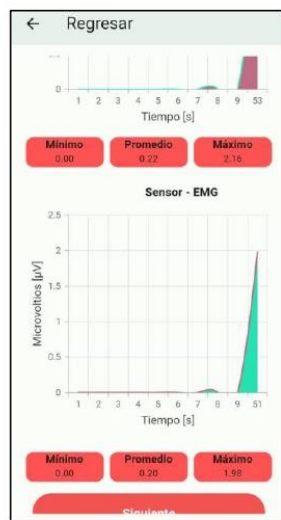


Presiona en el apartado llamado "Inicio de sesión" de



Ingresa los datos con los que te registraste y presiona "Iniciar sesión"

5. Observa como se registra la información de tus señales electrofisiológicas y espera 5 minutos hasta que la señal se estabilice.



Cuando la señal de haya estabilizado, presionan el botón de "Siguiente"

6. Responde el cuestionario en base a como te afecta el estrés en varios aspectos de tu vida académica.

Se debe responder con sinceridad para que el resultado sea preciso.

← Regresar

Responde el siguiente formulario asignando un valor, donde (1) es poco y (5) mucho según como te afecte cada parámetro, en base a tu nivel de estrés.

¿Cómo describes tu nivel actual de estrés?

0

¿Cuanto estrés sientes por la sobrecarga de tareas?

0

¿Cuanto te estresan las evaluaciones?

0

¿Cuanto te estresa el tipo de trabajo asignado?

0

Es un total de 8 preguntas. Cuando hayas terminado de responder tus preguntas, presiona el botón de siguiente para observar tus resultados

7. Observa tus resultados.

← Regresar

Resultados

Estrés moderado

Mostrar Resultados

Consejos para tu bienestar

Bajo Moderado Alto

Ver Reporte

En base al nivel de estrés detectado, selecciona tu nivel para observar los consejos para tu bienestar.

← Regresar

Resultados

Tratamiento estrés moderado:

Cuando enfrentas un nivel de estrés moderado, adopta medidas preventivas y de manejo. Equilibra tus responsabilidades, establece metas realistas y organiza tu tiempo. Integra pausas durante el estudio y practica actividades relajantes. Mantén hábitos alimenticios saludables, asegúrate de descansar adecuadamente y busca apoyo social. Esta gestión proactiva contribuye a mantener un equilibrio saludable en tu vida cotidiana.

Cerrar

Ver Reporte

8. Para analizar los resultados que se han obtenido cada vez que se ha ingresado a la aplicación.

← Regresar

Resultados

Estres moderado

Mostrar Resultados

Consejos para tu bienestar

Bajo Moderado Alto

Ver Reporte

← Regresar

Historial de Consultas

Resultados de Estres

Alto 1

Moderado 9

Bajo 2

Total 12

ID	EMG	GSR	Resultado
1	1.923	2.121	Estres moderado
2	1.923	2.121	Estres moderado
3	1.923	2.121	Estres moderado
4	1.923	2.121	Estres moderado
5	1.923	2.121	Estres moderado

Presionamos el botón de Ver Reporte.

Anexo K. Pruebas de funcionamiento del sistema

