



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA DE LA SALUD**

**INTELIGENCIA ARTIFICIAL APLICADA A UN SISTEMA DE
MONITORIZACIÓN DEL NIVEL DEL DOLOR**

Realizado por

Cristian Antonio Cabello Arango

Dirigido por

Juan Pedro Domínguez Morales

Departamento

Arquitectura y tecnología de computadores

Sevilla, junio del 2023

I- AGRADECIMIENTOS

A Dios, mi familia, compañera de proyecto y profesores.

II. RESUMEN:

Este proyecto está dirigido para todas aquellas personas que sufren dolor. Hoy en día, todavía no hay ningún método que nos diga con certeza el nivel de dolor que podemos llegar a tener. Este nuevo sistema, ayudará a la comunicación médico-paciente, lo que permitirá conocer mejor los niveles de dolor que podemos llegar a padecer. De esta forma, los médicos podrán amoldar sus métodos a sus pacientes con mayor facilidad en diferentes ámbitos de la medicina, desde, una rehabilitación, para aquellos que sufran algún tipo de rotura o daño que necesite de esta práctica, hasta, para una operación en la que el paciente este sedado. También, vemos gran aplicabilidad de este sistema en personas con dificultades a la hora de comunicarse como pueden ser aquellas con enfermedades como la parálisis cerebral, ELA o cualquier otra que provoque la pérdida del habla.

El objetivo de este proyecto es elaborar un sistema que permita la monitorización de un paciente a partir de la obtención de datos proporcionados por señales fisiológicas, tales como la frecuencia cardíaca, la frecuencia respiratoria, la respuesta galvánica de la piel y la temperatura corporal. Estos parámetros fisiológicos serán recogidos en un conjunto de entrenamiento que nos permitirá llevar a cabo un estudio de investigación, utilizando como base principal la *Inteligencia Artificial*. En otras palabras, analizaremos la información que proporciona dicho conjunto de entrenamiento utilizando técnicas de preprocesamiento y *machine learning* que nos permitan entrenar una serie de modelos que generen variables indicadoras del mejor camino para clasificar estas señales fisiológicas en función del nivel de dolor que padece el paciente, siguiendo los estándares médicos establecidos en las escalas métricas, así como un estudio del mismo, utilizando las *redes neuronales* como mecanismo para conseguir un estudio sobre el comportamiento del conjunto de datos en este algoritmo.

III. ABSTRACT:

This project is aimed at all those who suffer from pain. Nowadays, there is still no method that can accurately tell us the level of pain we may experience. This new system will help improve the doctor-patient communication, allowing for a better understanding of the levels of pain that patients may endure. In this way, doctors will be able to adapt their methods to their patients more easily in different areas of medicine, from rehabilitation for those who have suffered some kind of injury or damage that requires this practice, to sedated operations. Additionally, we see great applicability of this system in people with communication difficulties, such as those with diseases like cerebral palsy, ALS, or any other condition that causes speech loss.

The objective of this project is to develop a system that allows monitoring a patient by obtaining data from physiological signals, such as heart rate, respiratory rate, galvanic skin response, and body temperature. These physiological parameters will be collected in a training set that will enable us to conduct a research study, using Artificial Intelligence as the main basis. In other words, we will analyze the information provided by this training set using preprocessing techniques and machine learning to train a series of models that generate indicative variables for classifying these physiological signals according to the patient's level of pain, following established medical standards on metric scales. Neural networks will be used as a mechanism to achieve the expected results.

IV. ÍNDICE

I- Agradecimientos	2
II. Resumen:.....	3
III. Abstract:	4
IV. Índice.....	5
V. Índice de figuras.....	7
VI. Índice de tablas	8
Bloque I: Descripción del proyecto	9
1.1- Introducción y Motivación	9
1.2- Objetivos del proyecto:.....	13
1.2.1- Objetivos docentes:	13
1.2.2- Objetivos técnicos:.....	14
1.3- ESTADO DEL ARTE:	15
1.3.1- PMD_200.....	15
1.3.2- Monitor de dolor 2.0.....	17
1.3.3- Algómetro digital palpatronic	18
1.3.4- Patente: ES 2 170 014 B1	20
1.4- Elicitación de requisitos y análisis de riesgos.....	22
1.4.1- Requisitos funcionales.....	22
1.4.2- Requisitos no funcionales	23
1.4.3- Requisitos de información.....	24
Bloque 2: Ejecución del proyecto.....	25
2.1- Diseño del sistema:	25
2.2. Implementación:	28
2.2.1- Metodología:.....	28
2.2.2. Tecnologías empleadas:	30
2.2.2.1- Bloque Instrumental:	30
2.2.2.2- Bloque Inteligente:.....	35
2.2.3. Desarrollo:.....	36
2.2.3.1-Sprint 1:.....	37
2.2.3.2- Sprint 4: Aplicación	50
2.3. Pruebas del sistema:	54
Bloque 3: Planificación del proyecto.....	59
3.1- Planificación temporal:	59

3.1.1- Planificación temporal inicial	59
3.1.2- Planificación temporal final	61
3.2- Planificación financiera:	62
3.2.1- Planificación financiera inicial	62
3.2.1.1- Recursos materiales	62
3.2.1.2. Recursos humanos:	63
3.2.2.3- Gasto total inicial:	70
3.2.2- Planificación financiera final	70
3.2.2.1- Recursos materiales	70
3.2.2.2- Recursos humanos:	72
3.2.2.3- Gasto total final:.....	76
3.3- Estudio de mercado	79
3.3.1- Clientes potenciales	79
3.3.2. Plan de comercializaciones	80
VII- Conclusiones	84
VIII. Trabajo futuro	86
IX- Bibliografía	87
X- Anexo	89
X.I- Programación por bloques del AppInventor.....	89
X.II. LepasyVisualStudio.ino.....	91
X.III. LepasyAppInventor.ino	94

V. ÍNDICE DE FIGURAS

Figura 1: Dispositivo PMD_200.....	16
Figura 2: Interfaz de la aplicación Monitor de dolor 2.0	18
Figura 3: Diagrama de bloques del sistema.....	25
Figura 4: Leyenda del diagrama de bloques.....	26
Figura 5: Sensor de pulso.	30
Figura 6: Micrófono KY-038	31
Figura 7: Sensor de temperatura corporal MLX90614	31
Figura 8: Diagrama de bloques del sensor MLX90614	32
Figura 9: Sensor de respuesta galvánica CJMCU-6701	32
Figura 10: Arduino.....	33
Figura 11: Aplicación LEPASY.....	36
Figura 12: Circuito de adquisición de variables fisiológicas.	41
Figura 13: Diagrama de clases del circuito de adquisición de variables fisiológicas. ...	43
Figura 14: Interfaz de la aplicación que define el conjunto de entrenamiento.	45
Figura 15: Mensaje de conexión correcta entre Arduino y Aplicación.	46
Figura 16: Registro de nivel de dolor bajo.....	47
Figura 17: Registro de nivel de dolor medio	48
Figura 18: Registro de nivel de dolor alto.....	49
Figura 19: Diagrama circular. Porcentaje de características del atributo Nivel de dolor	49
Figura 20: Circuito de recogida y envío de datos.....	51
Figura 21: Aplicación que monitoriza el nivel de dolor.	53
Figura 22: Diagrama de Gantt. Planificación temporal inicial.....	60
Figura 23: Diagrama de Gantt. Planificación temporal final.	61
Figura 24: Análisis DAFO del sistema completo.	81

VI. ÍNDICE DE TABLAS

Cuadro 1: Comparación de los dispositivos analizados en el estado del arte.....	21
Cuadro 2: Rangos de las variables fisiológicas.....	40
Cuadro 3: Integración de los datos adquiridos por los sensores.....	42
Cuadro 4: Funciones para el cálculo de las variables fisiológicas.	44
Cuadro 5: Ubicación fisiológica de los sensores.....	46
Cuadro 6: Codificación de los valores del atributo Nivel de dolor en la lógica	52
Cuadro 7: Índices de la lista que almacena cada una de las variables fisiológicas en la aplicación.....	52
Cuadro 8: Ciclo 1 – Recibo de datos biomédicos correctamente desde los sensores. 55	
Cuadro 9: Ciclo 2 - Registro de nivel de dolor	55
Cuadro 10: Ciclo 3 - Almacenamiento de valores en fichero con formato .csv	56
Cuadro 11: Ciclo 1 - Recepción de datos biomédicos en la aplicación App Inventor ...	57
Cuadro 12: Ciclo 2 - Clasificación del nivel de dolor.....	58
Cuadro 13: Horario laboral del calendario del proyecto.....	59
Cuadro 14: Recursos materiales Hardware iniciales.....	63
Cuadro 15: Recursos materiales Software iniciales.	63
Cuadro 16: Salario de los recursos humanos.....	64
Cuadro 17: Salario de los recursos humanos con la cotización a la seguridad social..	65
Cuadro 18: Cálculo del gasto de los recursos en función de las horas estimadas.	69
Cuadro 23: Gasto total del proyecto.....	70
Cuadro 19: Recursos materiales Hardware finales.	72
Cuadro 20: Recursos materiales Software finales.....	72
Cuadro 21: Salario de los recursos humanos real con la cotización de la seguridad social.	73
Cuadro 22: Cálculo del gasto de los recursos en función de las horas reales.....	76
Cuadro 23: Gasto total del proyecto.....	77
Cuadro 24: Comparación del gasto total que supone cada tipo de recurso.	78
Cuadro 25: Objetivo de comparación del gasto total que supondría cada tipo de recurso.	78

BLOQUE I: DESCRIPCIÓN DEL PROYECTO

1.1- INTRODUCCIÓN Y MOTIVACIÓN

En la actualidad, los sistemas biomédicos inteligentes van adquiriendo un peso cada vez mayor dentro del complejo hospitalario. El sistema sanitario se está adaptando a un cambio tecnológico que avanza a pasos de gigantes. Cada vez surgen nuevas ideas sobre cómo poder recoger el mayor número de información sobre el paciente en el menor tiempo posible con el fin de desarrollar, definir o detectar diagnósticos o tratamientos de la forma más eficiente y mucho más segura.

Donde antes, si el doctor no encontraba explicación a los síntomas de un paciente, podía tardar meses en analizar y contrastar información hasta dar con el diagnóstico y su posterior tratamiento, ahora, hay muchas bases de datos en las que simplemente introduciendo una serie de palabras clave el doctor obtendrá una serie de información que le facilitará poder obtener un diagnóstico más temprano que permita que se pueda empezar con el tratamiento lo antes posible. Ya sabemos que muchas veces en la medicina el tiempo es muy importante, cuanto más avanzada sea la enfermedad, más complicada será curarla.

Estas personas sufren grandes dolores ya sea por, un accidente, la rehabilitación, o el postoperatorio. Estar pendientes en todo momento, administrarles sus analgésicos para que sufran el mínimo dolor posible, ser amable y atento es una muy buena forma de dignificarlos, poniéndonos en su piel, e intentando comprender su dolor. Todos sabemos que mañana podemos ser nosotros mismos los que estemos en ese lugar, ya que por desgracia estas cosas ocurren todos los días y a todas horas. De hecho, según el Portal Estadístico del Sistema Nacional de Salud, España dispone de 468 hospitales en los cuales se producen aproximadamente 4 millones de ingresos hospitalarios cuyo coste medio de cada hospitalización es de 6398 de euros, así como la realización de 82 millones de consultas cuyo tiempo de espera medio para primera consulta es de 95 días [1]. <https://www.sanidad.gob.es/estadEstudios/sanidadDatos/home.html>

Estos datos son alarmantes puesto que 95 días es mucho tiempo para que una persona pueda ser entrevistada por un médico para informar de su situación sanitaria incluso sabiendo que no todas las personas son tratadas en hospitales, hay muchas que debido a su situación son tratadas en sus propias casas, donde también es importante que estén atendidas lo mejor posible. Por ello, cada vez es mayor el número de sistemas de

monitorización a distancia que permiten al médico el estado del paciente sin necesidad de su asistencia al hospital, lo que reduciría el tiempo de espera interconsulta.

En este proyecto, nos centraremos en crear un sistema de monitorización para personas que sufren dolor, este aparato nos permitirá conocer el nivel de dolor que sufre un paciente para poder comprender sus necesidades, más allá de saber las horas a las que hay que administrarle sus medicamentos.

El dolor, como todos sabemos, no solo es un sentimiento, es una respuesta del sistema nervioso que se produce cuando algo no va bien. Normalmente se sienten pinchazos, hormigueo, ardor, picor o molestia.

El dolor puede tener distintas intensidades, por eso hay veces que es casi inexistente, y en muchas otras ocasiones es insoportable. Conocer el nivel de dolor de las personas es muy complicado, ya que, hasta que no lo sientes por ti mismo, no sabes bien de qué tipo de dolor hablamos. Por ello, podemos decir que el nivel de dolor es una variable subjetiva, que depende de muchos factores.

Todos hemos sentido dolor alguna vez en nuestra vida. De hecho, según indican algunos estudios el 29% de la población española no hospitalizada sufre algún tipo de dolor, y el 17% padece dolor crónico. [2]

Si solamente en España hay este número tan elevado de personas afectadas en su día a día por el dolor, esta cifra a nivel mundial será inimaginable.

Hoy en día, los doctores emplean distintos métodos para intentar hacerse una idea lo más cercana posible, para poder tratar al paciente y sobre todo a la hora de administrar medicamentos. El método comúnmente utilizado, es la escala visual analógica (EVA), en la que mediante una serie de dibujos entre los que se distinguen 6 niveles de dolor, situándose el nivel inferior pegado a la izquierda y el máximo nivel en la derecha, el paciente señala de manera subjetiva cual piensa que se corresponde al nivel de dolor que padece en ese instante.

Otro método también utilizado es el empleo de lo que comúnmente se conoce como dolorímetro o algómetro. Este instrumento es capaz de medir el umbral del dolor, utilizando la fuerza aplicada sobre la zona afectada, de tal manera que el médico va aplicando presión y cuando el paciente no la aguanta más se lo comunica. Pero una vez más, al igual que en la escala EVA, hasta qué punto esto llega a ser fiable del todo.

Como podemos observar, estos métodos no son precisamente exactos, muchas veces el dolor se puede cubrir o intensificar dependiendo del paciente, y no le vamos a poner

un polígrafo para saber hasta qué punto dice la verdad. Aunque, la mayoría de los pacientes no mentirían sobre este tema, sigue siendo algo subjetivo. Por ello, en este estudio analizaremos las variables fisiológicas más relacionadas con el nivel de dolor, según indiquen los profesionales para poder obtener resultados más concretos. De esta forma, también ayudaríamos a que el personal sanitario o los familiares que se hagan cargo, detecten con más exactitud el dolor cuando el paciente no se pueda comunicar, ya sea por alguna enfermedad, o porque esté en algún proceso asistencial que impliquen estados de cero comunicaciones, como por ejemplo durante la sedación en una cirugía.

El dolor y la mente van ligados, ya que esta limitación puede llegar a afectar al vínculo con familiares y amigos, esto se debe a que en numerosas ocasiones ellos tendrán que adaptarse a las nuevas necesidades de la persona que padece el dolor, hasta tal punto que pueden llegar a sentirse una carga para sus amigos, familiares, compañeros de trabajo etc. Por todo esto tienden a desarrollar frustración, estrés e incluso depresión, lo cual no solo empeora normalmente los dolores, sino que también puede producir un aumento de la presión arterial, incremento de la frecuencia respiratoria y cardíaca, así como causar tensión muscular dando lugar a sentimientos de fatiga, problemas para dormir y cambios en el apetito, entre otros.

Nuestro sistema aparte de ser un dolorímetro, dispondrá de sensores que midan las señales vitales, para que los médicos puedan llevar un control exhaustivo, previniendo así al máximo todos los problemas de salud citados anteriormente para que el dolor sea lo más llevadero posible y no se agrave aún más.

La finalidad de este proyecto es conseguir que nuestra propuesta de dolorímetro será completamente autónomo, es decir, que no se necesite de la intervención del paciente para conocer su dolor.

Para desarrollar este instrumento utilizaremos una serie de sensores, de bajo coste, que midan señales fisiológicas para ser recogidas, procesadas y transformadas a través de un microcontrolador que proporcione datos de la frecuencia cardíaca, la respiración, respuesta galvánica de la piel y temperatura corporal, para definir un conjunto de entrenamiento que, posteriormente, serán analizados para determinar un patrón que demuestre que, dependiendo del rango en el que se encuentren los parámetros, el paciente tendrá dolor bajo, medio o alto.

Hoy en día, solo hay un aparato capaz de medir el nivel de dolor de manera independiente a la subjetividad de cada paciente, y debido a su elevado precio no es

accesible para la inmensa mayoría. Con este nuevo sistema también pretendemos que, en el futuro, cualquiera que lo necesite pueda tener un sistema así, ya sea en un hospital, como en su propia casa.

Mejorar el precio del sistema de monitorización del dolor, e intentar conseguir que sea lo más exacto posible, es uno de nuestros principales objetivos, y nuestra motivación, por supuesto, es mejorar al máximo la calidad de vida de las personas, que al final es la clave principal que nos impulsa a realizar este proyecto.

1.2- OBJETIVOS DEL PROYECTO:

Los objetivos del proyecto son las metas que se pretenden conseguir en base a los conocimientos adquiridos a lo largo del desarrollo de este. A continuación, se presentan los dos tipos de objetivos presentes en este proyecto.

1.2.1- OBJETIVOS DOCENTES:

Por un lado, aparece este tipo de objetivos que entran dentro de lo que se espera conseguir, de manera crítica, resolutive y comunicativa, a lo largo de la realización de este proyecto. A continuación, se muestra una lista con estas metas:

- Conocer la estructura del microcontrolador Arduino UNO y sus componentes, así como el funcionamiento informático de este.
- Definición de los requisitos fundamentales que componen el sistema.
- Capacidad de selección y decisión de los elementos que van a formar parte del sistema, refiriéndose así al microcontrolador, sensores y componentes que en su totalidad dan lugar a un sistema de adquisición y procesamiento de señales biomédicas.
- Elaborar un circuito utilizando los elementos seleccionados, así como diagramas que muestren el funcionamiento de este sistema.
- Capacidad de programación del sistema, en lenguaje C ++, para que sea capaz de recoger señales y transformarlas a las unidades correspondientes en función del sensor que las obtenga y, a partir de ahí, realizar unas acciones concretas, todo ello desde el entorno de desarrollo de Arduino.
- Obtención de un conjunto de entrenamiento a partir los datos obtenidos del sistema en un formato concreto (.csv).
- Aplicar técnicas de análisis y preprocesamiento de los datos obtenidos en el conjunto de prueba.
- Capacidad de estudio y contraste de los datos obtenidos a partir de la aplicación de diferentes modelos sobre el conjunto de datos, comparando las ventajas y desventajas que presentan, a través de gráficos, tablas, matrices y datos estadísticos.
- Capacidad de decisión y justificación de los modelos más adecuados para la clasificación del nivel de dolor a partir de las variables biomédicas obtenidas a través del sistema, en función de los datos de precisión y error, entre otros datos proporcionados por dichos algoritmos.

- Capacidad de planificación temporal y financiera aplicada al proyecto.
- Aprender a elaborar una memoria con fines de investigación siguiendo la normativa vigente y cumpliendo los estándares de estilos definidos.

1.2.2- OBJETIVOS TÉCNICOS:

Por otro lado, los objetivos técnicos definen la finalidad de este proyecto a nivel de tecnológico y de herramientas que permitan conseguir el objetivo principal del proyecto.

A continuación, se muestra una lista con estas metas:

- Habilidad para el diseño y construcción de un sistema de bioinstrumentación utilizando elementos básicos y asequibles, tales como sensores, microcontrolador Arduino UNO, componentes analógicos y digitales, etc.
- Programación de variables biomédicas, tales como frecuencia cardíaca, frecuencia respiratoria, respuesta galvánica de la piel y temperatura corporal, a partir de los datos obtenidos por el sistema biomédico dentro del lenguaje de programación de Arduino.
- Capacidad de aprendizaje y empleo de técnicas que se ajusten al rango del problema y al entorno de desarrollo de Python, tales como: pandas, numpy, TensorFlow, Keras o matplotlib, entre otras librerías que permitan desarrollar la aplicación de algoritmos de IA al proyecto, prevaleciendo los resultados con mayor precisión posible.
- Análisis de datos, obtenidos al completar el estudio, representados a través de gráficos, tablas y matrices, entre otros, que permitan la definición de los resultados y conclusiones del proyecto.
- Implantación del sistema en una interfaz física que permita obtener los datos biomédicos, de personas seleccionadas para un estudio, de manera sencilla, fiable y ergonómica.

En resumen, el objetivo principal de este proyecto es la consecución de un sistema de monitorización del nivel de dolor en función de los datos proporcionados por las señales fisiológicas adquiridas y definidas dentro de modelos de Inteligencia Artificial, tal y como recalcamos al principio, con todos los aprendizajes y metas que el desarrollo de este motivo conlleva, tanto a nivel de docencia como a nivel técnico.

1.3- ESTADO DEL ARTE:

Tras numerosas búsquedas en el mercado, hemos identificado una serie de productos e ideas que comparten similitudes con el sistema que pretendemos diseñar en este proyecto. A continuación, se presenta una descripción de estos, así como las ventajas y desventajas que presentan en comparación con el cumplimiento de algunos de los requisitos que componen nuestro sistema.

1.3.1- PMD_200

En primer lugar, aparece el único mecanismo automatizado que permite monitorizar el estado de dolor de un paciente disponible en el mercado. Estamos hablando del dispositivo “PMD-200” [3]. Este sistema es un monitor de nocicepción no invasivo que permite representar la respuesta fisiológica de los pacientes en función de los estímulos dolorosos a través de un monitor. Este aparato ha sido diseñado por la empresa MedaSense y presenta la única solución inteligente para la evaluación del dolor en el mercado hoy en día.

El objetivo principal de este instrumento consiste en la evaluación del nivel de dolor que padecen los pacientes que están sometidos a la anestesia general, en una intervención quirúrgica o en una unidad de cuidados intensivos, y no pueden transmitirlo a los profesionales sanitarios. Esto permite que la monitorización del paciente anestesiado sea individual y eficaz, consiguiendo una evaluación nociceptiva importante que produce la disminución tanto de opioides (fentanilo, remifentanilo, morfina, propofol) hasta un 30% [4], como el riesgo de sufrir complicaciones durante la intervención para que los pacientes se puedan despertar de la anestesia con el nivel de dolor postoperatorio, relacionado con un exceso o una insuficiencia de administración de opioides [5], más bajo posible. Además, consigue reducir considerablemente los eventos adversos durante el tratamiento, tanto quirúrgico como farmacéutico, permitiendo una reducción considerable de la estancia hospitalaria del paciente y un ahorro hospitalario de, aproximadamente \$325 980 (304243,65 EUR) [6], económicamente hablando.

Antiguamente, se llevaba a cabo la monitorización nociceptiva del paciente a través de la frecuencia cardíaca y la presión arterial, entre otros. Sin embargo, algunos estudios han manifestado que estas constantes vitales no son suficientes para llevar un control del nivel del dolor que sufre un paciente anestesiado. Por ello, el sistema está diseñado en base al índice de nivel de nocicepción (NOL) que permite clasificar el dolor del paciente en un umbral variable de 0 a 100 mediante una serie de algoritmos

fundamentados en Inteligencia Artificial. Este umbral se calcula a través de una correlación entre los datos obtenidos a través de unos sensores que actúan como biopotenciales y la puntuación clínica basada en la estimación de la concentración de opioides y la capacidad del estímulo nociceptivo.

Para obtener los datos de los sensores, el sistema se compone de un aparato con forma de pinza, similar al pulsioxímetro, formado por cuatro sensores que permiten obtener variables fisiológicas de manera continua y no invasiva: temperatura, fotopletimografía, respuesta galvánica de la piel y acelerómetro. Estos datos serán los que principalmente permitan cuantificar dichas señales dentro del umbral del NOL. Esta escala aparece, de forma gráfica y numérica, en un monitor que permite al profesional sanitario obtener información sobre el estado de nocicepción del paciente de la siguiente manera:

Índice NOL = 0, ausencia de nocicepción (el paciente no padece dolor).

Índice NOL = 28, nivel de nocicepción calibrado al individuo.

Índice NOL = 100, respuesta nociceptiva extrema (el paciente padece dolor extremo).

En un pequeño resumen, podemos afirmar que los estudios realizados para la realización de este proyecto recalcan que, además de la frecuencia cardíaca y la presión arterial, es importante obtener más información sobre distintas señales fisiológicas para poder clasificar el nivel de dolor de forma más precisa. Por ello, podemos aprovechar este aporte de conocimiento para poder seleccionar una mayor cantidad de sensores que permitan obtener los datos que hagan que la herramienta sea lo más completa y eficiente posible, teniendo en cuenta lo económico, sin dudar.



Figura 1: Dispositivo PMD_200

A nivel funcional, aparece la capacidad de monitorizar el dolor a través de aplicaciones disponibles en dispositivos portátiles (smartphones y tablets) en cualquier lugar en el

que se encuentre el paciente, siempre y cuando esté conectado a la red wifi. Lo más normal es que sea el paciente el que administre datos a estos programas que irán realizando un estudio en función de los parámetros que se hayan definido en su diseño en lugar de que la aplicación sea la que capte dicha información.

Según un estudio, hay más de doscientas ochenta aplicaciones disponibles en las tiendas oficiales de los principales sistemas operativos (Android/iPhone) que permiten monitorizar y evaluar el estado nociceptivo del paciente [7]. Sin embargo, son muy escasos los programas que están desarrollados con el contraste y el análisis científico y mucho menos los que hayan sido sometidos a ciclos de pruebas que cumplan las exigencias planteadas en los certificados de calidad que rigen este tipo de software. Es interesante conocer que dos de las aplicaciones cuyos desarrollos están contrastados empíricamente están programadas en España. A continuación, se muestran, de manera resumida, las características de estos programas que permiten llevar un manejo de la enfermedad.

1.3.2- MONITOR DE DOLOR 2.0

Investigadores de la Universidad Jaume I, la Universidad de Barcelona y el Instituto de Salud Carlos III desarrollaron un software que permite registrar toda la información en la vida diaria de las personas que padecen esta dolencia monitorizando el nivel de dolor que sufren y la eficacia del tratamiento proporcionado por el profesional sanitario, y, afirman que *“con esta app, el paciente estará monitorizado durante el tiempo que transcurre entre consultas y esto permite a los facultativos conocer si el tratamiento funciona o si hay que realizar algún ajuste o intervención”* [8].

El funcionamiento principal de esta aplicación consiste en un estudio anónimo de un conjunto de datos que el paciente debe ir proporcionando a lo largo de un mes lunar.

En otras palabras, el programa realizará un breve cuestionario, de unos 2 minutos, cada mañana, tarde y noche. Las preguntas del cuestionario están relacionadas con el nivel del dolor del paciente, el estado de ánimo y el cansancio. La interfaz es muy simple y se podría decir que su funcionalidad principal es que sea una herramienta que permita al médico conocer el estado del paciente de forma remota y sin necesidad de que este asista a la consulta brindando así una *atención integral* [9].



Figura 2: Interfaz de la aplicación Monitor de dolor 2.0

Las instrucciones que se deben cumplir para sacar el máximo partido al programa consisten en responder los cuestionarios como mínimo 2 veces al día y registrar el nivel del dolor cuando el paciente esté sufriendo un episodio. Sin embargo, esto puede suponer una forma de obtención de información muy poco fiable pues pueden surgir varias situaciones que impidan que el paciente pueda registrar los datos, como, por ejemplo, que al usuario se le olvide responder a los cuestionarios o tenga la sensación de que llevar a cabo ese tratamiento no tiene ningún tipo de respuesta ya que el feedback con el profesional y la aplicación es casi nulo, entre otras desventajas.

La principal ventaja es que esta aplicación es gratuita ya que el objetivo principal es la adquisición de datos para la investigación, por lo que, no supondría coste alguno para el paciente, aunque sí podría, para las personas que llevan a cabo el soporte del software que la respalda e investigan a partir de la información que se ha registrado. Cabe resaltar que está cobijada por el artículo 37 de la Ley española 22/1987 de Propiedad Intelectual y que puede ser utilizada desde cualquier plataforma. Para más información, acudir a la descripción del programa en la plataforma oficial de Android [10] o de iPhone [11].

1.3.3- ALGÓMETRO DIGITAL PALPATRONIC

Actualmente, a nivel comercial hay una gran cantidad de sistemas que cumplen la función de cuantificar el nivel nociceptivo que padece un paciente. Estamos hablando de los algómetros, o también comúnmente conocidos como dolorímetros.

Estos dispositivos son estructuras hardware compuestas por algún tipo de sensor que recoge bioseñales, las procesa y les da un determinado valor en función del dolor. Sin

embargo, el principal problema es que su arquitectura informática es inmodificable (cerrada), es decir, es solamente accesible por los desarrolladores, y, además, la gran mayoría están basados en células de carga, lo que produce que su valor económico sea muy alto [12].

En resumen, podemos decir que son sistemas que permiten localizar la ubicación del dolor y, de alguna manera, darle un valor, pero no analizan estos datos para su posterior estudio o clasificación. A continuación, se muestra una breve descripción de alguno de los sistemas que hemos pensado que serían más importantes para comentar en este estudio.

En el sentido digital, aparece este aparato diseñado y programado por palpatronic como instrumento de terapia física que muestra la fuerza que se ejerce en los puntos de gatillo. Los puntos de gatillo son las zonas musculares contraídas que producen nivel nociceptivo a través de un impulso nervioso transmitido por el Sistema Nervioso Central hacia otras regiones anatómicas diferentes.

Los componentes fundamentales de este sistema son:

- El medidor de fuerza que permite localizar la zona en la que se encuentra la dolencia.
- La batería con capacidad para poder realizar un uso de manera portátil.
- Luces LED cuyo encendido muestra un valor que corresponde a la fuerza aplicada en el tejido correspondiente.
- Sonidos acústicos editables que indican la duración del tratamiento.
- El microcontrolador, programado para cumplir con todas estas funciones.

A parte de estas principales características, su valor en el mercado, aproximadamente 180 EUR [13], en función del proveedor, genera valor añadido a este instrumento de monitorización sensorial.

Como hemos visto, hay una gran variedad de dispositivos que cumplen la función de monitorizar el nivel de dolor que sufre un paciente de diferentes maneras, ya sea de forma analógica, ya sea de forma digital o bien, a través de una aplicación. Aun así, se siguen realizando estudios que permitan mejorar y complementar los instrumentos disponibles hoy en día.

Por ello, es importante señalar algunos estudios relacionados con el diseño de dispositivos similares a los descritos anteriormente, realizando una pequeña investigación sobre algunas patentes.

1.3.4- PATENTE: ES 2 170 014 B1

La herramienta desarrollada en esta invención permite medir el umbral de dolor aplicando un sensor de presión sobre un punto de la piel de forma incremental mientras un contador va registrando el valor de la fuerza aplicada, en gramos, de manera directa, a lo largo de cada segundo y que será representada en una pantalla contenida dentro del mismo instrumento. La principal ventaja de este instrumento es que no es necesario que esté conectado a un ordenador, lo que permite que el transporte sea más sencillo y eficaz para el profesional en cualquiera de las áreas sanitarias que componen un hospital [14].

Su diseño está principalmente orientado a solventar el problema del traslado de aparatos debido a su reducido tamaño y su ergonomía, permitiendo así que el aparato se pueda utilizar en varias zonas hospitalarias, e incluso en el domicilio del paciente, si fuese necesario. Para ello, dispone de un módulo de presión, compuesto por un sensor de presión, valga la redundancia, un tubo con una almohadilla que se colocará en la zona a tratar y un módulo de cuantificación que se encargará de registrar la información obtenida y darle un valor para poder representarlo.

A continuación, presentamos una breve tabla comparativa de los instrumentos comentados anteriormente para poder realizar un mejor análisis de los instrumentos disponibles en el mercado.

Dispositivo	Precio	Ventajas
PMD-200	N/S	Obtención, monitorización y clasificación de datos efectivas debido a la tecnología NOL.
Monitor de dolor 2.0	Gratis	Cuestionarios para monitorizar el estado del paciente. Disponible en todas las plataformas digitales.

Algómetro Palpatronic	179,00	Cálculo de la fuerza aplicada en la zona nociceptiva de forma digital.
ES2170014 B1	N/S	Cálculo del nivel de dolor en un punto para obtener el umbral de dolor.

Cuadro 1: Comparación de los dispositivos analizados en el estado del arte

Tras este análisis mercantil, podemos concluir que actualmente hay varias herramientas que permiten cubrir algunos de los objetivos principales de este proyecto. Sin embargo, solamente hay un dispositivo capaz de cumplir con las expectativas que se proponen, ese dispositivo es el pmd-200 pues es el único instrumento que permite monitorizar el nivel nociceptivo y clasificarlo en un parámetro concreto en función de los datos obtenidos, en este caso utilizando la tecnología NOL.

Esto no quiere decir que el resto de los dispositivos sean de menor importancia pues, cada uno de ellos cumple con una característica individual concreta.

Por ejemplo, el cálculo de secuencias y el nivel del dolor en función aplicada, utilizando los dispositivos definidos en las patentes, permiten la localización de la dolencia, además, la precisión de los algómetros, permite que la obtención de bioparámetros sea lo más fiable y precisa posible, y, la evaluación del nivel del dolor de un paciente a través del feedback que este proporciona, a través de cuestionarios o escalas de medición del nivel nociceptivo, hacen que estos sistemas puedan formar parte de la construcción del tratamiento más correcto posible para un paciente que padezca esta enfermedad.

Con todo ello, podemos afirmar que una combinación de estas funcionalidades sumando un coste que permita que la herramienta sea accesible económicamente a todas las ramas sanitarias conocidas, pueden dar lugar a un producto completo con una alta capacidad de proyección en el mercado. Este análisis nos puede servir para obtener una buena redacción a la hora de llevar a cabo la definición de requisitos necesarios para nuestra herramienta.

1.4- ELICITACIÓN DE REQUISITOS Y ANÁLISIS DE RIESGOS

Siguiendo lo comentado en el apartado anterior, además de entrevista con el tutor del proyecto en simulación de un cliente real, sumado a los objetivos definidos al inicio del proyecto, podemos dar lugar a una definición de requisitos que se adecuen lo máximo posible a las necesidades de nuestro sistema. Para ello, se ha elaborado una lista por cada uno de los tipos de requisitos fundamentales que componen el sistema: funcionales, no funcionales y de información.

1.4.1- REQUISITOS FUNCIONALES

Estos requisitos funcionales describen las acciones y capacidades necesarias para alcanzar los objetivos docentes y técnicos establecidos en el proyecto. En otras palabras, este tipo de requisitos describen cómo se comporta la solución a nivel técnico. Los requisitos funcionales del sistema son los siguientes:

Requisito	Descripción
RF_01	El sistema debe estar compuesto por elementos (el microcontrolador, sensores y componentes necesarios) que permitan la adquisición y el procesamiento de señales biomédicas.
RF_02	El sistema debe transformar las señales obtenidas para establecerlas en su escala y unidades correspondientes.
RF_03	El entorno de programación del sistema será en Arduino, cuyo lenguaje de programación en C++.
RF_04	Se podrán almacenar los datos obtenidos en un fichero (formato .csv) a través de una aplicación elaborada desde el entorno de desarrollo de aplicaciones: WindowsForms.
RF_05	Se podrá definir un conjunto de entrenamiento válido utilizando el fichero .csv obtenido a través de la aplicación.
RF_06	El sistema debe ser capaz de mostrar los resultados de la clasificación del nivel de dolor, así como las variables que están indicando el porqué de esa clasificación a través de una aplicación disponible para smartphones y PC.

1.4.2- REQUISITOS NO FUNCIONALES

Este tipo de requisitos describen características fundamentales para su correcto funcionamiento, así como el establecimiento de la calidad del sistema. Los requisitos no funcionales del sistema son los siguientes.

Requisitos no funcionales	Descripción
RNF_01	Eficiencia: El sistema debe ser capaz de adquirir y procesar las señales biomédicas de manera eficiente asegurando un tiempo de respuesta adecuado de 2 segundos entre envíos de datos.
RNF_02	Rendimiento: El sistema debe ser capaz de ser óptimo a nivel de rendimiento. Es decir, que sea capaz procesar y analizar los datos biomédicos en tiempo real.
RNF_03	Fiabilidad: El sistema debe adquirir la máxima confianza y precisión, para ello, se debe evitar el máximo número de errores y fallos en la adquisición, procesamiento y análisis de los datos biomédicos.
RNF_04	Usabilidad: El sistema debe ser fácil de utilizar tanto a la hora de la recogida de datos, como la representación de la clasificación de estos.
RNF_05	Sencillez: El sistema debe ser entendible por el usuario. Para ello, las interfaces deben ser sencillas y visualmente entendibles a simple vista.
RNF_06	Mantenimiento: El sistema debe presentar un desarrollo que permita un fácil mantenimiento y actualización, de cara a mejoras futuras.
RNF_07	Escalabilidad: El sistema debe ser capaz de procesar volúmenes de datos de tamaño medio.
RNF_08	Portabilidad: El sistema debe ser utilizado en diferentes entornos de programación (Arduino, Visual Studio y App Inventor), permitiendo así, su implementación en contextos diferentes: lectura y recogida de datos, envío de los resultados de la clasificación, etc.
RNF_09	Lógica de datos: El sistema debe definir una lógica de datos similar al algoritmo escogido en el estudio.

RNF_10	Seguridad: El sistema debe garantizar la LPGD de cada persona que haga uso de este.
--------	---

1.4.3- REQUISITOS DE INFORMACIÓN

Este tipo de requisitos representan qué datos necesita el sistema y qué datos va a generar. Permiten obtener conclusiones válidas y significativas a partir de ellos.

Requisitos de información	Descripción
RI_01	Recopilación de datos biomédicos: El sistema debe ser capaz de recopilar datos biomédicos que proporcionen información sobre el estado de algún parámetro del paciente.
RI_02	Transformación de datos biomédicos: El sistema debe ser capaz de convertir los datos biomédicos recopilados en variables precisas con sus respectivas unidades del SI correspondientes. Estas variables deben ser: frecuencia cardíaca, frecuencia respiratoria, respuesta galvánica de la piel y temperatura corporal.
RI_03	Almacenamiento de datos: El sistema debe ser capaz de integrarse con una aplicación que permita almacenar los datos biomédicos recopilados y transformados en un formato compatible con la definición de un conjunto de entrenamiento para IA. Dicho formato corresponde con el Comma-Separated-Values (csv).
RI_04	Normativas y estándares: El sistema debe cumplir con las normativas vigentes y los estándares establecidos para la investigación científica y el manejo de datos biomédicos. Esto incluye aspectos éticos, de privacidad y de seguridad de la información.

BLOQUE 2: EJECUCIÓN DEL PROYECTO

2.1- DISEÑO DEL SISTEMA:

En general, el sistema está compuesto por dos grandes bloques, que a su vez componen varios subbloques, dando lugar a la representación de todas las estaciones de trabajo que pueden ser llevadas a cabo en él y, además, cumpliendo los requisitos que hemos definido anteriormente.

En este apartado, se irá comentando cada bloque que compone el sistema en función de la información que se puede visualizar en el diagrama de bloques.

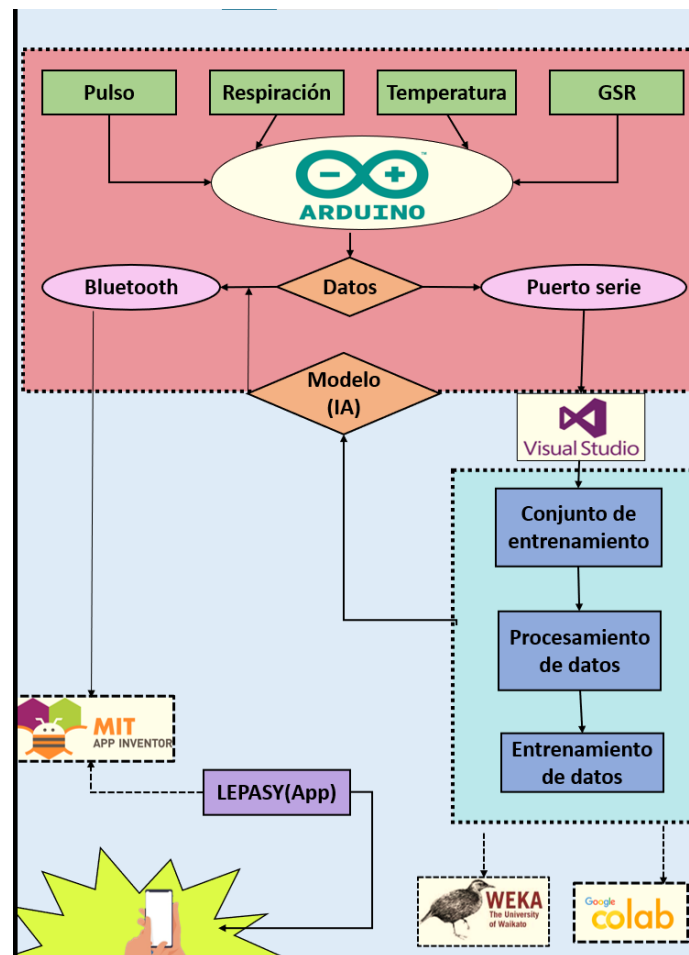


Figura 3: Diagrama de bloques del sistema.

Nota. Diagrama realizado por elaboración propia.

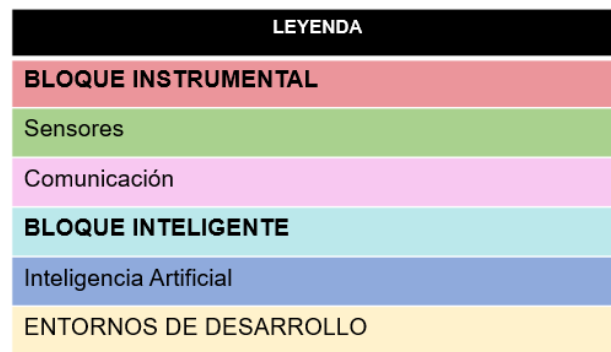


Figura 4: Leyenda del diagrama de bloques.

Tal y como se observa en la figura anterior, los dos bloques principales que componen el sistema son los definidos como bloque instrumental y bloque inteligente.

El primero de ellos es el bloque instrumental. Este conjunto de elementos está compuesto por la suma de todos los componentes y funciones necesarias que permitan obtener las variables fisiológicas que definan el conjunto de entrenamiento del sistema. A continuación, iremos comentando cada uno de los componentes y funciones que han sido necesarios para conseguir el objetivo de este bloque.

El elemento principal del sistema es un conjunto de sensores que captan bioseñales fisiológicas que posteriormente serán enviadas en tiempo real a una placa de Arduino UNO para ser leídas, procesadas y transformadas, en el entorno de programación C++, pasando de magnitudes fisicoquímicas a magnitudes eléctricas, y, todo ello en función del fin en concreto de cada componente.

Para ello, hemos hecho empleo de un total de cuatro sensores programados para captar información del paciente en tiempo real. Este conjunto de componentes está formado por un sensor de pulso, un micrófono, un sensor de respuesta galvánica y otro de temperatura que permiten obtener la frecuencia cardíaca (BPM), frecuencia respiratoria (RPM), respuesta galvánica de la piel (GSR) y temperatura corporal (°C), respectivamente, en sus unidades concretas. Cada uno de estos sensores están directamente conectados en su lugar correspondiente de la placa de Arduino UNO, permitiendo así captar las señales y poder procesarlas en función de unas librerías y funciones concretas implementadas en el entorno de desarrollo de Arduino, cuyo lenguaje de programación es en C++, obteniendo como resultado las variables fisiológicas que necesitamos en tiempo real.

Esta información será enviada por el puerto serie en un formato concreto, que será reconocida por una aplicación de interfaz sencilla diseñada a partir del lenguaje de

programación C# cuyo entorno de programación permite establecer una comunicación bidireccional entre Arduino y Visual Studio necesaria para la definición de un conjunto de entrenamiento en un formato estándar (.csv). Este conjunto de entrenamiento nos permitirá recoger información de los parámetros biomédicos de serie de usuarios que han formado parte de un ensayo clínico para poder realizar el estudio.

El objetivo de la obtención de este fichero de datos separado por comas es definirlo como conjunto de entrenamiento para combinar algoritmos de inteligencia artificial que nos permitan obtener parámetros e información para realizar un pequeño estudio sobre cómo clasificar la frecuencia cardíaca, respiratoria, respuesta galvánica y temperatura corporal en nivel de dolor alto, medio o bajo. Por ello, se enviará el conjunto a dos entornos de programación que permiten realizar este estudio, Google Colab y Weka.

2.2. IMPLEMENTACIÓN:

En este apartado mostramos los procedimientos que han ido dando forma a la solución propuesta en este proyecto en función de las tecnologías y la metodología utilizada y que veremos a continuación.

2.2.1- METODOLOGÍA:

Cuando se está desarrollando un proyecto software de alto nivel, es muy importante seleccionar la metodología que se va a seguir a lo largo del proyecto. Nosotros hemos decidido utilizar Scrum, ya que es muy común en este tipo de proyectos.

Scrum es un conjunto de buenas prácticas para trabajar en un equipo multidisciplinar. <https://proyectosagiles.org/que-es-scrum/> En nuestro proyecto, esta metodología está implementada en la parte de desarrollo, haciéndolo valioso desde sus primeras versiones.

Los equipos trabajan con ciclos de ejecución llamados “sprints” que permiten las versiones por las que se construye el proyecto con el fin de lograr un sistema completamente desarrollado de manera tan rápida y continua.

La metodología Scrum define diferentes roles asociados a cada uno de los componentes del equipo multidisciplinar:

- Scrum Master: es el equivalente a lo que comúnmente se conoce como jefe de proyecto. Su labor es definir y analizar cómo va a trabajar el equipo multifuncional, así como de ser el responsable de conseguir que se sigan los valores y las prácticas de ‘scrum’. Ayuda a los miembros del equipo para que trabajen de forma autónoma y autoorganizada. Se ocupa también de eliminar problemas y obstáculos que puedan poner en riesgo el objetivo del ‘sprint’.
- Product owner: Su mirada está siempre puesta en el cliente, y en lo que el equipo va a desarrollar. Es responsable de que el producto vaya incrementando su valor con cada ‘sprint’. Además, es la persona encargada de marcar el objetivo de manera clara y acordada con el resto del equipo.
- Equipo de desarrollo: Es el grupo de profesionales que hace el trabajo necesario para poder entregar el incremento de valor en el producto. Se autoorganizan para realizar el trabajo y han de estar disponibles a tiempo completo en el proyecto.

El personal que compone este proyecto está formado por un scrum master, asignado al tutor del trabajo, y los alumnos que lo realizan, asignados como dos integrantes pertenecientes al equipo de desarrollo.

Cada uno de los miembros del equipo deben cumplir una serie de condiciones cuando estén ejerciendo su rol correspondiente:

1. No se deben realizar cambios que pongan en peligro el objetivo.
2. El 'Product owner' y el equipo de desarrollo trabajan conjuntamente ajustando el detalle de las funcionalidades planificadas para el 'sprint'.
3. Además de construir el producto, todo equipo trabaja conjuntamente como 'Product owner' y los miembros del equipo aclaran y negocian entre ellos a medida que se va elaborando el proyecto.

La duración máxima recomendada para un sprint será de un mes, porque si es más largo corremos el riesgo de aumentar la complejidad del producto, aumentando así el riesgo de no entregar lo que el cliente espera.

Una vez acabado el sprint será necesario llevar a cabo una labor de inspección y revisión del trabajo realizado, en la que haya un 'feedback' por parte del "Product owner" o incluso el propio cliente. La reunión debe cumplir una serie de condiciones:

1. Los asistentes son el equipo 'scrum' y otros interesados que puedan resultar clave.
2. Se identifica lo que se ha hecho y lo que no respecto a lo inicialmente acordado.
3. Se detectan los problemas y se analiza cómo se resolvieron.
4. Todo el equipo debe colaborar a la hora de decidir qué hacer a continuación.

Todas las metodologías ágiles buscan mejorar de manera continua la forma en la que el equipo se relaciona durante el proceso de desarrollo. En scrum existe otra sesión específicamente definida para lograrlo: la retrospectiva. Se trata de una oportunidad para que el equipo comparta sus impresiones y recomendaciones sobre la forma en la que han trabajado durante el 'sprint' que acaba de finalizar, con el objetivo de identificar las lecciones aprendidas.

2.2.2. TECNOLOGÍAS EMPLEADAS:

2.2.2.1- Bloque Instrumental:

- **Sensor de pulso:**

Este aparato es empleado para medir el ritmo cardiaco en tiempo real, y puede ser colocado tanto en la punta de los dedos como en el lóbulo de la oreja. Como ya sabemos, nuestro sistema irá colocado entre frente y nuca, por lo que se decidió ubicar este sensor en el lóbulo del oído derecho.

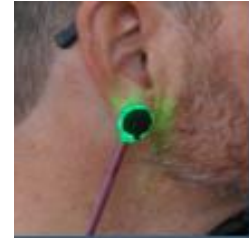


Figura 5: Sensor de pulso.

Como se puede sobreentender, es necesario que permanezca en contacto con la piel durante la lectura para que la señal pueda ser obtenida de manera correcta y todo ello sin ejercer mucha presión sobre la zona medida para que se puedan obtener los resultados más fiables posibles.

El circuito de este componente analógico está compuesto por un sensor de ritmo cardiaco óptico, componente que emite una luz de color verde que mide pequeños cambios de volumen en el flujo de la sangre que atraviesa la capa epidérmica de la piel en forma de señal analógica de voltaje, dando lugar a la onda que grafica el fotopletismograma, una etapa de amplificación y normalización que sitúa la señal alrededor del punto medio en voltaje, y un filtro para el ruido, lo que hace que la señal de salida sea confiable y estable.

- **Sensor de frecuencia respiratoria (micrófono):**

Un micrófono es un transductor que convierte las ondas sonoras en señales eléctricas por lo que se suele utilizar para detectar sonidos.

La salida producida por un micrófono es una señal eléctrica analógica que representa el sonido recibido. Sin embargo, en general, esta señal es demasiado baja para ser medida y tiene que ser amplificada.

En este sistema se ha implementado el componente KY-038 que contiene un micrófono que permite obtener la lectura tanto en valor digital como en analógico.

Este tipo de sensores pueden ser útiles por ejemplo para encender una luz al realizar una palmada. En nuestro caso, nos permitido medir las respiraciones por segundo de los pacientes, ya que al introducirlo en una mascarilla al estar completamente en silencio

lo único que detectará es la respiración, pudiendo así determinar cuánto se acelera la misma en una situación en la que las personas sufran dolor.



Figura 6: Micrófono KY-038

- **Sensor de temperatura corporal:**

El MLX90614 es un sensor de temperatura infrarrojo sin contacto que permite medir la temperatura de un objeto a distancia. La comunicación de este componente se realiza a través de pin analógico que permite la sencilla conexión con Arduino. Este tipo de termómetros infrarrojos tiene un gran número de aplicaciones, incluyendo sistemas de control de temperatura en edificios, control industrial de temperatura, detección de movimiento y aplicaciones de salud. En nuestro proyecto constituye un componente primordial pues nos permite obtener la temperatura corporal como variable fisiológica a medir.

Según la ley de Stefan-Boltzmann, todo objeto por encima del cero absoluto ($^{\circ}\text{K}$) emite radiación cuyo espectro es proporcional a su temperatura. El MLX90614 recoge esta radiación y su salida es una señal eléctrica proporcional a la temperatura de todos los objetos en su campo de visión. Internamente el MLX90614 está constituido con un chip de silicio con una fina membrana micromecanizada sensible a la radiación infrarroja, junto con la electrónica necesaria para amplificar y digitalizar la señal y calcular la temperatura. El conjunto incluye un amplificador de bajo ruido, un conversor ADC de 17 bits, un DSP (procesador digital de señal) y compensación de la temperatura ambiente.

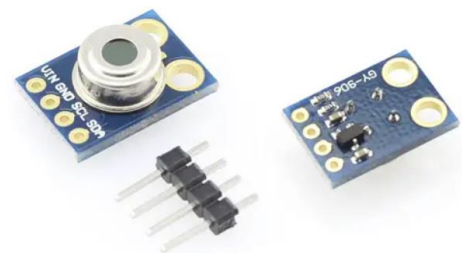


Figura 7: Sensor de temperatura corporal MLX90614

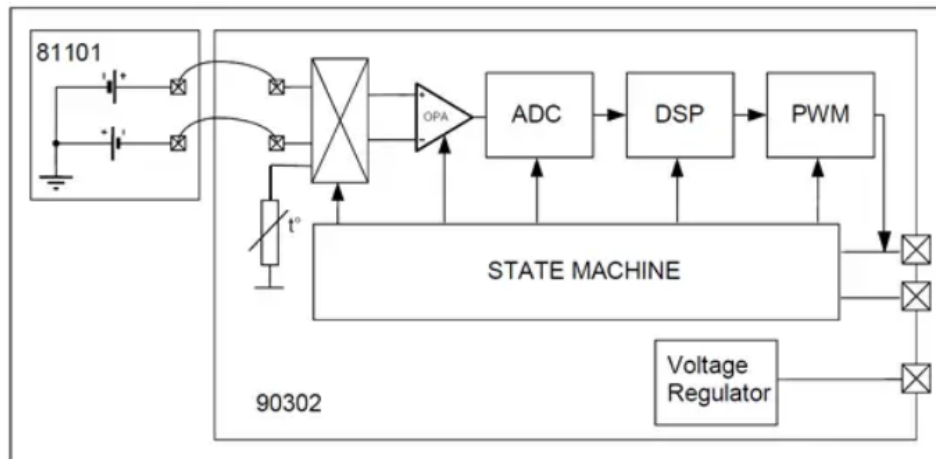


Figura 8: Diagrama de bloques del sensor MLX90614

El datasheet del elemento MLX90614 muestra que el rango de temperaturas que pueden ser medidas es muy alto pues oscila desde los -40°C hasta los 85°C para la temperatura ambiente y desde -70°C hasta 382°C para la temperatura de objetos. La precisión estándar es de 0.5°C referente a la temperatura ambiente por lo que podemos afirmar que los resultados que obtiene este sensor son bastante fiables para ser un elemento de bajo costo, pero, como en el resto de los sensores, debemos tener cuidado a la hora de tomar datos que no representen valores reales.

- **Sensor de sudoración:**

La respuesta galvánica de la piel (GSR) es una variable teórica que mide la variación de la conductividad eléctrica de la piel. En este caso, utilizaremos el sensor de respuesta galvánica CJMCU-6701.

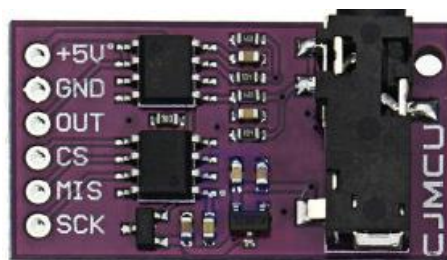


Figura 9: Sensor de respuesta galvánica CJMCU-6701

La teoría tradicional del análisis de la respuesta galvánica de la piel está basada en la suposición de que la resistencia de la piel varía con el estado de las glándulas sudoríparas de la piel. La sudoración del cuerpo humano está regulada por el Sistema Nervioso Autónomo (SNA). En particular, si la rama simpática (SNS) del Sistema Nervioso Autónomo está muy excitada, la actividad de la glándula sudorípara también

aumenta, lo que a su vez aumenta la conductancia de la piel, y viceversa. De esta forma, la conductancia de la piel puede ser una medida de las respuestas del Sistema Nervioso Simpático humano. Tal sistema está directamente involucrado en la regulación del comportamiento emocional en los humanos. Otros estudios han destacado la relación entre la señal GSR y algunos estados mentales, como por ejemplo estrés y cansancio. La señal GSR es muy fácil de registrar: en general sólo son necesarios dos electrodos en alguna zona del brazo. La variación de una corriente aplicada de bajo voltaje entre los dos electrodos se utiliza como medida de la actividad electrodérmica (EDA).

- **Arduino:**

Es una plataforma de hardware y software libre que permite crear millones de proyectos. En la actualidad hay múltiples placas que permiten realizar operaciones similares ya que su estructura es muy parecida. En este proyecto enfocaremos el punto de atención a la placa de Arduino UNO cuya implementación será llevada a cabo dentro del sistema. Podríamos decir que es el componente más importante de todo el sistema pues la lógica de captura y envío de datos viene definida en su entorno de programación. Dicho entorno está basado presenta una arquitectura basada en el lenguaje de programación C++ [15]. <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-de-arduino-estructura-de-un-programa/>

Por un lado, a nivel de hardware, es de vital importancia conocer que Arduino UNO está formado principalmente en torno al microcontrolador ATmega 328. Este componente computacional está formado, a su vez, por:

- Procesador AVR basado en arquitectura Harvard, con memorias y buses separados para instrucciones y datos.
- Memorias para ejecutar programas (Flash), almacenar las variables definidas (SRAM) y para almacenar constantes de configuración (EPROM).
- Timer que ejecuta las instrucciones del microcontrolador a través de ciclos.
- Puertos GPIO (3) de 8 bits cada uno que hacen referencia a los pines de entrada/salida que se distribuyen entre analógicos y digitales, en función del elemento a conectar. LIBRO INB



Figura 10: Arduino.

Por otro lado, el aspecto software se refiere al programa que constituye el entorno de desarrollo de Arduino que permite programar la placa para darle todo tipo de utilidades. Para ello, se utiliza la plataforma Arduino IDE cuyo lenguaje de programación es una adaptación de C++ y cuya estructura para definir programas que interactúen con los circuitos de la placa está compuesta por dos funciones, las cuales no entraremos mucho a detalle:

Setup (): función que se ejecuta cada vez que el código programado en la plataforma se compila en la placa.

Loop (): como su propio nombre indica, es una función que se ejecuta en bucle, durante un cierto tiempo determinado, desde que se compila el código definido en la plataforma en la placa, hasta que se desconecta el puerto serie o se vuelve a compilar el proyecto.

El objetivo principal de esta herramienta en nuestro proyecto es elaborar dos subproyectos que permitan la medida de las variables fisiológicas: pulso, frecuencia respiratoria, respuesta galvánica y temperatura corporal y el envío en un formato estándar concreto para uno de los subproyectos y para el otro la lectura de las variables fisiológicas anteriores sumado a la lógica que proporcione la clasificación del dolor de dolor en función de dichas variables y el envío por bluetooth a una aplicación móvil que permita la visualización de los resultados.

Para poder desarrollar el código del proyecto que cumpla con los objetivos y requisitos definidos inicialmente, se ha acudido a la utilización de librerías de procesamiento y transformación de señales biomédicas a sus correspondientes valores y unidades normales.

A continuación, se comentan las librerías que hemos empleado para realizar el código de Arduino IDE:

PulseSensorPlayGround: es una colección de métodos de código abierto que permiten el cálculo de todo tipo de información relacionada con el pulso y la frecuencia cardíaca. <https://github.com/WorldFamousElectronics/PulseSensorPlayground>

Adafruit _MLX90614: es un conjunto de herramientas compatibles con todo tipo de arquitecturas Arduino que permiten obtener y procesar los valores analógicos del sensor infrarrojo de temperatura corporal y transformarlos a la escala y unidades correspondientes. <https://reference.arduino.cc/reference/en/libraries/adafruit-mlx90614-library/>

2.2.2.2- Bloque Inteligente:

En este bloque se emplea la tecnología que define la Inteligencia Artificial en nuestro sistema. Para ello, se utilizaron como herramientas diferentes entornos de desarrollo que nos llevaran a completar los requisitos del proyecto relativos a este bloque con el objetivo de la realización de un estudio a partir de las métricas y los resultados de la ejecución de distintos modelos de Inteligencia Artificial.

Una vez se haya completado esta parte del bloque inteligente, se da paso a la implementación del modelo seleccionado en el entorno de Arduino IDE. Para ello, a los componentes del bloque instrumental se les suma un módulo bluetooth que envíe datos a una aplicación. La lógica, básicamente es una función procesa los datos de la variable fisiológica a través de bloques condicionales y cálculo de un resultado correspondiente a la clasificación del nivel de dolor en función de las variables fisiológicas que es enviado a la aplicación con el mismo periodo que las variables fisiológicas.

La aplicación para dispositivos móviles, portátiles, tablets y ordenadores de sobremesa, programada en lenguaje de programación en bloques a través de APP Inventor, consiste en una interfaz compuesta por varias pantallas que muestra las variables fisiológicas, a través de etiquetas, y la clasificación del nivel de dolor a través de una barra que se sitúa debajo de unas imágenes, definidas en la escala EVA, que representan esta clasificación. Esta aplicación está diseñada para que se permita la conexión bluetooth entre el entorno de desarrollo en el que se produce la lectura de los datos y la lógica de clasificación (Arduino IDE) y la aplicación.



Figura 11: Aplicación LEPASY.

Nota. Figura realizada por elaboración propia.

2.2.3. DESARROLLO:

Como hemos comentado anteriormente, el sistema completo se compone fundamentalmente de dos bloques en los cuales se aplican diversas metodologías y funcionalidades que permitan cumplir los requisitos definidos. Por ello, vamos a definir una serie de etapas que dividen los procesos que componen el sistema en base a la metodología Scrum. Estas etapas se conocen como sprints.

La decisión de definir de esta manera consiste en que cada sprint está directamente relacionado con la cantidad de tecnologías y procedimientos que hay que implementar en el sistema. El objetivo de esta metodología es la construcción de versiones del sistema que vayan dando forma al proyecto hasta su consecución.

El primer sprint consiste en un estudio de los datos biomédicos que queremos medir, para saber qué es lo que se desea obtener y cómo. Por ello, en este primer sprint se realizará la adquisición de los datos y su transformación al formato deseado, con el objetivo de poder obtener un conjunto de datos apropiados. Este primer sprint es de vital importancia el estudio que se realice sobre los datos que se obtengan pues nos puede proporcionar una suma de mejores resultados aplicados a los procesos que se llevarán a cabo posteriormente.

En el segundo sprint se recibirán los resultados obtenidos en el primero con el objetivo de aplicar el conjunto de entrenamiento aplicado al Deep Learning para realizar un breve estudio sobre el comportamiento del conjunto de entrenamiento aplicado al Deep Learning.

El tercer sprint tendrá una estructura similar a la del sprint anterior, pero en este caso, se procesará el conjunto de entrenamiento obtenido en el primer sprint con el objetivo de aplicar los diferentes modelos de machine learning para obtener parámetros que nos permitan realizar un estudio, no sin antes pasar por varias técnicas de preprocesamiento que añadan valor añadido a los datos del conjunto. El resultado del estudio será la elección de un modelo en base a los parámetros obtenidos en la ejecución de los distintos algoritmos.

El cuarto y último sprint consistirá en la implementación del modelo obtenido como resultado del sprint anterior en una lógica que permita mostrar por pantalla la clasificación del nivel dolor de un paciente en función de los datos que proporciona el mismo en tiempo real.

El orden que se ha elegido está basado en las siguientes ideas:

- Alternar los bloques en los sprints de manera cruzada permiten que se pueda ir revisando y actualizando cada uno de los bloques en función de las necesidades que se vayan obteniendo.
- La decisión del método que implementará la lógica del sistema no será decidida hasta que se complete la ejecución de los algoritmos seleccionados para su estudio, permitiendo así obtener todos los resultados para que puedan ser analizados en forma de tablas, gráficos y matrices.
- La implementación del sistema estará completa cuando se realicen todos los sprints.

En este caso, centraremos en los sprints que hacen referencia al bloque instrumental del proyecto.

2.2.3.1-Sprint 1:

En este estadio inicial se programaron sensores de bajo coste para enviar información sobre las señales fisiológicas que captan al entorno de Arduino IDE donde son procesadas y transformadas, pasando de magnitud física a eléctrica dentro de sus respectivas escalas y unidades. Este paso es de vital importancia pues su desarrollo nos permitirá definir el conjunto de entrenamiento que utilizaremos para el estudio de

algoritmos de Inteligencia Artificial, con mayor o menor fiabilidad. Por ello, la cantidad de datos, la precisión de los sensores y el formato adecuado son características muy importantes que entran en juego en esta fase.

2.2.3.1.1- Análisis:

Hoy en día, la tecnología permite obtener una gran cantidad de información sobre lo que ocurre en el cuerpo humano en un instante concreto. Por ello, es muy importante saber qué tipo de variables fisiológicas son las más importantes a medir para alcanzar los objetivos del sistema. Es decir, conocer cuáles son los datos biomédicos más relevantes a la hora de clasificar el nivel de dolor en distintos niveles es la base del desarrollo del sistema.

Según un estudio, el sistema nervioso autónomo es el causante de la respuesta al estímulo que produce la sensación de dolor [16], por ello, los parámetros que pueden traducir la intensidad del dolor pueden ser las variables relacionadas con el flujo sanguíneo como la frecuencia cardíaca, la tensión arterial, así como variables eléctrico-resistivas de la piel como la respuesta galvánica y la conductancia de la piel, la sudación y el lagrimeo y la temperatura pueden aportar información sobre el estado del paciente cuando está siendo sometido a un estímulo nociceptivo como puede ser una actividad quirúrgica, un tratamiento de rehabilitación o una enfermedad crónica. De hecho, las variables volumétricas pulmonares como la capacidad respiratoria, así como la frecuencia cardíaca también mantienen gran relación con el índice de dolor, en este caso a nivel postestímulo, como puede ser un proceso posoperatorio o después de una actividad de gran esfuerzo. Otros parámetros indicadores pueden ser las alteraciones a nivel fisiológico (alteración de glucagón, catecolaminas, etc.) pero estas variables son muy difíciles de obtener en la actualidad.

Sin embargo, otros estudios indican que el nivel de dolor no está definido solamente por las señales fisiológicas de respuesta a dicho estímulo, sino que hay otras variables que entran en juego como son las psicológicas y las situacionales. Anatómicamente hablando, la corteza somatosensorial e insular, así como el cerebelo, ganglios y tálamo son los aparatos que transmiten y procesan la información neurológica del dolor [17]. En estas zonas también es donde se produce la activación sensitivo-cognitiva durante episodios de depresión y ansiedad o malos hábitos de vida saludable, lo que ha dado lugar al debate dolor-nocicepción en algunos otros estudios recientes que afirman que la experiencia dolorosa no tiene por qué ser producida por respuesta a un estímulo, sino

que también puede ser producida por otros factores psicosociales [18] como los descritos anteriormente.

Tras realizar un análisis informativo sobre esta información, se llegó a la conclusión de obviar los factores psicosociales debido a la relativa subjetividad que presentan este tipo de datos. Por tanto, se decidió centrar la visión del proyecto en los factores fisiológicos ya que son variables que se pueden medir a través de sensores y presentar valores cuantitativos que pueden presentar ciertos patrones a la hora de clasificar el nivel de dolor.

Como hemos comentado anteriormente, hay muchas variables relacionadas con la respuesta fisiológica que permiten dar datos que muestren una cierta correlación con el nivel nociceptivo, pero no todas son tan fáciles de medir ni tampoco muestran resultados significativos que presenten información sobre el nivel de dolor.

Por ello, se realizó una investigación de la literatura sobre las variables que encuentran mayor asociación con el nivel de dolor. Según varios estudios, las constantes vitales representan la relación más fiable para la estimación del nivel de dolor [19].

En concreto, la frecuencia respiratoria es el parámetro más predictivo nociceptivamente hablando, seguido de la frecuencia cardíaca [20]. Estas variables miden la cantidad de volumen que bombean los órganos respectivos en forma de respiraciones/pulsaciones por minutos (RPM/BPM). Por ello, se afirma que cuando se produce una señal de dolor a nivel fisiológico, el propio cuerpo presenta una respuesta frente a ese estímulo, provocando, un aumento del bombeo sanguíneo y la demanda de oxígeno, dando lugar a un aumento del pulso a 5-10 latidos por minuto más de lo normal [21] y a un aumento de la necesidad de oxígeno provocando valores por encima de las 25 respiraciones por minuto, respectivamente. Según indican otros estudios, esta respuesta fisiológica puede dar lugar a otra serie de mecanismo como el aumento de la temperatura corporal y la actividad electro-resistiva de la piel. El calor, forma parte del proceso contráctil del latido del corazón mostrando un aumento directamente proporcional de la temperatura corporal cuando la frecuencia cardíaca aumenta [22] debido al aumento de la permeabilidad iónica de la membrana muscular del corazón.

Este mecanismo eléctrico-fisiológico también aparece representado en las membranas de la piel. El mismo sistema que regula la respuesta estimulante del dolor, es el que lleva el control del estado de sudoración en la piel. La respuesta galvánica de la piel está basada en la teoría de la variación resistiva de la piel. Esta teoría se conoce como Actividad Electrodermica de la piel (EDA) y afirma que la actividad de las glándulas

sudoríparas aumenta de manera directamente proporcional a conductancia de la piel provocando la sudoración [23].

Estas variaciones se pueden medir aplicando una corriente de baja intensidad a través de herramientas especialmente diseñadas para ello [24].

En resumen, las variables más trascendentales para la clasificación del dolor son la frecuencia respiratoria y la respuesta cardíaca, en ese orden. En un principio se estimó contar solamente con estos dos componentes, pero se pensó que sumando estas variables a las de respuesta galvánica y la temperatura corporal, podemos definir el conjunto de entrenamiento más completo y presentando un alto grado de relación con la clasificación del nivel de dolor.

Variables fisiológicas	Unidades	Rango bajo	Rango normal	Rango alto
Frecuencia cardíaca	Pulsaciones por minuto (BPM)	<60	[60-100]	>100
Frecuencia respiratoria	Respiraciones por minuto (RPM)	<12	[12-20]	>25
Respuesta galvánica	Conductividad eléctrica en voltios (V)	[0.01-0.03]	[0.03-0.07]	[0.07-0.15]
Temperatura corporal	Grados centígrados (°C)	<35	[36.5-37.5]	>37.5

Cuadro 2: Rangos de las variables fisiológicas.

Los principales inconvenientes de estos datos consisten en que la información que proporcionan puede variar entre un paciente y otro pues las constantes vitales presentan mucha diferencia en función de la edad, el sexo, el peso e incluso de la enfermedad. También hay que tener en cuenta que el nivel de dolor es una variable subjetiva ya que es muy susceptible a la opinión o sensación de cada persona.

Además, estas variables deben ser integradas en nuestro sistema de alguna manera para que puedan ser representadas de manera analógica lo que puede introducir algunos valores medidos que estén fuera de lo normal.

2.2.3.1.2- Implementación:

Para realizar el circuito de adquisición, procesamiento y transformación de señales fisiológicas se necesitan los siguientes componentes: sensores, placa de Arduino, cableado, puerto serie, led RGB, resistencias (330 Ohmios), felpa de tela elástica, mascarilla.

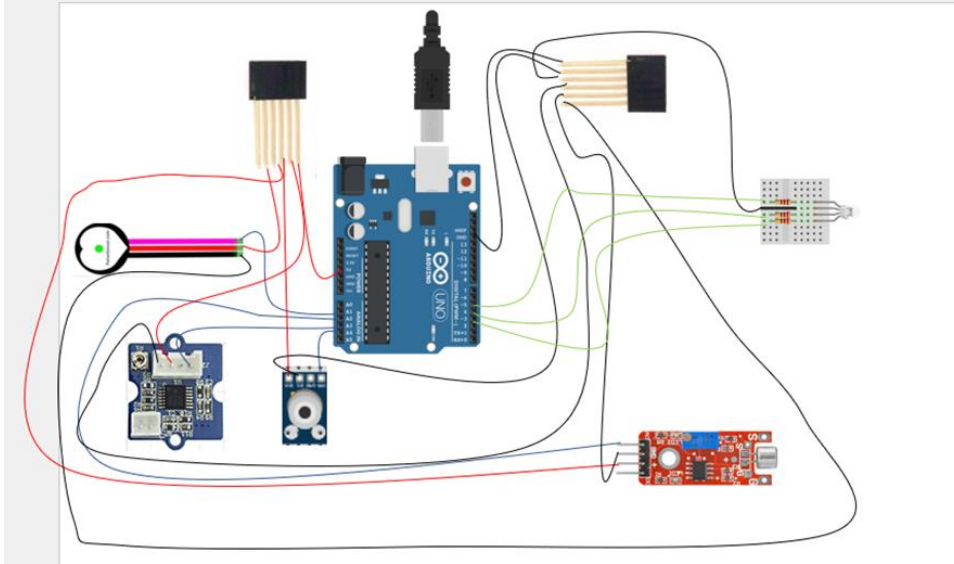


Figura 12: Circuito de adquisición de variables fisiológicas.

Nota. Circuito realizado por elaboración propia.

Los conectores de esta integración consisten en un conjunto de sensores conectados a la placa de Arduino UNO. Cada uno de ellos medirá una variable concreta, tal y como se indica a continuación:

Variable fisiológica	Sensor	Descripción	Librería/Funciones
Frecuencia cardíaca	KY-039	El sensor emite una luz infrarroja por el LED que atraviesa el lóbulo de la oreja derecha y se capta en el fototransistor, cambiando la resistencia	PulseSensor

Frecuencia respiratoria	KY-038 (W104)	El sensor es un transductor que detecta las señales sonoras y las transforma en señales eléctricas	Función de estimación
Respuesta galvánica	CJMCU-6701	Los electrodos miden la corriente eléctrica que circula a través de la superficie de la piel.	AnalogRead
Temperatura corporal	MLX90614	El sensor emite una luz infrarroja por el LED que choca la frente y es captada por un chip de silicio con una fina membrana micromecanizada sensible a la radiación	Adafruit_MLX90614

Cuadro 3: Integración de los datos adquiridos por los sensores.

Como podemos observar, durante la programación de algunos sensores se han utilizado librerías como base para la transformación de las señales. En otros casos, se han utilizado la definición de las funciones de estimación y la simple lectura de los valores. Cabe resaltar que todos los sensores utilizados son de tipo analógico lo que significa que la información de las señales recibidas llega de manera continua.

Durante el desarrollo del proyecto se ha ido programando un sketch por cada sensor, para poder procesar la información de manera detenida. Una vez se confirma el correcto funcionamiento de cada uno de los componentes, se han unido en un mismo sketch, cuyo diagrama de clases es el siguiente:

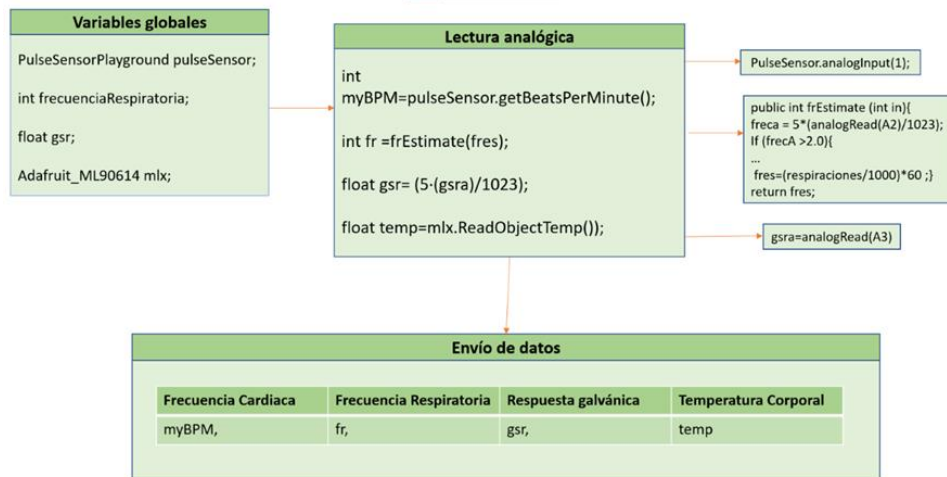


Figura 13: Diagrama de clases del circuito de adquisición de variables fisiológicas.

Nota. Diagrama realizado por elaboración propia.

Como podemos observar, el proceso está compuesto por las siguientes fases:

Variables globales: en esta parte de la programación, se lleva a cabo la definición de la nomenclatura y el tipo de las variables que serán utilizadas a lo largo de la ejecución.

Lectura analógica y procesamiento: en esta fase se produce la lectura de las variables fisiológicas para su posterior información, así como la transformación a la escala y las unidades correspondientes. Como todos los sensores son analógicos, se ha acudido a la función `analogRead ()` para obtener la información que miden.

Para obtener la frecuencia cardíaca se ha acudido a una librería conocido como `PulseSensorPlayground` como base para obtener el valor analógico del sensor y pulsaciones por minuto. Lo primero que se debe realizar consiste en incluir la librería y definir las variables globales necesarias. Después, se inicializa el umbral (`setThreshold`), pin analógico correspondiente de lectura (`analogInput`), así como la función que realiza la transformación completa (`getBeatsPerMinutes`). Esta función calcula las pulsaciones por minuto, en base al umbral, para ser almacenadas en la variable `myBPM`.

En cambio, la frecuencia respiratoria no ha implicado la adición de una librería pues no existe una que permita calcular el número de respiraciones por minuto utilizando un micrófono como sensor. Por ello, se decidió programar una función (`frEstimate`) que recibiera el valor obtenido por el micrófono no sin antes pasarlo del analógico al rango de voltaje. Dentro de esta función se calcula el número de respiraciones siempre que la señal recibida supere un umbral establecido tras varias pruebas del funcionamiento del código. Una vez se almacenen el número de respiraciones, se calculan cuantas

respiraciones equivalen por minuto. Para proporcionar un feedback de manera visual de este procedimiento, el sistema también contiene un led RGB, conectado a tres resistencias de 330 Ohmios, cada una a su patillaje correspondiente. Este elemento proporciona un parpadeo cada vez que se detecta una respiración correcta.

Para calcular la respuesta galvánica simplemente se ha realizado la transformación de la señal analógica recibida al rango de voltaje estándar debido a que el objetivo del sensor consiste en medir la actividad eléctrica de la piel.

Sin embargo, para calcular la temperatura corporal, también ha sido necesaria la incorporación de una librería, en concreto, Adafruit_MLX90614. Esta librería captura los datos del pin analógico en el que está conectado el sensor de temperatura corporal para estandarizar los valores leídos y obtener la temperatura ambiente en grados Celsius.

En este punto ya se han obtenido las variables fisiológicas en su unidad y escala correspondientes y se han unido en un mismo programa dentro del entorno de programación de Arduino IDE.

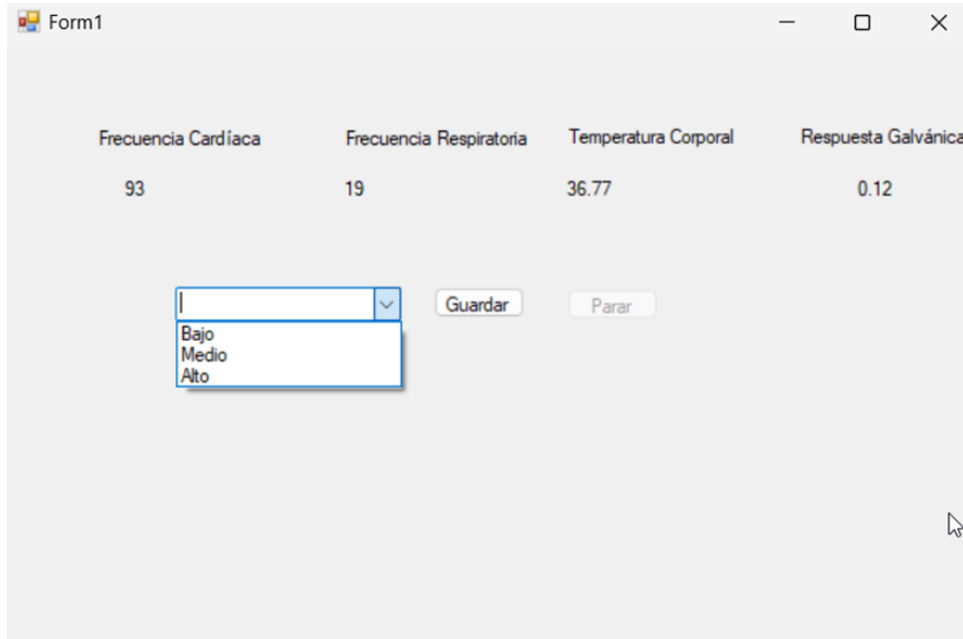
Salida en formato concreto: en esta última fase del proceso de ejecución del sketch se realiza el envío de los datos por el puerto serie. Estos datos están incluidos dentro de una estructura condicional que permite enviar los datos cada X segundos, de manera discreta, para que la escala temporal de dicha salida sea simple y ordenada.

Variable fisiológica	Valor	Unidad
Frecuencia Cardíaca	myBPM	Pulsaciones por minuto (BPM)
Frecuencia Respiratoria	fr	Respiraciones por minuto (RPM)
Respuesta Galvánica	gsr	Voltaje (V)
Temperatura Corporal	temp	Grados centígrados (°C)

Cuadro 4: Funciones para el cálculo de las variables fisiológicas.

Esta salida es recogida por una aplicación de lectura-escritura que nos sirve de apoyo para el procesamiento y traducción al formato .csv que define el conjunto de entrenamiento. La interfaz de esta herramienta está compuesta por una serie de etiquetas que indican los valores de cada variable fisiológica más un combo con los tres

niveles de dolor definidos para el estudio y dos botones, uno que permite escribir en un almacenar los datos de la aplicación, y otro que permite parar el almacenamiento de los datos.



The screenshot shows a window titled 'Form1' with a light gray background. At the top, there are four labels: 'Frecuencia Cardíaca', 'Frecuencia Respiratoria', 'Temperatura Corporal', and 'Respuesta Galvánica'. Below these labels are the corresponding numerical values: 93, 19, 36.77, and 0.12. In the lower-left area, there is a dropdown menu with a blue border and a downward arrow. The menu is open, showing three options: 'Bajo', 'Medio', and 'Alto'. To the right of the dropdown menu are two buttons: 'Guardar' and 'Parar'.

Figura 14: Interfaz de la aplicación que define el conjunto de entrenamiento.

Nota. Figura realizada por elaboración propia.

En este punto, ya se podría realizar el almacenamiento de las variables fisiológicas más el nivel de dolor, en tiempo real.

Para finalizar con este sprint, se realiza la implementación de esta parte del sistema de manera real a través de un ensayo clínico en el que se escogió un conjunto de X personas que participaron como voluntarios para la puesta en marcha del sistema en este punto y se les pidió que siguieran los siguientes pasos para poder realizar el estudio de manera correcta:

1. Situar al personal dentro de un ambiente relajado para que se encuentren en el mayor estado de reposo posible. Esto nos proporcionará una mayor fiabilidad de los datos obtenidos.
2. Colocar la felpa en la cabeza de manera pausada y con precaución.
3. Ajustar los sensores en su región anatómica correspondiente:

Sensor	Región anatómica	Dispositivo
Frecuencia Cardíaca	Lóbulo oído derecho	Fototransistor

Frecuencia Respiratoria	Frontal	Micrófono
Respuesta Galvánica	Antebrazo anterior	Electrodos
Temperatura Corporal	Maxilar	Sensor infrarrojo

Cuadro 5: Ubicación fisiológica de los sensores.

4. Preguntar al paciente si se encuentra cómodo.
5. Iniciar los programas de (Arduino) y (Visual Studio).
6. Observar que en la aplicación lepsyDatasetGenerator aparece un mensaje con la siguiente descripción: “El puerto se ha conectado correctamente”. Este mensaje representa que se ha comenzado la comunicación Arduino-Applicación.

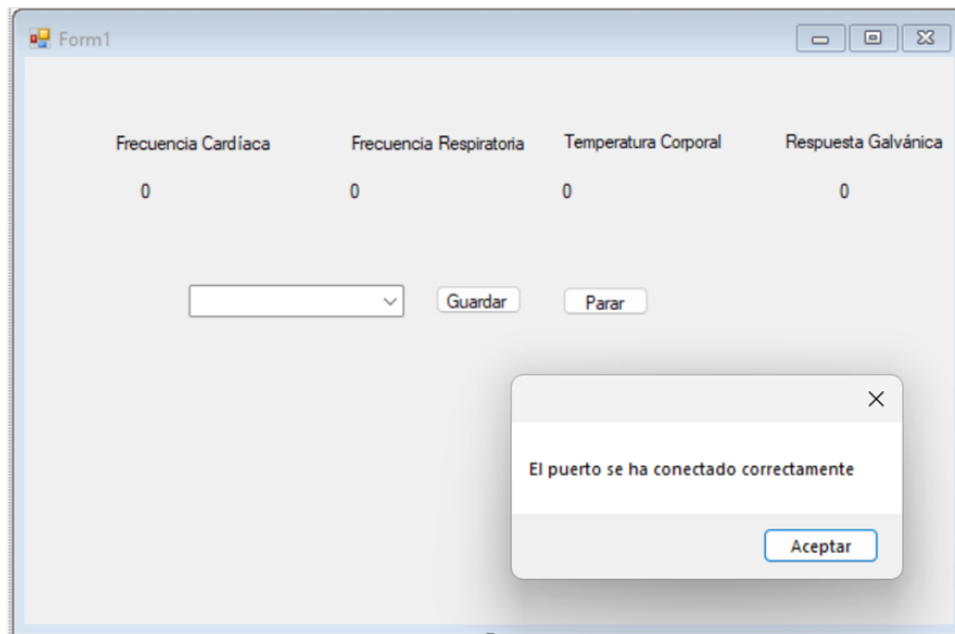


Figura 15: Mensaje de conexión correcta entre Arduino y Aplicación.

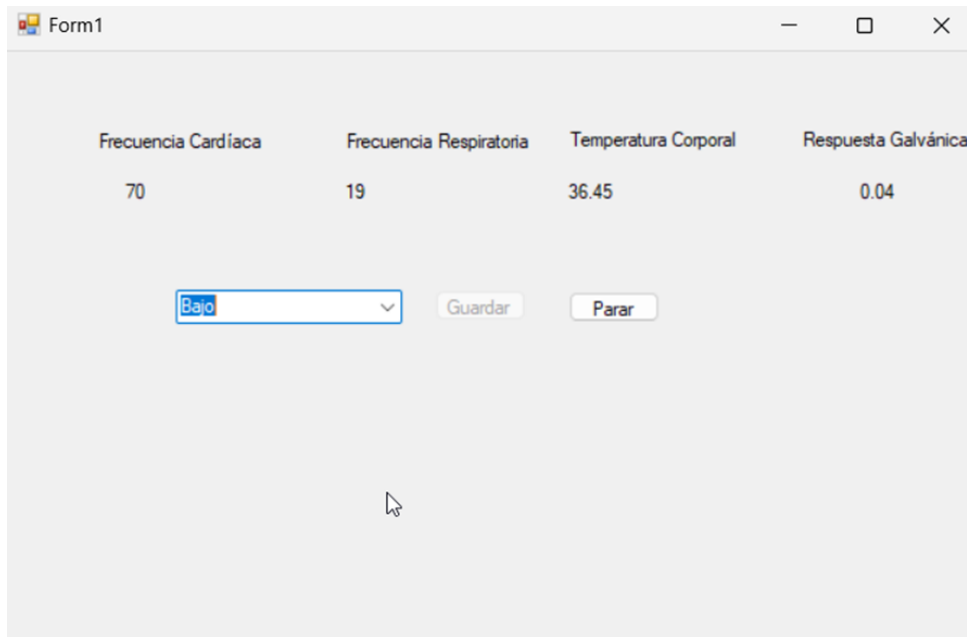
Nota. Figura realizada por elaboración propia.

7. Observar que los valores fisiológicos que se reciben se establecen dentro de los valores normales.
8. Indicar qué nivel de dolor está padeciendo el paciente en ese instante y presionar el botón “Guardar”.
9. Esperar unos segundos, o hasta que el paciente indique que se encuentra en otro nivel de dolor y pulsar el botón “Parar”.
10. Repetir el séptimo, octavo y noveno paso varias veces más.

2.2.3.1.3- Resultados:

En el ensayo clínico, se pidió al paciente que nos indicara el nivel de dolor que iba padeciendo a lo largo del estudio. A continuación, aparecen unos ejemplos de la toma de datos a lo largo del ensayo.

En este instante, el paciente está indicando que el nivel de dolor que padece es bajo. En principio, parece que las variables fisiológicas medidas en ese momento entran dentro del rango de valores bajos definido anteriormente.



Frecuencia Cardíaca	Frecuencia Respiratoria	Temperatura Corporal	Respuesta Galvánica
70	19	36.45	0.04

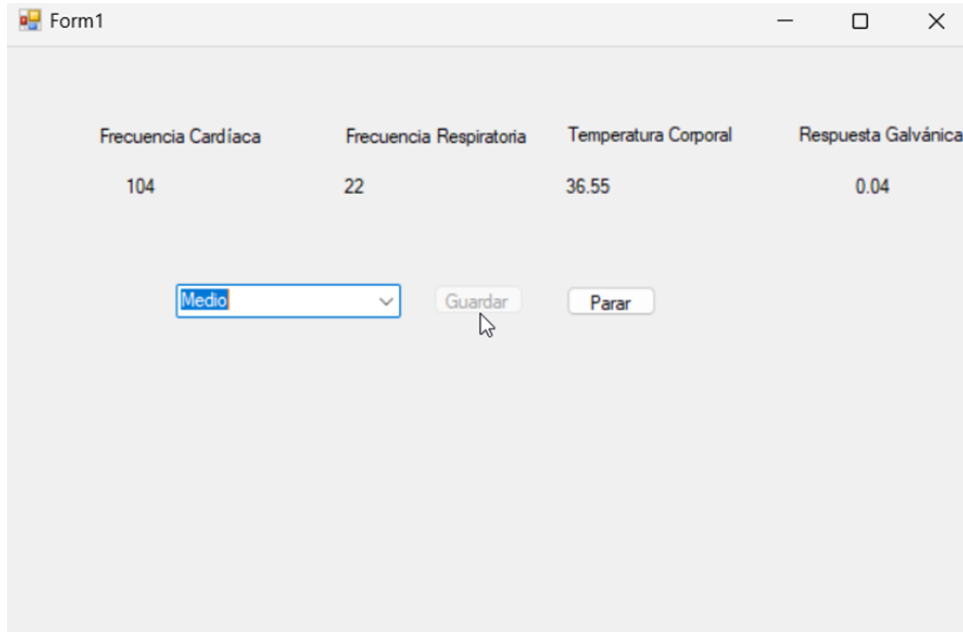
Bajo

Guardar Parar

Figura 16: Registro de nivel de dolor bajo.

Nota. Figura realizada por elaboración propia.

Minutos después, mientras el ensayo está en curso, el paciente recibió un estímulo el cual fue clasificado en nivel medio de dolor. Las variables que presentan mayores diferencias son la frecuencia cardíaca y respiratoria, mientras que la temperatura corporal solamente varía en unidades decimales y la respuesta galvánica se mantiene igual que en el estado anterior.



Frecuencia Cardíaca	Frecuencia Respiratoria	Temperatura Corporal	Respuesta Galvánica
104	22	36.55	0.04

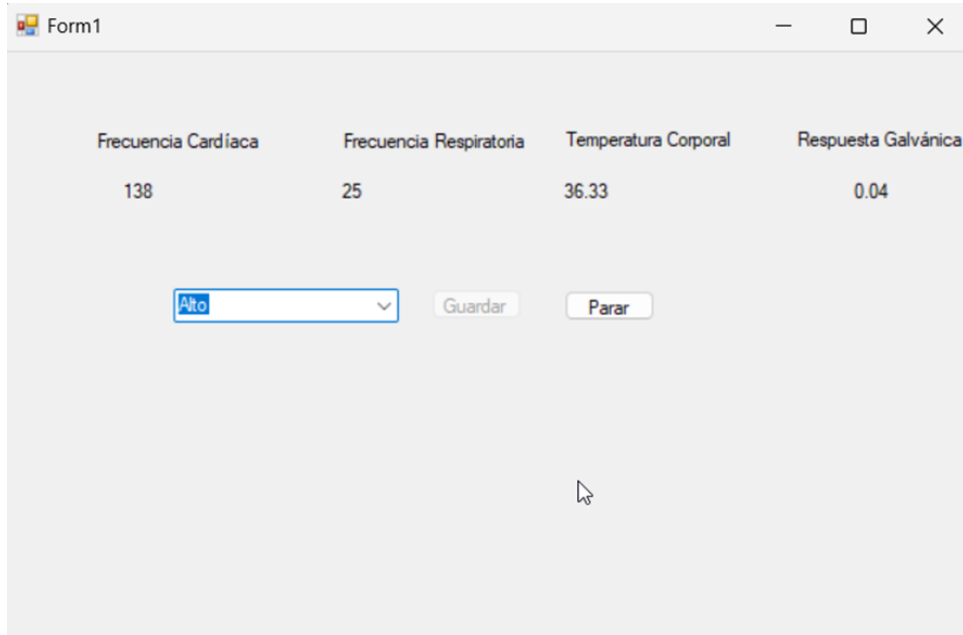
Medio ▼

Guardar Parar

Figura 17: Registro de nivel de dolor medio

Nota. Figura realizada por elaboración propia.

Más tarde, se indicó el estado de nivel de dolor alto en función de las variables fisiológicas recibidas. En este estado la respuesta galvánica tampoco presenta cambios comparando con el resto del estado, al igual que la temperatura corporal, cuya variación entre niveles de dolor es muy baja. Por lo que podemos afirmar que las variables más determinantes son la frecuencia cardíaca y la respiratoria. Esta premisa se intentará justificar más adelante a través del procesamiento de los datos.



Frecuencia Cardíaca	Frecuencia Respiratoria	Temperatura Corporal	Respuesta Galvánica
138	25	36.33	0.04

Alto (dropdown menu)

Guardar Parar

Figura 18: Registro de nivel de dolor alto

Nota. Figura realizada por elaboración propia.

Como podemos observar, estas imágenes muestran un ejemplo de los valores que presentan las variables fisiológicas mientras se registra el nivel nociceptivo que el paciente va indicando.

Una vez se ha concluido con el ensayo clínico se obtuvo el conjunto de entrenamiento (dataset), adecuado para el sistema con un tamaño de 1000 muestras.

En la imagen siguiente, aparece una imagen que representa el porcentaje de datos del conjunto de entrenamiento según la clasificación del nivel de dolor.

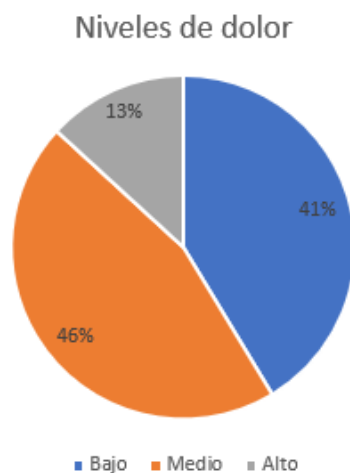


Figura 19: Diagrama circular. Porcentaje de características del atributo Nivel de dolor.

Nota. Diagrama realizado por elaboración propia.

Recalcamos que las muestras representan el valor de las variables fisiológicas y el nivel de dolor que el paciente ha clasificado en el estudio anterior.

Cada muestra se distingue de la anterior por un salto de línea y contienen los datos recibidos por los sensores, separados por coma dando lugar al fichero *lepasyDataset.csv*. Este fichero será almacenado y enviado al siguiente punto, dando como finalizado este sprint y comienzo a los sprints correspondientes al bloque inteligente. Una vez completados dichas fases, se puede dar paso al siguiente sprint del bloque instrumental, completando así todas las etapas del proyecto.

2.2.3.2- Sprint 4: Aplicación

2.2.3.2.1- Análisis:

A partir de ahora, necesitaremos la aplicación para obtener las variables fisiológicas e implemente el modelo seleccionado en los sprints equivalentes al bloque inteligente. En esos apartados, se llegó a la conclusión de que el mejor modelo para representar la lógica que permita al sistema clasificar el nivel de dolor sea el bosque aleatorio.

Los pasos de este sprint se basan en la conexión mediante el puerto serie al ordenador para que procese todos los datos recogidos de los componentes del circuito y recojan los datos del paciente en tiempo real e implementaremos un módulo bluetooth que permita enviar los resultados de la integración del modelo obtenido en el sprint anterior a una aplicación que permita conseguir todos los objetivos del proyecto. Para crear la aplicación utilizamos App Inventor para mostrar una interfaz que permita representar los resultados del sistema. Como resultado obtuvimos una aplicación, que llamamos LEPASY (level pain system), en la que se muestran los datos obtenidos por los sensores y el correspondiente resultado, clasificando el dolor en bajo, medio o alto en función de la lógica que representa el modelo obtenido en el sprint anterior.

La creación de la aplicación móvil se llevó a cabo utilizando App Inventor, una plataforma visual de programación que permite el desarrollo de aplicaciones sin necesidad de conocimientos avanzados de programación. Con App Inventor, diseñamos una interfaz intuitiva y amigable que mostraba los datos obtenidos por los sensores en tiempo real, así como los resultados de la evaluación del dolor realizada por el modelo de inteligencia artificial. Esta aplicación, que llamamos LEPASY (Level Pain System), proporcionaba

una herramienta útil y práctica tanto para los profesionales de la salud como para los pacientes.

2.2.3.2.2- Implementación:

La implementación de este sprint se compone de dos fases. La primera compone todo el proceso de implementación del resultado del modelo Bosque Aleatorio en el entorno de Arduino IDE, así como el envío de los resultados de la clasificación de los valores fisiológicos recibidos desde un módulo bluetooth a una aplicación. La segunda parte, define el proceso de recepción de los datos enviados por el módulo bluetooth de Arduino para mostrar por pantalla dichos resultados, permitiendo así la monitorización del paciente de manera visual.

Arduino:

El circuito de adquisición, procesamiento y transformación de señales fisiológicas se asimila mucho al circuito diseñado en el primer sprint al que hay que sumarle un módulo bluetooth. Por tanto, para el circuito de este sprint se necesitan los siguientes componentes: sensores, placa de Arduino, cableado, puerto serie, led RGB, resistencias (330 Ohmios), módulo bluetooth, felpa de tela elástica, mascarilla.

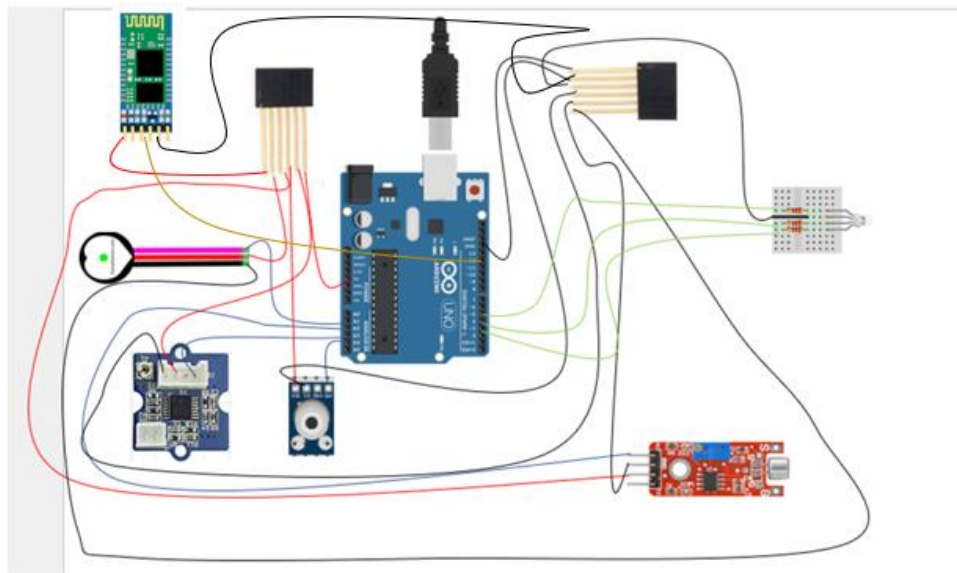


Figura 20: Circuito de recogida y envío de datos.

Nota. Circuito realizado por elaboración propia.

El nuevo componente que se suma al circuito es el HC-05. Este módulo permite enviar datos como si estuviera representando por el puerto serie.

Para implementar el modelo Bosque Aleatorio en el entorno de Arduino IDE, se utilizarán las estructuras condicionales para representar cada nodo del árbol de la siguiente manera:

Una vez la lógica obtenga el resultado concreto, se enviará un valor a la aplicación representado por un número para que esta la procese a través del módulo bluetooth.

Nivel de dolor	Resultado de la lógica
Bajo	1
Medio	2
Alto	3

Cuadro 6: Codificación de los valores del atributo Nivel de dolor en la lógica

A parte de este resultado de la lógica, se enviarán los valores fisiológicos recogidos en ese instante y que representan el nivel de dolor clasificado por el modelo.

La aplicación recibe estos datos y los procesa. Para ello, se almacenan estos valores en una lista cuyos índices serán mostrados en las etiquetas de la aplicación correspondientes.

Etiqueta	Índice
Frecuencia Cardíaca	1
Frecuencia Respiratoria	2
Respuesta Galvánica	4
Temperatura Corporal	3
Nivel de dolor	5

Cuadro 7: Índices de la lista que almacena cada una de las variables fisiológicas en la aplicación.

El índice que representa la clasificación del nivel de dolor es recibido por una estructura condicional que permitirá establecer la barra de la aplicación en distintos estados. Además, la barra presenta la opción de ir cambiando de dolor en función del nivel de dolor, así como situar el nivel de dolor debajo de unas imágenes obtenidas de la escala métrica EVA. A su vez, la luz RGB se irá encendiendo en el color que muestra la barra para así obtener más información sobre el nivel de dolor monitorizado en ese instante.

2.2.3.2.3- Resultados:

El entorno de Arduino IDE recibe los valores fisiológicos que son procesados en la lógica que implementa el modelo seleccionado. Esta lógica calcula el valor del nivel de dolor que padece el usuario en ese instante y que es enviado al mismo tiempo que los valores fisiológicos a través del módulo bluetooth a la aplicación que representa la

monitorización del nivel de dolor de manera visual. A continuación, aparecen algunas imágenes del proceso de monitorización de un paciente representado en la aplicación:

Una vez completada la fase de desarrollo, pasamos a la etapa de pruebas y validación del sistema. Realizamos pruebas exhaustivas para evaluar la precisión y confiabilidad de la detección de dolor realizada por nuestro sistema. Para ello, recurrimos a una variedad de escenarios y casos de prueba, involucrando a pacientes reales y comparando los resultados de nuestro sistema con las evaluaciones de dolor realizadas por profesionales médicos. Estas pruebas nos permitieron ajustar y afinar el modelo de inteligencia artificial, así como identificar posibles mejoras en la configuración de los sensores y la aplicación móvil.



Figura 21: Aplicación que monitoriza el nivel de dolor.

Nota. Figura realizada por elaboración propia.

En conclusión, el desarrollo de un sistema de detección de dolor basado en tecnologías como Arduino, inteligencia artificial y aplicaciones móviles presenta un gran potencial para mejorar la evaluación y el manejo del dolor en el ámbito médico. A lo largo de este proyecto, hemos diseñado y desarrollado un sistema que integra de manera efectiva todos estos componentes, permitiendo la detección precisa y la evaluación del dolor en los pacientes. Si bien el sistema ha demostrado ser prometedor en las pruebas

preliminares, es importante destacar que aún se requieren más investigaciones y validaciones para asegurar su eficacia y aplicabilidad en entornos clínicos. Sin embargo, los resultados obtenidos hasta ahora son alentadores y nos motivan a seguir mejorando y refinando este sistema en beneficio de los pacientes y los profesionales de la salud.

2.3. PRUEBAS DEL SISTEMA:

En este apartado se mostrará el plan de pruebas definido para verificar el funcionamiento del sistema desarrollado al completo. Para ello, definiremos una metodología de ciclos cuyo sumatorio de como resultado el plan de pruebas total del sistema que se obtiene al integrar todos y cada uno de dichos ciclos. Para definir dicha estructura, partimos de la base de que cada funcionalidad del sistema está compuesta por una serie de circuitos que definen el comportamiento del sistema. A su vez, para comenzar cada ciclo, puede ser necesario algún tipo de prerequisite general que indique cuándo, cómo y porque debe comenzar el ciclo en función de las necesidades del sistema. Definir este ítem es opcional, pues algunas veces no será necesario definir una acción que impida o permita comenzar el ciclo.

De forma más completa, se presenta la estructura que debe llevarse a cabo para la definición del plan de pruebas del sistema:

- Prerrequisitos generales (PG): pasos necesarios para ejecutar el ciclo, de manera ordenada.
 - Ciclos: pasos necesarios para ejecutar la prueba. Cada ciclo estará compuesto por uno o varios subciclos que describirán los circuitos que debe ir pasando el usuario que realice las pruebas del sistema. Este apartado corresponderá con la definición del test de usabilidad general propuesto para este sistema. Para ello, se utilizarán varias tablas compuestas por dos columnas, una que en la que se definan los criterios de validación o aceptación y otra en la que se defina el resultado esperado que confirma la realización del ciclo, por tanto, que ha pasado la prueba.
- PG_O1: El código debe estar desarrollado en un sketch programado en el entorno Arduino-IDE. Este código simplemente debe obtener, procesar, transformar y enviar datos desde Arduino UNO a la aplicación de WindowsForms, a través del puerto serie.

- PG_02: Conexión correcta placa entre placa de Arduino UNO-Arduino IDE y componentes.
 - Ciclo 1: Recibir datos biomédicos correctamente desde los sensores.

Criterios de validación / aceptación	Resultado esperado
Abrir entorno Arduino IDE -> Seleccionar puerto -> Compilar proyecto: lepasyVisualStudio.ino .	El proyecto se compila correctamente.
Abrir proyecto de Windows Forms, lepasys , desde visual studio -> Arrancar aplicación (botón Iniciar).	Aparece un mensaje que “el puerto se ha conectado correctamente”. Se abre una ventana en la que aparecen las cuatro variables biomédicas capturadas por Arduino UNO actualizándose en tiempo real.

Cuadro 8: Ciclo 1 – Recibo de datos biomédicos correctamente desde los sensores.

- Ciclo 2: Registro de nivel de dolor.

Criterios de validación / aceptación	Resultado esperado
Abrir entorno Arduino IDE -> Seleccionar puerto -> Compilar proyecto: lepasyVisualStudio.ino .	El proyecto se compila correctamente.
Abrir proyecto de Windows Forms, lepasys , desde visual studio -> Arrancar aplicación (botón Iniciar).	Aparece un mensaje que “el puerto se ha conectado correctamente”. Se abre una ventana en la que aparecen las cuatro variables biomédicas capturadas por Arduino UNO actualizándose en tiempo real. En la ventana hay un combo que permite especificar el nivel de dolor que padece en ese instante.

Cuadro 9: Ciclo 2 - Registro de nivel de dolor

- Ciclo 3: Almacenamiento de valores en fichero con formato .csv.

Criterios de validación / aceptación	Resultado esperado
Abrir entorno Arduino IDE -> Seleccionar puerto -> Compilar proyecto: lepasyVisualStudio.ino .	El proyecto se compila correctamente.
Abrir proyecto de Windows Forms, lepasy , desde visual studio -> Arrancar aplicación (botón Iniciar).	<p>Aparece un mensaje que “el puerto se ha conectado correctamente”.</p> <p>Se abre una ventana en la que aparecen las cuatro variables biomédicas capturadas por Arduino UNO actualizándose en tiempo real.</p> <p>En la ventana hay un combo que permite especificar el nivel de dolor que padece en ese instante.</p> <p>Pulsar el botón Guardar (automáticamente se deshabilita), esperar un momento y pulsar el botón Parar (se vuelve a activar el botón Guardar).</p>
Acceder a la siguiente ruta, dentro del explorador de archivos del sistema:	Aparece un fichero en formato .csv con los valores registrados entre la pulsación del botón Guardar y la pulsación del botón Parar.
Abrir el fichero .csv	Aparece un fichero cuyas columnas corresponden a las cuatro variables biomédicas obtenidas más el nivel de dolor registrado.

Cuadro 10: Ciclo 3 - Almacenamiento de valores en fichero con formato .csv

- PG_O3: Debe haber un conjunto de entrenamiento disponible para el estudio de IA, así como una copia de este.

- PG_04: Todas las librerías deben estar importadas en el entorno de desarrollo correspondiente (Google Colab / Weka).
- PG_05: La lógica que implementa el modelo definido en el sistema debe integrarse en el entorno de Arduino IDE y no en la aplicación.
- PG_06: La aplicación, realizada en App Inventor, y la placa de Arduino deben estar correctamente conectados vía Bluetooth.
- PG_07: El entorno de Arduino debe ser emisor de datos y la aplicación receptora de datos.
 - Ciclo 1: Recepción de datos biomédicos en la aplicación App Inventor.

Criterios de validación / aceptación	Resultado esperado
Abrir entorno Arduino IDE -> Seleccionar puerto -> Compilar proyecto: lepasyAppInventor.ino .	El proyecto se compila correctamente.
Abrir proyecto de App Inventor, lepasys , -> Pulsar botón Conectar -> Seleccionar el dispositivo cuyo nombre es BTJI -> Conectar bluetooth	Aparece un mensaje que “Conexión establecida”, indicando que la conexión entre el componente bluetooth de Arduino IDE con la aplicación es correcta. Se abre una ventana en la que aparecen las cuatro variables biomédicas capturadas por Arduino UNO actualizándose en tiempo real.
Esperar a que se abra la pantalla principal de recepción de datos biomédicos.	Los datos biomédicos se reciben correctamente en la aplicación (no hay errores, los valores concuerdan con las unidades y el dato al que refieren).

Cuadro 11: Ciclo 1 - Recepción de datos biomédicos en la aplicación App Inventor

- Ciclo 2: Clasificación del nivel de dolor.

Criterios de validación / aceptación	Resultado esperado
Abrir entorno Arduino IDE -> Seleccionar puerto -> Compilar proyecto: lepasyApplInventor.ino .	El proyecto se compila correctamente.
Abrir proyecto de ApplInventor, lepasys , -> Pulsar botón Conectar -> Seleccionar el dispositivo cuyo nombre es BTJI -> Conectar bluetooth	Aparece un mensaje que “Conexión establecida”, indicando que la conexión entre el componente bluetooth de Arduino IDE con la aplicación es correcta. Se abre una ventana en la que aparecen las cuatro variables biomédicas capturadas por Arduino UNO actualizándose en tiempo real. en ese instante.
Esperar a que se abra la pantalla principal de recepción de datos biomédicos.	Aparece una barra cuyo valor (verde, amarillo, rojo) corresponden a la recepción del envío de los datos procesados desde la lógica. Se ilumina el led RGB en función de la barra de usuario activa.

Cuadro 12: Ciclo 2 - Clasificación del nivel de dolor

BLOQUE 3: PLANIFICACIÓN DEL PROYECTO.

3.1- PLANIFICACIÓN TEMPORAL:

Para obtener de manera sencilla la información sobre la planificación temporal, estimada (inicial) y real (final), se definió un calendario laboral que simulara las horas disponibles en las que se realizaría el proyecto en función de la disponibilidad de los participantes del proyecto. En el propio calendario, también se definieron las fechas festivas siguiendo el calendario del curso escolar.

Horario	Fechas	Número de días	Horas/día
Jornada normal (17:00-20:00)	12/10/2022 15/02/2023	126 días	3 horas/día
Jornada reducida (12:00-14:00)	16/02/2023 20/04/2023	63 días	2 horas/día
Jornada completa (12:00-14:00) (17:00-21:00)	03/05/2023 01/07/2023	59 días	6 horas/días

Cuadro 13: Horario laboral del calendario del proyecto.

3.1.1- PLANIFICACIÓN TEMPORAL INICIAL

A continuación, aparece el diagrama de Gantt estimado para la consecución del proyecto.

La estimación de la duración de las tareas se calculó utilizando la ecuación de la estimación PERT:

$$E_{PERT} = \frac{TO + (4 \cdot TMP) + TP}{6}, \text{ donde:}$$

TO = optimista.

TMP = tiempo más probable.

TP = tiempo pesimista.

El resultado fue que la duración del proyecto corresponde a unas 402 horas.

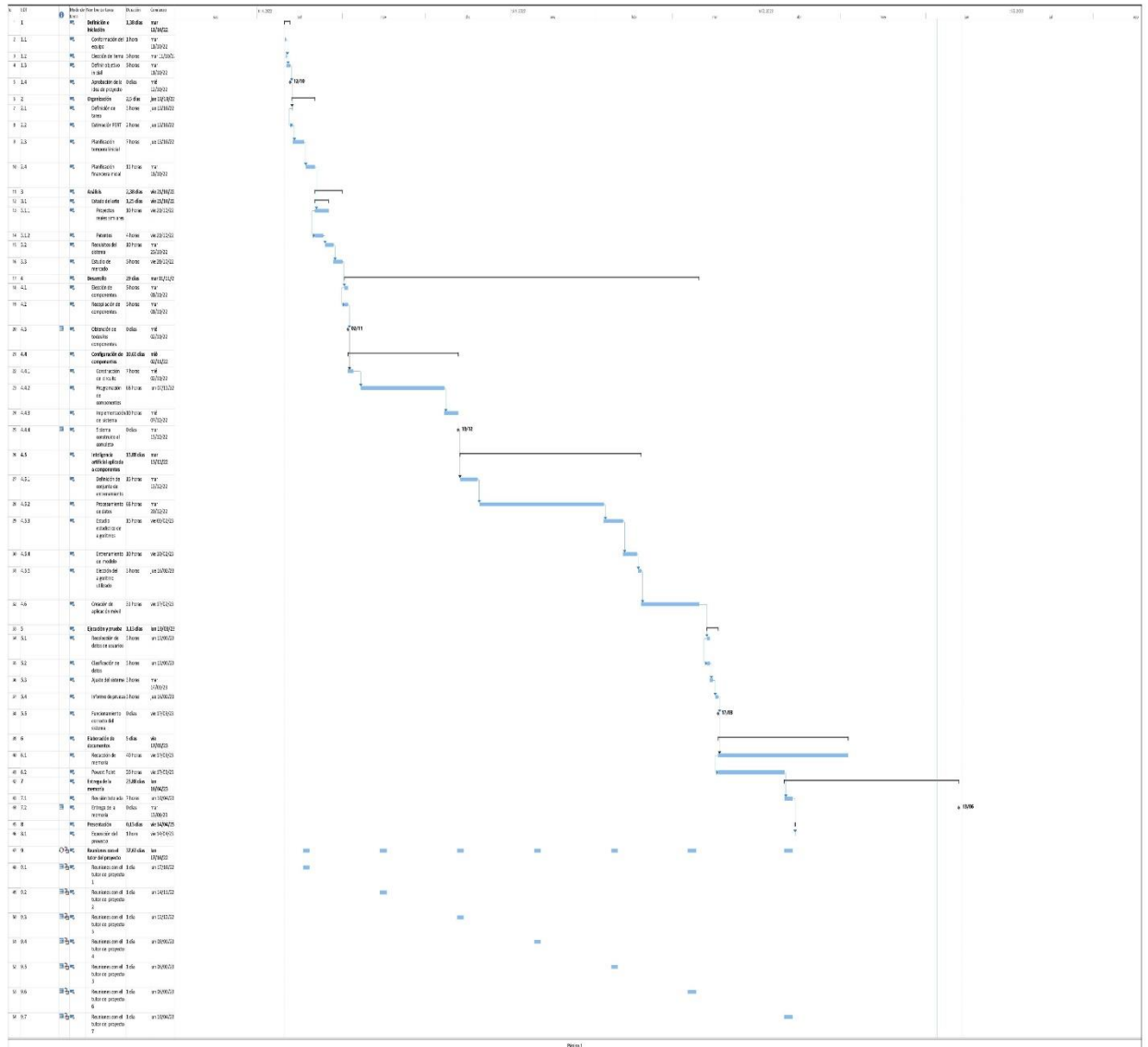


Figura 22: Diagrama de Gantt. Planificación temporal inicial.

Nota. Diagrama realizado por elaboración propia.

3.1.2- PLANIFICACIÓN TEMPORAL FINAL

A continuación, aparece el diagrama de Gantt estimado para la consecución del proyecto.

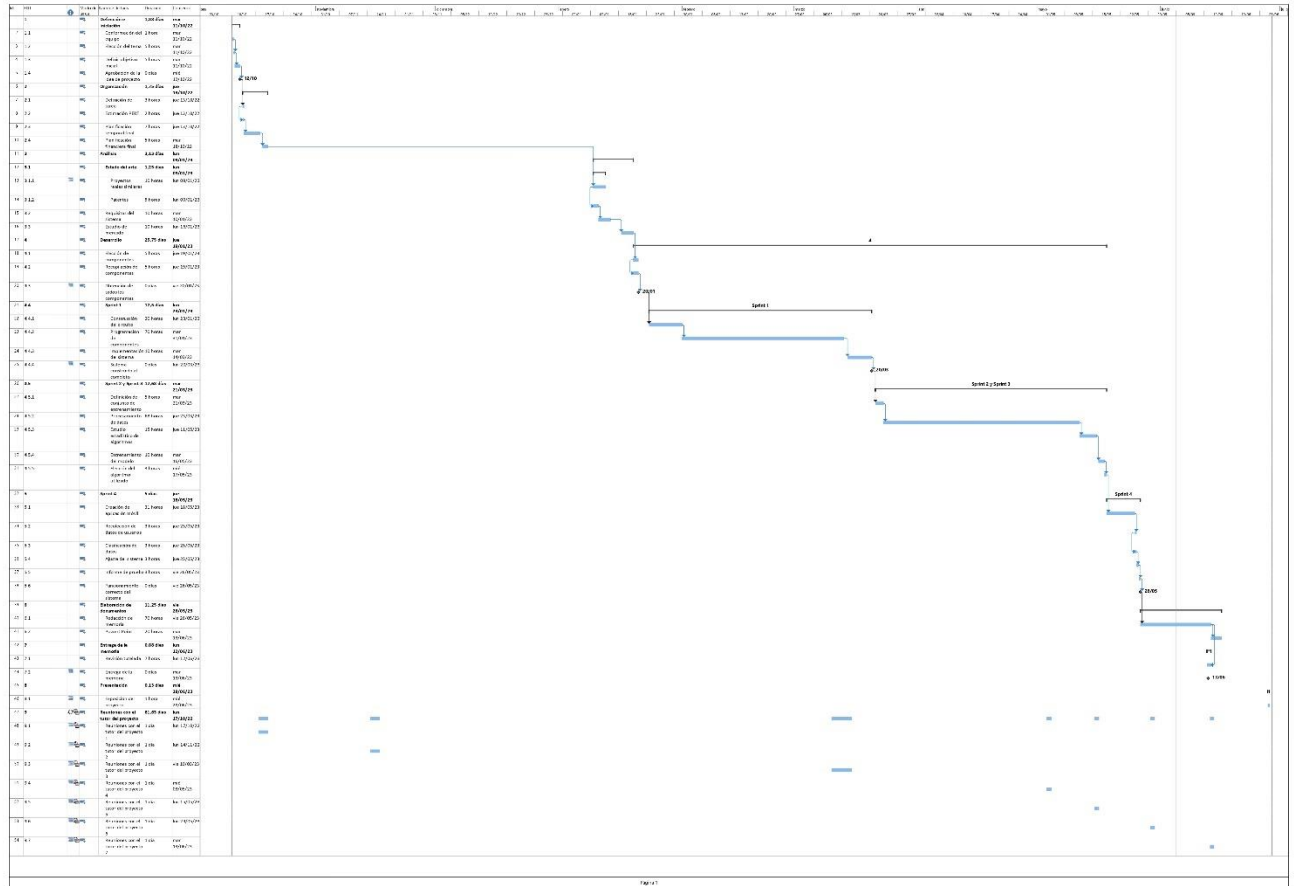


Figura 23: Diagrama de Gantt. Planificación temporal final.

Nota. Diagrama realizado por elaboración propia.

El resultado fue que la duración del proyecto corresponde a unas 442 horas.

3.2- PLANIFICACIÓN FINANCIERA:

La estructura financiera del proyecto se compone de dos grupos de análisis financiero, el primero tiene que ver con los recursos humanos, y el segundo, recursos materiales. Estos datos, sumados a la descripción del gasto total dan como resultado una planificación financiera adecuada a las características del proyecto.

3.2.1- PLANIFICACIÓN FINANCIERA INICIAL


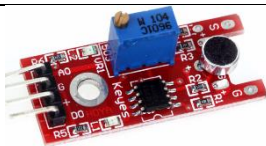


En este caso, los resultados que veremos a continuación representan una estimación financiera de los costes que supondría el sistema de manera aproximada.


Teniendo en cuenta que el presente proyecto se ha realizado en grupo, podemos calcular los costes para un proyecto real con el mismo número de personas que han realizado este trabajo. También, es importante mencionar que la estimación de horas de este proyecto, 402 horas, presentan un indicador de que dos es el número de personas necesarias para completar el sistema.

3.2.1.1- Recursos materiales

En este punto, aparecen los elementos que inicialmente se han decidido tener en cuenta para la estimación inicial del proyecto:

3.2.1.1.1- Hardware

Material	Imagen	Precio
Sensor de pulso		5€
Sensor de frecuencia respiratoria		8€
Arduino		30€
Felpa y cables		9€ + 6€

Led RGB y resistencias (330 Ohmios)		9€
TOTAL	—	67€

Cuadro 14: Recursos materiales Hardware iniciales.

3.2.1.1.2- Software

Los entornos de desarrollo software incluyen un precio base de adquisición de licencias mucho antes de la utilización de estos. Sin embargo, se estima que todos los entornos de programación forman parte de software libre y que no aplica ningún gasto de licencia o utilización. Simplemente se tendría en cuenta el precio de las licencias de Windows (259 EUR) y OpenOffice (51 EUR).

Entornos de desarrollo	Precio
Arduino IDE	0€ (incluido en la compra de la placa Arduino UNO)
Visual Studio	0€
Google Colab	0€
App Inventor	0€
Licencia Windows 10	259€
Licencia OpenOffice	51€
TOTAL	310€

Cuadro 15: Recursos materiales Software iniciales.

3.2.1.2. Recursos humanos:

Basándonos en XVII Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública (TIC) publicado en el BOE el 22 de febrero del 2018 podemos analizar los salarios fijos anuales en función de los grupos profesionales que se definen [https://www.boe.es/eli/es/res/2018/02/22/\(3\)](https://www.boe.es/eli/es/res/2018/02/22/(3)).

Considerando que las asignaturas estudiadas a lo largo de la carrera nos han permitido los conocimientos necesarios para afrontar el proyecto y la poca experiencia que los integrantes del equipo del proyecto tienen, se ha decidido clasificar a los dos miembros del equipo dentro del área 3 (consultoría, desarrollo y sistemas) nivel 3 del grupo C. Esto quiere decir que los recursos humanos de este proyecto pertenecen a un grupo profesional de tipo informático-técnico, responsables de la programación y supervisión

de las tareas definidas. Además, es necesario el rol de jefe de proyecto para un trabajo de semejante magnitud que será ocupado por el tutor del proyecto.

$$\text{Salario total} = \text{Salario base} + \text{Plus convenio}$$

Como observamos en la ecuación anterior, salario total está formado por la suma del salario base más el plus convenio. Realizando una observación al convenio colectivo, el salario total de las personas que se incluyen en dichos grupos profesionales es el siguiente:

Grupo profesional	Salario base	Plus convenio	Salario total (EUR/año)	Salario total (EUR/hora)
Jefe de Proyecto	25.035,54	1.754,77	26.790,31	14,88
Desarrollador de aplicaciones	18.222,75	1.277,25	19.500,00	10,83

Cuadro 16: Salario de los recursos humanos.

La ecuación para calcular el salario base por hora consiste en multiplicar en dividir el salario total de un año por el número de horas laborales que el mismo presenta.

A estos costes se les ha de añadir otra serie de primas que se calculan en función de los porcentajes determinados por la seguridad social y que corren a cargo de la empresa que contrata el equipo. La base de cotización se calcula como la suma de todos los conceptos salariales de la nómina del trabajador.

La seguridad social a cargo de la empresa cubre los siguientes riesgos:

- Contingencias comunes (CC): Cubre las derivadas de enfermedad común, accidente no laboral y maternidad. Este coste se calcula aplicando un 23,6% a la base de cotización de cada trabajador.
- Desempleo (DE): Cubre el riesgo de perder el trabajo para los que se encuentran en la situación de no ejercer la profesión por cualquier tipo de causa justificada. La cotización es distinta en función del tipo de contrato. En este proceso, el porcentaje correspondería en un 7,70% ya que el tipo de contratos del personal para la elaboración de un sistema de esta índole consiste en un contrato temporal que finalizaría una vez se concluya con el cierre del proyecto.

- Fondo de Garantía Salarial (FOGASA): este tipo de riesgo cubre las faltas de pago a los trabajadores derivadas de la mala praxis de los contratadores. Se calcula aplicando un 0,2 % sobre la base de cotización.
- Riesgo laboral (RL): ofrece las coberturas del riesgo de los accidentes de trabajo y enfermedades profesionales. Estos aspectos se dividen en la cotización por la incapacidad temporal y la de invalidez, muerte y supervivencia, cuya suma proporciona el porcentaje total cubierto:
 - 0,65% por el riesgo de incapacidad temporal derivado del trabajo
 - 1% riesgo de incapacidad permanente, muerte y supervivencia derivados del trabajo.
 - Total: $0.65\% + 1\% = 1.65\%$
- Formación profesional (FP): hace referencia al porcentaje destinado para los cursos de formación y reciclaje personal de los trabajadores. El porcentaje se calcula aplicando el 0.6% sobre la base de cotización.

Analizando cada porcentaje de cada riesgo, el porcentaje total sobre la base de cotización de cada trabajador de la seguridad social a cargo de la empresa es el siguiente:

$$\%Base\ Cotizacion = \%CC + \%DE + \%FOGASA + \%RL + \%FP$$

$$\%Base\ Cotizacion = = 23,6 + 7,7 + 0,2 + 1,65 + 0,6 = 33,75\%$$

Multiplicando el salario total en EUR/hora por el porcentaje de la base de cotización y el número de unidades, obtenemos que el coste por hora que genera cada uno de los trabajadores del proyecto es el siguiente:

Grupo profesional	Salario total (EUR/hora)	Base de cotización (%)	Total (EUR/hora)
Jefe de Proyecto	14,88	0,3375	19,90
Desarrollador de aplicaciones 1	10,83	0,3375	14,48
Desarrollador de aplicaciones 2	10,83	0,3375	14,48

Cuadro 17: Salario de los recursos humanos con la cotización a la seguridad social.

El cálculo de los costes de personal se calcula por cada tarea, duración de la misma y rol de la persona que la realiza. Se obtiene el coste asociado a realizar dicha tarea y para obtener el total se hace la suma del coste de todas las tareas.

Nombre	Rol RH	Salario (EUR/hora)	Estimación PERT (horas)	Coste (EUR)
Definición e iniciación				
Conformación del equipo	Jefe de Proyecto	19,90	1	19.9
	Desarrollador de aplicaciones 1			14.48
	Desarrollador de aplicaciones 2	14,48		
Elección del tema	Jefe de Proyecto	19,90	5	99.5
	Desarrollador de aplicaciones 1			72.4
	Desarrollador de aplicaciones 2	14,48		
Definir objetivo inicial	Desarrollador de aplicaciones 1		5	72.4
	Desarrollador de aplicaciones 2	14,48		
Aprobación de la idea de proyecto				
Total			11	278.68
Organización				
Definición de tareas	Desarrollador de aplicaciones 1		3	43.44
	Desarrollador de aplicaciones 2	14,48		
Estimación PERT	Desarrollador de aplicaciones 1		2	28.96
	Desarrollador de aplicaciones 2	14,48		
Planificación temporal inicial	Desarrollador de aplicaciones 1		7	101.36
	Desarrollador de aplicaciones 2	14,48		
Planificación financiera inicial	Desarrollador de aplicaciones 1		11	159.28
	Desarrollador de aplicaciones 2	14,48		
Total			23	333.04
Estado del arte				
Proyectos reales similares	Desarrollador de aplicaciones 1		10	144.80
	Desarrollador de aplicaciones 2	14,48		
Patentes	Desarrollador de aplicaciones 1		4	57.92
	Desarrollador de aplicaciones 2	14,48		
Requisitos del sistema	Desarrollador de aplicaciones 1		10	144.8
	Desarrollador de aplicaciones 2	14,48		
Estudio de mercado	Desarrollador de aplicaciones 1		5	72.4
	Desarrollador de aplicaciones 2	14,48		
Total			29	622.64

Elección de componentes	Desarrollador de aplicaciones 1	14,48	5	72.4
	Desarrollador de aplicaciones 2			
Recopilación de componentes	Desarrollador de aplicaciones 1	14,48	5	72.4
	Desarrollador de aplicaciones 2			
Obtención de todos los componentes			0	0
Configuración de componentes	Desarrollador de aplicaciones 1	14,48	85	1230
	Desarrollador de aplicaciones 2			
Construcción del circuito	Desarrollador de aplicaciones 1	14,48	7	101.36
	Desarrollador de aplicaciones 2			
Programación de componentes	Desarrollador de aplicaciones 1	14,48	68	984.64
	Desarrollador de aplicaciones 2			
Implementación del sistema	Desarrollador de aplicaciones 1	14,48	10	144.8
	Desarrollador de aplicaciones 2			
Sistema construido al completo	Desarrollador de aplicaciones 1		0	0
	Desarrollador de aplicaciones 2			
Inteligencia artificial aplicada a componentes	Desarrollador de aplicaciones 1	14,48	111	1607.28
	Desarrollador de aplicaciones 2			
Definición de conjunto de entrenamiento	Desarrollador de aplicaciones 1	14,48	15	217.2
	Desarrollador de aplicaciones 2			
Procesamiento de datos	Desarrollador de aplicaciones 1	14,48	68	984.64
	Desarrollador de aplicaciones 2			
Estudio estadístico de algoritmos	Desarrollador de aplicaciones 1	14,48	15	217.2
	Desarrollador de aplicaciones 2			
Entrenamiento del modelo	Desarrollador de aplicaciones 1	14,48	10	144.8
	Desarrollador de aplicaciones 2			
Elección del algoritmo utilizado	Desarrollador de aplicaciones 1	14,48	3	43.44
	Desarrollador de aplicaciones 2			
Creación de aplicación móvil	Desarrollador de aplicaciones 1	14,48	31	448.88
	Desarrollador de aplicaciones 2			
Total			237	5067.2

Ejecución y prueba				
Recolección de datos de usuarios	Desarrollador de aplicaciones 1	14,48	3	43.44
	Desarrollador de aplicaciones 2			
Clasificación de datos	Desarrollador de aplicaciones1	14,48	3	43.44
	Desarrollador de aplicaciones2			
Ajuste del sistema	Desarrollador de aplicaciones1	14,48	3	43.44
	Desarrollador de aplicaciones2			
Informe de prueba	Desarrollador de aplicaciones1	14,48	3	43.44
	Desarrollador de aplicaciones2			
Funcionamiento correcto del sistema			0	0
Total			12	173.76
Elaboración de documentos				
Redacción de memoria	Desarrollador de aplicaciones1	14,48	40	579.2
	Desarrollador de aplicaciones2			
Power Point	Desarrollador de aplicaciones	14,48	20	289.6
	Desarrollador de aplicaciones2			
Total			60	868.8
Entrega de la memoria				
Revisión tutelada	Jefe de Proyecto	19,90	7	139.3
Entrega de la memoria	Jefe de Proyecto	19,90	0	0
	Desarrollador de aplicaciones1	14,48		0
	Desarrollador de aplicaciones2			
Total			7	139.3
Presentación				
Exposición del proyecto	Jefe de Proyecto	19,90	1	19.90
	Desarrollador de aplicaciones1	14,48		14.48
	Desarrollador de aplicaciones2			
Total			1	34.38

Reuniones con el tutor del proyecto				
Reuniones con el tutor del proyecto 1	Jefe de Proyecto	19,90	1	19.90
	Desarrollador de aplicaciones1			
	Desarrollador de aplicaciones2	14,48		14.48
Reuniones con el tutor del proyecto 2	Jefe de Proyecto	19,90	1	19.90
	Desarrollador de aplicaciones1			
	Desarrollador de aplicaciones2	14,48		14.48
Reuniones con el tutor del proyecto 3	Jefe de Proyecto	19,90	1	19.90
	Desarrollador de aplicaciones1			
	Desarrollador de aplicaciones2	14,48		14.48
Reuniones con el tutor del proyecto 4	Jefe de Proyecto	19,90	1	19.90
	Desarrollador de aplicaciones1			
	Desarrollador de aplicaciones2	14,48		14.48
Reuniones con el tutor del proyecto 5	Jefe de Proyecto	19,90	1	19.90
	Desarrollador de aplicaciones1			
	Desarrollador de aplicaciones2	14,48		14.48
Reuniones con el tutor del proyecto 6	Jefe de Proyecto	19,90	1	19.90
	Desarrollador de aplicaciones1			
	Desarrollador de aplicaciones2	14,48		14.48
Reuniones con el tutor del proyecto 7	Jefe de Proyecto	19,90	1	19.90
	Desarrollador de aplicaciones1			
	Desarrollador de aplicaciones2	14,48		14.48
Reuniones con el tutor del proyecto 8	Jefe de Proyecto	19,90	5	99.5
	Desarrollador de aplicaciones1			
	Desarrollador de aplicaciones2	14,48		72.4
Reuniones con el tutor del proyecto 9	Jefe de Proyecto	19,90	5	99.5
	Desarrollador de aplicaciones1			
	Desarrollador de aplicaciones2	14,48		72.4
Reuniones con el tutor del proyecto 10	Jefe de Proyecto	19,90	5	99.5
	Desarrollador de aplicaciones1			
	Desarrollador de aplicaciones2	14,48		72.4
Total			22	756.36
Total(HORAS)			402	8274.16

Cuadro 18: Cálculo del gasto de los recursos en función de las horas estimadas.

Nota. Cuadro realizado por elaboración propia.

Puede parecer que el coste por horas de un miembro del equipo sea bastante bajo. Sin embargo, este coste está estipulado por el gobierno español por lo que la variación se lleva a cabo en función del plus convenio que cada empresa quiera aplicar, afectando al salario total de manera directamente proporcional. Por ello, podemos afirmar que el aumento o la disminución de los costes en la planificación financiera final, va a depender del número de componentes que aumenten los costes de los recursos materiales, así como del rendimiento de los miembros del equipo cuya eficiencia se verá reflejada en un mayor o menor número de horas dedicadas al proyecto, disminuyendo así los costes de los recursos humanos.

Por supuesto pensamos que, en el futuro, cuando el producto sea impulsado al mercado, tendrá numerosas empresas interesadas en el mismo y esto nos permitirá abaratar los costes totales para que el producto sea más barato para el cliente y, podrán incrementar las ganancias de los trabajadores. Todo ello suponiendo su éxito, del que estamos bastantes seguros debido a la gran utilidad de este.

3.2.2.3- Gasto total inicial:

Como podemos observar, en este apartado aparece una estimación inicial de los recursos materiales y humanos necesarios para la realización del proyecto.

En esta parte hemos obviado el sensor de temperatura corporal y de sudoración y se han tenido en cuenta el precio de las licencias de los productos software que vamos a utilizar.

Recursos materiales	Recursos humanos	Gasto total
377€	8274,16€	8651.16€

Cuadro 19: Gasto total del proyecto.

3.2.2- PLANIFICACIÓN FINANCIERA FINAL




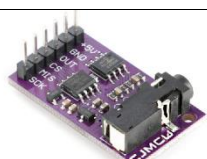


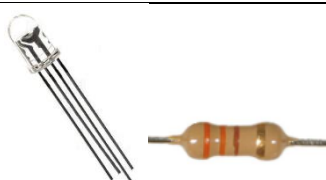
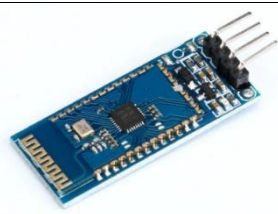
En este caso, la planificación se realiza con los datos proporcionados al realizar la totalidad del proyecto, sin ningún tipo de estimación.

3.2.2.1- Recursos materiales

En esta categoría se incluye la compra de todos los materiales necesario para elaborar el sistema. Al final del proyecto, el coste total de estos recursos se ha visto incrementados hasta llegar a la cifra de 116€ debido a que se han ido incluyendo nuevos componentes a lo largo del proyecto con los que no habíamos no contado en la

planificación financiera inicial. En la tabla que se muestra a continuación, se recogen los precios de cada material, para exponer de forma más concisa el coste de cada elemento que compone el sistema realizado.

3.2.2.1.1- Hardware

Material	Imagen	Precio
Sensor de pulso		5€
Sensor de frecuencia respiratoria		8€
Sensor de temperatura corporal		19€
Sensor de sudoración		19€
Arduino		30€
Felpa y cables		9€ + 6€
Led RGB y resistencia (330 Ohmios)		9€
Módulo Bluetooth		11€

TOTAL	—	116€
--------------	---	------

Cuadro 20: Recursos materiales Hardware finales.

Todos los precios de los materiales están cotejados con diferentes tiendas oficiales, como por ejemplo en Amazon. Como podemos observar, al ser el primer sistema que elaboramos los costos se mantienen más o menos constantes pues se han añadido varias componentes (bluetooth, sensor GSR y sensor de temperatura corporal) de bajo coste. Pero es cierto que, si se llegara a comercializar y tuviéramos que elaborar un mayor número del producto, podríamos abaratar costes ya que normalmente al comprar productos en masa, al por mayor, baja un poco el precio por cada unidad de material. Lo cual nos permitiría comprar componentes de mayor calidad y mejorar el diseño del producto lo máximo posible.

3.2.2.1.2- Software

Los entornos de desarrollo software suelen incluir un precio base de adquisición de licencias mucho antes de la utilización de estos. Sin embargo, se estima que todos los entornos forman parte del material proporcionado por la empresa, ningún gasto de licencia o utilización ha sido considerado:

Entornos de desarrollo	Precio
Arduino IDE	0€ (incluido en la compra de la placa Arduino UNO)
Visual Studio	0€
Google Colab	0€
App Inventor	0€
Licencia Windows 10	0€
Licencia OpenOffice	0€
TOTAL	0€

Cuadro 21: Recursos materiales Software finales.

Los entornos de desarrollo software suelen incluir un precio base de adquisición de licencias mucho antes de la utilización de estos. De hecho, todos los entornos que hemos utilizado forman parte de software libre cuyo uso no ha mostrado ningún gasto de licencia o derivados.

3.2.2.2- Recursos humanos:

Para recordar, el coste estimado en la planificación financiera inicial por hora que genera cada uno de los trabajadores del proyecto era el siguiente:

Grupo profesional	Salario total (EUR/hora)	Base de cotización (%)	Total (EUR/hora)
Jefe de Proyecto	14,88	0,3375	19,90
Desarrollador de aplicaciones 1	10,83	0,3375	14,48
Desarrollador de aplicaciones 2	10,83	0,3375	14,48

Cuadro 22: Salario de los recursos humanos real con la cotización de la seguridad social.

A continuación, se realiza el cálculo de los costes de personal se calcula por cada tarea final. Esto quiere decir que los valores que se obtengan presentan resultados reales ya que están basados en las tareas definidas en la planificación temporal final, obteniéndose así el coste asociado a realizar la suma de todas las tareas.

Nombre	Rol RH	Salario (EUR/hora)	Horas reales	Total
Definición e iniciación				
Conformación del equipo	Jefe de proyecto	19.9		19.9
	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	1	14.48
Elección del tema	Jefe de proyecto	19.9		99.5
	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	5	72.4
Definir objetivo inicial	Desarrollador de aplicaciones 1			
Aprobación de la idea de proyecto	Desarrollador de aplicaciones 2	14.48	5	72.4
Total			11	278.68
Organización				
Definición de tareas	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	3	43.44
Estimación PERT	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	2	28.96
Planificación temporal inicial	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	7	101.36
Planificación financiera inicial	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	5	72.4
Total			17	246.16
Estado del arte				
Proyectos reales similares	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	10	144.8
Patentes	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	5	72.4
Requisitos del sistema	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	10	144.8
Estudio de mercado	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	10	144.8
Total			35	506.8
Desarrollo				
Elección de componentes	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	5	72.4
Recopilación de componentes	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	5	72.4
Obtención de todos los componentes			0	0
Total			10	144.8
Sprint 1				
Construcción del circuito	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	20	289.6
Programación de componentes	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	70	1013.6
Implementación del sistema	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	10	144.8
Sistema construido al completo			0	0
Total			100	1448

Sprint 2 y Sprint 3				
Definición de conjunto de entrenamiento	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	5	72.4
Procesamiento de datos	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	68	984.64
Estudio estadístico de algoritmos	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	15	217.2
Entrenamiento del modelo	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	10	144.8
Elección del algoritmo utilizado	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	3	43.44
Total			101	1462.48
Sprint 4				
Creación de aplicación móvil	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	31	448.88
Recolección de datos de usuarios	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	3	43.44
Clasificación de datos	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	3	43.44
Ajuste del sistema	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	3	43.44
Informe de prueba	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	3	43.44
Funcionamiento correcto del sistema	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	36	521.28
Total			79	1143.92
Elaboración de documentos				
Redacción de memoria	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	70	1013.6
Power Point	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	20	289.6
Total			90	1303.2
Entrega de la memoria				
Revisión tutelada	Jefe de proyecto	19.9	7	139.3
	Jefe de proyecto			
Entrega de la memoria	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2		0	
Total			7	139.3
Presentación				
Exposición del proyecto	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	1	14.48
Total			1	14.48
Reuniones con el tutor del proyecto				
Reuniones con el tutor del proyecto 1	Jefe de proyecto	19.9		19.9
	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	1	14.48
Reuniones con el tutor del proyecto 2	Jefe de proyecto	19.9		19.9
	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	1	14.48
Reuniones con el tutor del proyecto 3	Jefe de proyecto	19.9		19.9
	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48	1	14.48

Reuniones con el tutor del proyecto 4	Jefe de proyecto	19.9	1	19.9
	Desarrollador de aplicaciones 1			
Reuniones con el tutor del proyecto 5	Desarrollador de aplicaciones 2	14.48	1	14.48
	Jefe de proyecto	19.9		19.9
Reuniones con el tutor del proyecto 6	Desarrollador de aplicaciones 1		1	
	Desarrollador de aplicaciones 2	14.48		14.48
Reuniones con el tutor del proyecto 7	Jefe de proyecto	19.9	1	19.9
	Desarrollador de aplicaciones 1			
	Desarrollador de aplicaciones 2	14.48		14.48
Total			7	240.66
Total(horas)			448	6928.48

Cuadro 23: Cálculo del gasto de los recursos en función de las horas reales.

Nota. Cuadro realizado por elaboración propia.

Como podemos observar, el número de componentes han aumentado, pues se ha añadido los módulos bluetooth, sensor de respuesta galvánica y sensor de temperatura corporal, dando lugar a un aumento de los costes relacionados con los recursos materiales. Sin embargo, el número de horas totales del proyecto han disminuido debido a la eficacia del trabajo del equipo que compone el proyecto, disminuyendo así los costes de los recursos humanos.

Por supuesto pensamos que, en el futuro, cuando el producto sea impulsado al mercado, tendrá numerosas empresas interesadas en el mismo y esto nos permitirá abaratar los costes totales para que el producto sea más barato para el cliente puesto que los componentes se irán obteniendo en masa, lo que permite que su precio se reduzca y se incrementen las ganancias del sistema. Todo ello suponiendo su éxito, del que estamos bastantes seguros debido a la gran utilidad de este y novedad dentro del mercado de productos biomédicos.

3.2.2.3- Gasto total final:

Como podemos observar, tanto los recursos materiales como los humanos se han disminuido con respecto al estimado en el plan inicial. Sin embargo, esta disminución de gasto no es muy significativa pues no hay mucha diferencia económica entre ellas.

Por una parte, el incluir el sensor de temperatura corporal y de sudoración ha hecho que los valores procesados sean mucho más fiables y que a la hora de procesarlos el resultado final sea lo más cercano posible a la realidad de los pacientes, que al fin y al cabo era nuestro objetivo principal del proyecto. De nada sirve todo este proyecto si ahora los parámetros no indican con certeza los resultados y se amoldan a la vida real ya que no cumpliría el objetivo de ser una herramienta de soporte y ayuda para médicos,

familiares, amigos e incluso los propios pacientes. Esta adición de componentes ha dado lugar a que el gasto en recursos materiales aumente.

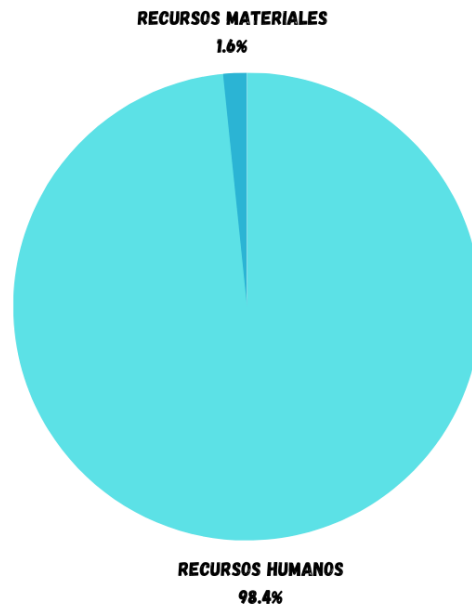
Si pensamos este proyecto como un sistema a gran escala de demanda, podemos afirmar que merece la pena reducir el precio total, ya que calidad-precio del producto se correspondería de manera más aproximada a las necesidades del cliente.

Recursos materiales	Recursos humanos	Gasto total
116€	6928,48€	7044.48€

Cuadro 24: Gasto total del proyecto.

El gasto final considerado es el elaborado evaluando mi trabajo de forma exclusiva, sin incluir el coste que supondría el trabajo de mi compañero, que sería el mismo ya que hemos realizado las mismas horas de trabajo.

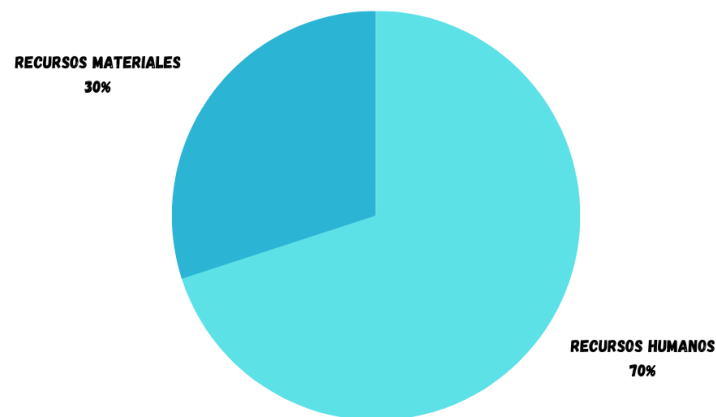
Como se observa, el gasto en recursos humanos se reduce considerablemente pues se las licencias de los productos software se despreció ya que estaban incluidas en el proyecto. También, se puede ver que el gasto en recursos humanos desciende en comparación del gasto total estimado en la planificación financiera inicial. Esto ocurre porque la organización del proyecto en sprints hizo que el esfuerzo de los elaboradores del proyecto centraran sus esfuerzos en los aspectos más relevantes del proyecto.



Cuadro 25: Comparación del gasto total que supone cada tipo de recurso.

Nota. Diagrama realizado por elaboración propia.

Como podemos observar en la gráfica los recursos humanos componen casi la totalidad del gasto. A futuro, nos gustaría dar un giro a esta tabla e intentar que los recursos materiales sean de la mejor calidad posible y el diseño sea lo más cómodo posible para los pacientes.



Cuadro 26: Objetivo de comparación del gasto total que supondría cada tipo de recurso.

Nota. Diagrama realizado por elaboración propia.

3.3- ESTUDIO DE MERCADO

Todo proyecto necesita conocer a sus posibles clientes potenciales sobre todo cuando se trata de uno con unas características como las nuestras. Además, la falta de experiencia de los miembros del equipo en la elaboración de un proyecto de esta índole nos obliga a realizar un estudio que nos haga conocer con exactitud la posición que debe tomar en el mercado el sistema propuesto y los requisitos que los clientes desean cumplir. Por ello, es de vital importancia realizar un estudio de mercado que nos permita adaptar el sistema al máximo a las necesidades de mercado y así como la adaptación a los posibles clientes. Cabe resaltar que, si los futuros clientes desean mejorar el sistema, se podría valorar ya que, al fin y al cabo, en este proyecto, se plantea más bien un prototipo base, que está en su propio auge ya que puede mejorar mucho más sus características y elevar su valor en el mercado.

3.3.1- CLIENTES POTENCIALES

En primer lugar, en este estudio nos gustaría identificar a los clientes potenciales del proyecto. Para ello, hemos hecho una gran búsqueda, teniendo en cuenta nuestros objetivos, sobre qué tipo de personas podrían estar interesadas en tener un sistema de estas características en su día a día principal. La respuesta fue sencilla los clientes incluirá tanto a algunas empresas, como a particulares que necesiten o quieran monitorizar el dolor tanto en el ámbito hospitalario correspondiente como en su casa sin necesidad de ir al hospital.

Algunas de las empresas que forman parte de este sector de diagnósticos serían:

1. Instituciones educativas: Escuelas, universidades o centros de formación que ofrecen programas de educación en campos relacionados con la bioinstrumentación, la medicina, la biotecnología, la ingeniería biomédica y disciplinas afines. Estas instituciones podrían estar interesadas en utilizar este sistema como una herramienta de aprendizaje práctico para enseñar a los estudiantes sobre la adquisición y procesamiento de señales biomédicas, así como sobre la aplicación de algoritmos de IA en la clasificación del nivel de dolor de manera técnico-práctica.
2. Centros de investigación: Organizaciones de investigación en áreas como la medicina, la salud, la neurociencia o la psicología, que llevan a cabo investigaciones relacionadas con el dolor y buscan herramientas avanzadas para medir y analizar las señales biomédicas asociadas al dolor. Este sistema

- podría proporcionarles una solución accesible y de bajo costo para la monitorización y clasificación del dolor en sus estudios.
3. Empresas de tecnología médica: Compañías dedicadas al desarrollo y fabricación de dispositivos y equipos médicos podrían estar interesadas en este sistema como una solución complementaria para la monitorización del dolor en entornos clínicos o de investigación, ofreciendo a los profesionales de la salud una herramienta adicional para evaluar y tratar el dolor en pacientes.
 4. Hospitales: para administrar una ayuda a su personal incluyendo a sus, médicos, enfermeros, fisioterapeutas y otros profesionales de la salud que trabajan en entornos clínicos o de rehabilitación, qué podrían encontrar útil este sistema para evaluar el nivel de dolor en sus pacientes. Les permitiría recopilar datos objetivos y cuantificables sobre el dolor, lo que podría respaldar el diagnóstico y la toma de decisiones en el tratamiento.

Es importante destacar que el éxito del sistema de monitorización del nivel de dolor dependerá de varios factores, como la calidad y precisión de las mediciones, la facilidad de uso, la confiabilidad de los resultados y la capacidad de integración con otros sistemas y herramientas existentes en el ámbito médico. Además, se deberá considerar la competencia existente en el mercado y la aceptación de nuevas tecnologías en el sector de la salud.

Un análisis más detallado y específico del mercado objetivo, incluyendo el tamaño del mercado, las tendencias del sector, la competencia y las necesidades de los clientes potenciales, sería necesario para una evaluación más precisa de las oportunidades comerciales y la estrategia de comercialización del sistema.

3.3.2. PLAN DE COMERCIALIZACIONES

En primer lugar, hemos realizado un análisis DAFO (debilidad, amenaza, fortaleza y oportunidad) para conocer la situación del proyecto sabiendo los puntos negativos, descritos como debilidades y amenazas, así como los puntos positivos que serían los de fortalezas y oportunidades. De esta misma manera, también vemos que las debilidades y las fortalezas son algo interno del proyecto que depende totalmente de nosotros, sin embargo, las amenazas y las oportunidades tienen que ver con exterior, por tanto, no depende solo de nosotros mismos, sino que también depende de los propios clientes y de la actividad del mercado.

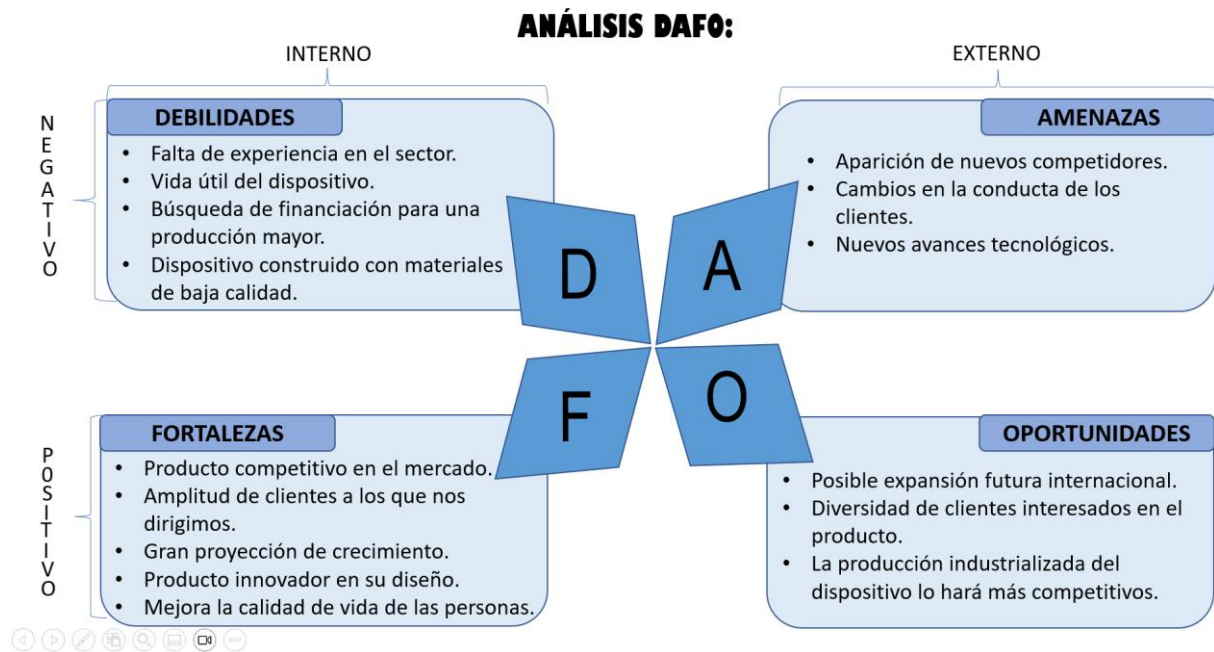


Figura 24: Análisis DAFO del sistema completo.

Una vez realizado el análisis DAFO, para desarrollar el plan de comercialización hemos identificado el mercado interesado, que serían los clientes potenciales definidos en el apartado anterior.

Nos centraremos en principio en la venta a particulares, hospitales y a empresas tecnológicas, ya que no tenemos stock suficiente para abastecer a más clientes. Evidentemente, el volumen de compras del hospital y la empresa no será el mismo que el de un particular, debido a este volumen el precio de cada aparato les saldrá más económico ya que nos permite comprar en masa y obtener precios más baratos.

Uno de los objetivos a futuro es obtener el precio más económico del mercado, y de esta manera, que el producto sea accesible para todos los posibles clientes ya sean empresas o particulares y también nos gustaría incrementar la calidad de nuestros sensores, ya que como hemos indicado en el análisis DAFO, teníamos muy poco presupuesto para elaborar el proyecto completo.

Por todos estos motivos nos inquieta bastante conseguir la financiación, ya que, sin ella no podremos cumplir nada de lo dicho anterior, nos sería imposible realizar el proyecto. Hay que tener en cuenta que el precio total que hemos calculado en el plan financiero sería únicamente para el primer lote del producto, ya que para los siguientes habría muchos pasos que nos saltaríamos al estar ya hechos, por tanto, el precio sería muy inferior, y muy competitivo en el mercado. Prevemos, que el producto será un éxito, con

mucha demanda, por tanto, será esencial la industrialización del proceso, ya que nos sería imposible vender a tantos clientes sin tener el montaje adecuado en un tiempo mínimo. Creemos que con la gran proyección que tiene el proyecto alguna empresa se embarcará en ayudarnos con la financiación.

También es muy importante conocer a la competencia, solo hay un producto que se parece al propuesto en el proyecto llamado PMD_200, por tanto, vemos muy importante conseguir un precio competitivo, que haga que nuestro producto se imponga por delante en el mercado.

Nuestro producto al contrario que el PMD_200 no está orientado solamente a hospitales, tenemos también la idea de darnos a conocer en otros sectores, obteniendo de esta manera clientes de distintas disciplinas, como he indicado anteriormente, es importante ya que tenemos más oportunidad que ellos de triunfar, habrá mayor número de clientes que conozcan nuestros dispositivos.

Por otro lado, también nos diferenciamos en las características de nuestro sistema, ya que hemos cambiado el acelerómetro que ellos utilizan, por un sensor de respiración, que según diversos estudios es un parámetro que varía dependiendo del dolor. Otra ventaja es que nosotros utilizamos el sensor de pulso, sin embargo, ellos utilizan fotoplestimografía lo cual es algo menos fiable, ya que cualquier flexión del dedo puede aumentar el tamaño del musculo dando valores inciertos, no funciona bien en pieles oscuras o con tatuajes, y los valores obtenidos pueden depender mucho de la presión que ejerza la cinta en el dedo.

Nosotros hemos apostado por la comodidad del paciente ya que al ser en la cabeza les permite llevar a cabo cualquier actividad con las manos, al contrario que en PMD_200 que al tenerlo en el dedo les impide realizar cualquier tipo de actividad durante la motorización.

El plan de marketing de nuestro dispositivo se hará mediante sitios web, redes sociales y anuncios en línea, también daremos conferencias para hablar sobre nuestro producto y visitaremos diferentes entidades con el fin de darnos a conocer entre nuestros posibles clientes.

En cuanto a la distribución, será sencilla, nuestra intención es crear una página web en la que nuestro producto se pueda comprar de forma online, esto al menos durará hasta conseguir una buena situación en el mercado, y para ello, necesitaremos distribuidores especializados en tecnología médica, la asociación con instituciones educativas para su

inclusión en programas de estudio, y la colaboración con proveedores de servicios de salud para su implementación en entornos clínicos. Cuando consigamos cumplir todos estos propósitos, nos plantearemos crear alguna tienda física, sobre todo en las grandes ciudades.

Hasta que el cliente se adapte al producto, contaremos con un sistema de atención al cliente totalmente gratuito, en el que estaremos dispuestos a atenderlos en cualquier momento si el producto da algún tipo de error o el cliente tiene alguna duda sobre su uso. Sin embargo, al ser un wearable fácil de manejar, no será necesario incluir personal de instalación del dispositivo.

Y por supuesto lo más importante para nosotros, será adaptar el producto a los requisitos impuestos por los clientes, por lo que, en el futuro haremos diferentes versiones de este, intentado amoldarlo a todas y cada una de las necesidades de nuestros clientes. Al final, en un plan comercial, lo más importante es obtener ganancias y que el cliente este satisfecho con el servicio prestado.

VII- CONCLUSIONES

La inteligencia artificial es un tema que actualmente tiene una gran demanda, ya que cada vez hay más sistemas que la utilizan y que en el futuro seguramente eliminará muchos puestos de trabajos porque existirán máquinas inteligentes capaces de hacerlo. Esto hará que se creen otro tipo de trabajos más orientados hacia la fabricación y cuidados de estas máquinas, estamos en medio de una de las revoluciones más grandes que ha podido ver el ser humano y está en nuestras manos aprovecharla sabiamente para avanzar y mejorar. Hoy en día, este es un tema muy controvertido, ya que aparecen preguntas sobre si puede llegar un momento en el que las máquinas puedan alcanzar más inteligencia que los seres humanos. Por ello, hay que tener cuidado porque al tener tanto poder se puede llegar a atentar contra la ética y la moral humana con leyes que establezcan los límites máximos a los que puede llegar esta nueva tecnología.

Este proyecto tiene una gran proyección de crecimiento en este sentido, ya que como hemos podido observar, podría ser una herramienta útil en diferentes disciplinas de trabajos médicos como el diagnóstico de enfermedades relacionadas con el nivel de dolor o la monitorización de pacientes en diferentes estados (hospitalización quirúrgica, rehabilitación, etc.).

Mediante un estudio de inteligencia artificial, hemos logrado elegir un modelo que permita entrenar los datos de un conjunto de entrenamiento, definido a partir de un ensayo clínico, de manera que el propio sistema implementa un reconocimiento de las variables fisiológicas de frecuencia cardíaca, frecuencia respiratoria y temperatura corporal y los clasifica en el nivel de dolor correspondiente con una precisión del 80%, aproximadamente, lo cual nos parece bastante elevado teniendo en cuenta los pocos recursos obtenidos para elaborar el proyecto. Con una financiación podríamos mejorar el sistema y aumentar ese porcentaje lo máximo posible, para que la máquina sea más fiable. Para ello, hemos trabajado con diferentes entornos, Google Colab, Visual Studio, Arduino y App Inventor, que nos han hecho aprender mucho sobre este tema. Por tanto, podemos decir que este proyecto no solo ha sido una mera investigación y estudio, sino que también nos ha servido de ayuda para ampliar nuestros conocimientos de cara a la vida laboral.

Es cierto que el dolor es algo subjetivo que cada persona sufre diferente, que a uno no le duela no significa que a otro sí, y es una de las causas más comunes, que dan lugar a que el paciente lleve peor cualquier tipo de enfermedad. Además, podemos afirmar

que el estudio de modelos de Inteligencia Artificial podría incluir la comparación con más algoritmos, pero pensamos que, en general, el estudio realizado es un éxito y permite demostrar la posibilidad de crear un sistema de motorización del dolor, sin necesidad de tener grandes recursos tecnológicos, sin duda, con ellos, el sistema aumentaría su fiabilidad y precisión.

Además, hemos podido comprobar que parámetros como el pulso, respiración, sudoración y temperatura permiten establecer una clasificación del nivel de dolor adecuada, aunque la correlación entre algunos de ellos sea baja. Seguro que hay más parámetros relacionados que se podrían introducir para mejorar la precisión del aparato, pero esto ya lo dejamos para estudios futuros.

En conclusión, pensamos que este proyecto tiene un gran futuro, es algo muy innovador, ya que hoy en día solamente hay un producto que se asemeje al nuestro. Además, hospitalariamente hablando, un sistema de estas características puede permitir una disminución de la congestión hospitalaria y un gran ahorro a la larga ya que evitaría muchos procesos hospitalarios derivados de un mal diagnóstico o tratamiento. Nosotros, hemos estudiado esta carrera para ayudar a las personas, pensar que con este aparato se pueda mejorar la calidad de vida de muchos ha sido lo que principalmente nos ha impulsado a elaborar el proyecto.

VIII. TRABAJO FUTURO

Tras finalizar este proyecto, hemos visto una serie de cuestiones a mejorar, al final lo que se muestra es un prototipo que con ciertos retoques en un futuro quizás se podría implantar en la vida real, y como hemos podido observar tras el estudio de mercado, sin duda sería un antes y un después en este ámbito de la medicina, con gran cantidad de clientes interesados.

Es cierto que, si hubiéramos elevado el presupuesto, simplemente con unos sensores con mayor precisión y una mejora del diseño exterior, se podría decir que más bien que prototipo se convertiría en la primera versión a la venta del producto con total seguridad.

También nos gustaría mejorar la aplicación dándole un aire más profesional, y añadiendo algunas pantallas aportando más datos a la aplicación, como por ejemplo poder acceder a graficas que elaboren un estudio estadístico de la motorización del dolor de manera automática, para que se obtenga más datos tanto para su diagnóstico como para la aplicación de su posterior tratamiento.

Con más tiempo podríamos haber analizado más algoritmos, para conocer si el actual definitivamente es el más preciso. También a la hora de obtener los resultados del análisis de datos, al final utilizamos lógica, sin embargo, pensamos que quizás con redes neuronales se obtendría un análisis más preciso.

Evidentemente esto es un caso hipotético, por tanto, no tenemos unos requisitos definidos por los verdaderos clientes, posibles compradores del producto, en caso de implementarlo de forma real tendríamos una cita para conocer sus propuestas, y establecer con mayor exactitud los requisitos en los que estarían más interesados a la hora de usar la aplicación, lo que nos haría mejorarla y hacer que sea lo más útil posible para aquellos que al final van a trabajar con el sistema en su día a día.

Por ello definitivamente, con más tiempo y presupuesto, pensamos que nuestro sistema podría mejorar todas estas cuestiones, para poder implementarlo de forma real, y convertirse en un producto muy prometedor de cara al público.

IX- BIBLIOGRAFÍA

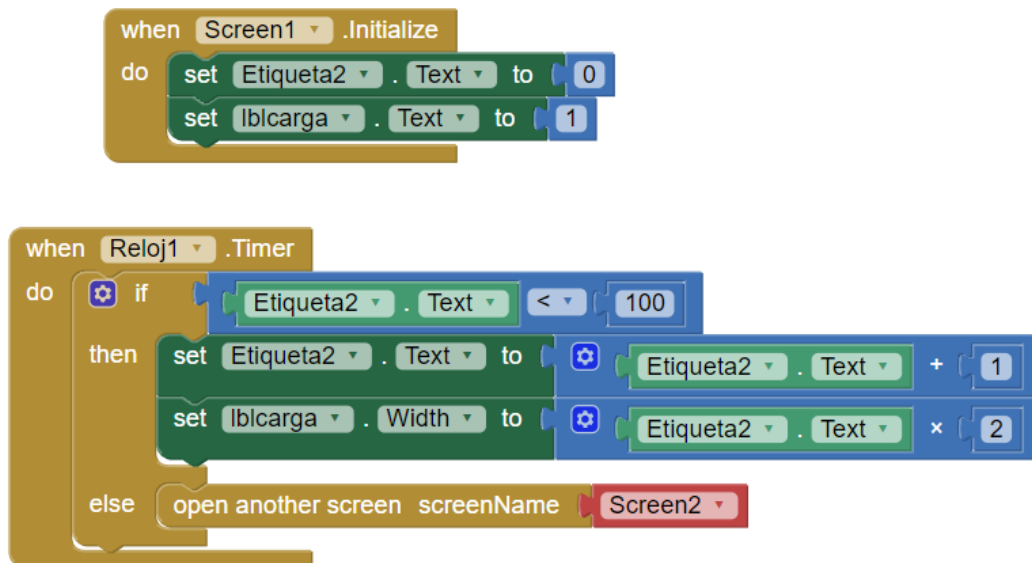
- [1] Ministerio de Sanidad. (s.f.). *Sanidad.gob*. Obtenido de Sanidad en un vistazo: <https://www.sanidad.gob.es/estadEstudios/sanidadDatos/home.html>
- [2] Berna, C.-B., Moreno Millán, N., González-Barboteo, J., Solsona Díaz, L., & Porta Sales, J. (Noviembre de 2011). Prevalencia de dolor como motivo de consulta y su influencia en el sueño: experiencia en un centro de atención primaria. *Atención primaria*, 43(11), 568-575.
- [3] MedaSense. (2023). *MedaSense*. Obtenido de PMD-200: <https://medasense.com/pmd-200/>
- [4] Meijer, F. S., Martini, C. H., Broens, S., Boon, M., Niesters, M., Aarts, L., . . . Dahan, A. (Mayo de 2019). Nociception-guided versus Standard Care during Remifentanyl–Propofol Anesthesia: A Randomized Controlled Trial. *130*, 745-755.
- [5] Meijer, F. S., Martini, C. H., Broens, S., Boon, M., Niesters, M., Aarts, L., . . . Dahan, A. (Diciembre de 2020). Reduced postoperative pain using Nociception Level-guided fentanyl dosing during sevoflurane anaesthesia: a randomised controlled trial. *British Journal of Anaesthesia*, 125(6), 1070-1078.
- [6] Saunders, R., & Weissbrod R. (Mayo de 2020). COST-BENEFIT OF PERSONALIZING INTRAOPERATIVE PAIN MANAGEMENT. *Value in health*, 23(5). Obtenido de ISPOR: <https://www.ispor.org/heor-resources/presentations-database/presentation/intl2020-3182/101924>
- [7] De la Vega, R., & Miró, J. (Julio de 2014). mHealth: a strategic field without a solid scientific soul. a systematic review of pain-related apps. *PLoS One*.
- [8] Barroso, G. (1 de Diciembre de 2016). Una app capaz de medir y monitorizar el dolor crónico. *InnovaSpain*. Obtenido de InnovaSpain.
- [9] Herrero, V. (s.f.). *Una aplicación digital permite a los médicos controlar el dolor crónico de sus pacientes*. Obtenido de 65 y Más : https://www.65ymas.com/salud/avances/quiero-conocer-aplicacion-monitoriza-dolor-cronico_11755_102.html
- [10] Garcia-Palacios, A. (s.f.). *Monitor de dolor 2.0*. Obtenido de Google Play Store: <https://play.google.com/store/apps/details?id=monitordedolor.code&hl=es>

- [11] Universitat Jaume I de Castelló. (s.f.). *Monitor de Dolor Multicéntrico*. Obtenido de App Store: <https://apps.apple.com/es/app/monitor-de-dolor-multic%C3%A9ntrico/id1546241257>
- [12] Hernández, P., García, C., Akhras, N., Azpurua, A., Hernández, J., Hernández, D., & Hernández, D. (2001). Puntos de gatillo. *Acta Odontológica Venezolana*, 39(1).
- [13] LaCasaDelFisio. (s.f.). *Palpatronic*. Obtenido de LaCasaDelFisio: <https://www.lacasadelfisio.com/584-palpatronic>
- [14] Pérez García, M., Ballesteros Martínez, F., Jiménez Alonso, J., Godoy García, J. F., & López de la Osa García, A. (2000). *España Patente nº ES2170014B1*.
- [15] Simmers, A. (s.f.). *Lenguaje de programación de Arduino, estructura de un programa*. Obtenido de Aprendiendo Arduino: <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-de-arduino-estructura-de-un-programa/>
- [16] García Romero, J., Jiménez Romero, M., Fernández Abascal, M., Fernández-Abascal Puente, A., Sánchez Carrillo, F., & Gil Fernández, M. (2002). La medición del dolor: una puesta al día. *Medicina Integral*, 39(7), 317-320.
- [17] Deus, J. (2009). ¿Se puede ver el dolor? *Reumatol Clínica*, 5(5), 228-232.
- [18] Brosschot, J. (2002). Cognitive-emotional sensitization and somatic health complaints. *Scand J Psychol*, 43(2), 113-121.
- [19] Gélinas, C. (Junio de 2016). Pain assessment in the critically ill adult: Recent evidence and new trends. *Intensive Crit Care Nurs*, 1-11.
- [20] Bendall, J. C., Simpson, P. M., & Middleton, P. M. (Diciembre de 2011). Prehospital vital signs can predict pain severity: analysis using ordinal logistic regression. *Eur J Emerg Med*, 334-339.
- [21] American Heart Association. (s.f.). *Todo acerca de la frecuencia cardíaca (pulso)*. Obtenido de GoRedForWomen: <https://www.goredforwomen.org/es/health-topics/high-blood-pressure/the-facts-about-high-blood-pressure/all-about-heart-rate-pulse>
- [22] Elsevier. (12 de Febrero de 2021). Efecto de la temperatura sobre la función cardíaca. *Elsevier Connect*.
- [23] Sapienza, Universidad De Roma. (2023). *Respuesta galvánica de la piel (GSR)*. Obtenido de Brain Signs: <https://www.brainsigns.com/es/science/s2/technologies/gsr>
- [24] Fowles, D., Christie, M., Edelberg, R., Grings, W., & Lykken, D. V. (1981). Committee report: Publication recommendations for electrodermal measurements. *Psychophysiol*, 232-239.

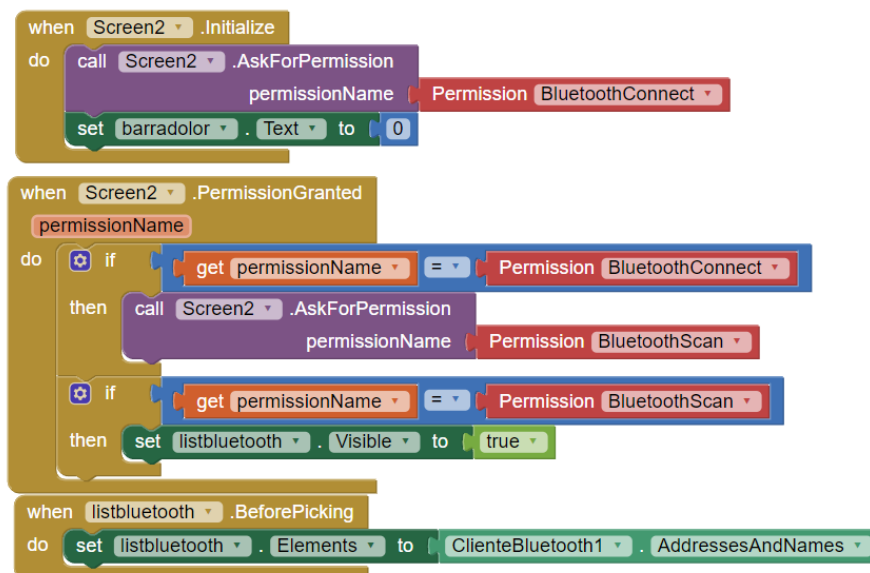
X- ANEXO

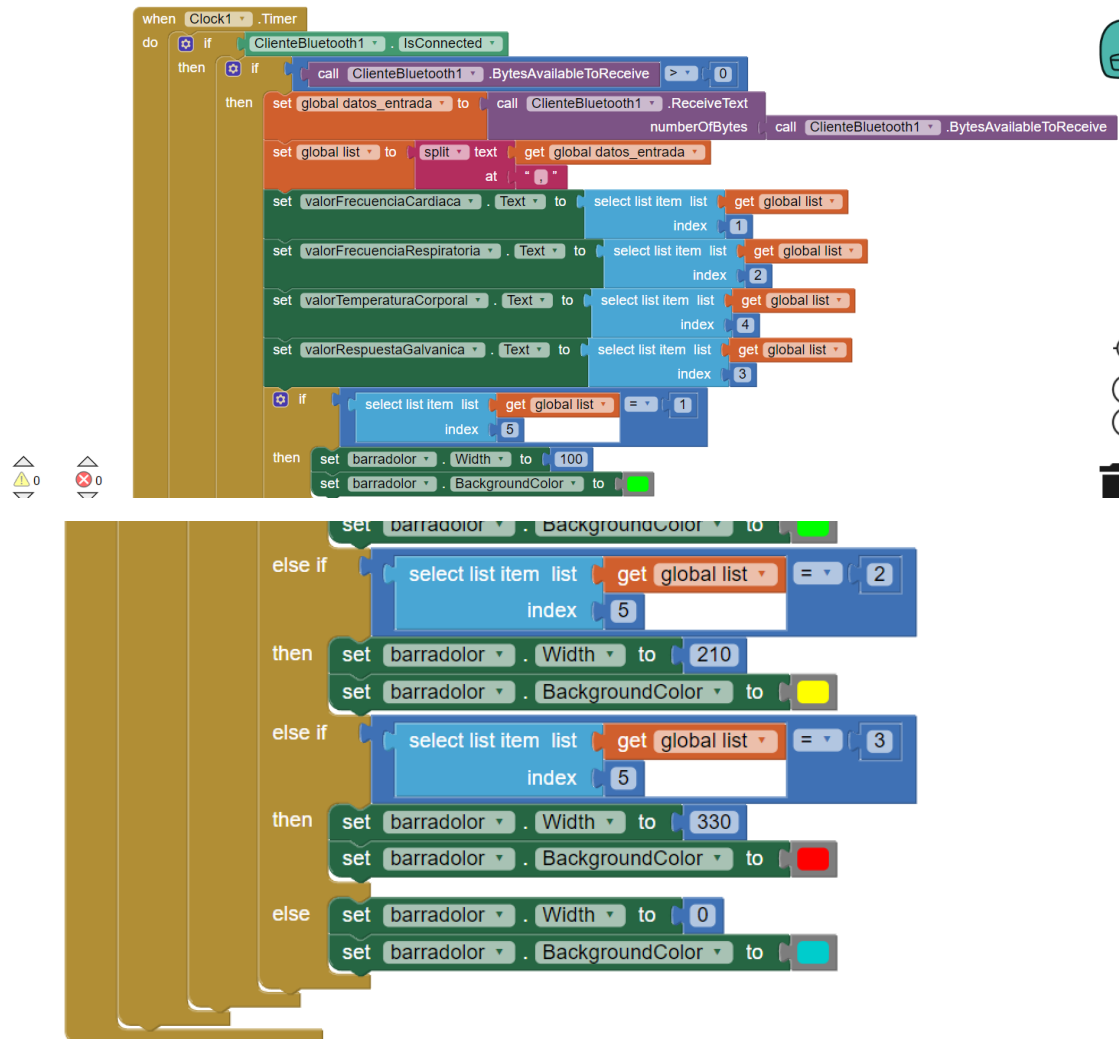
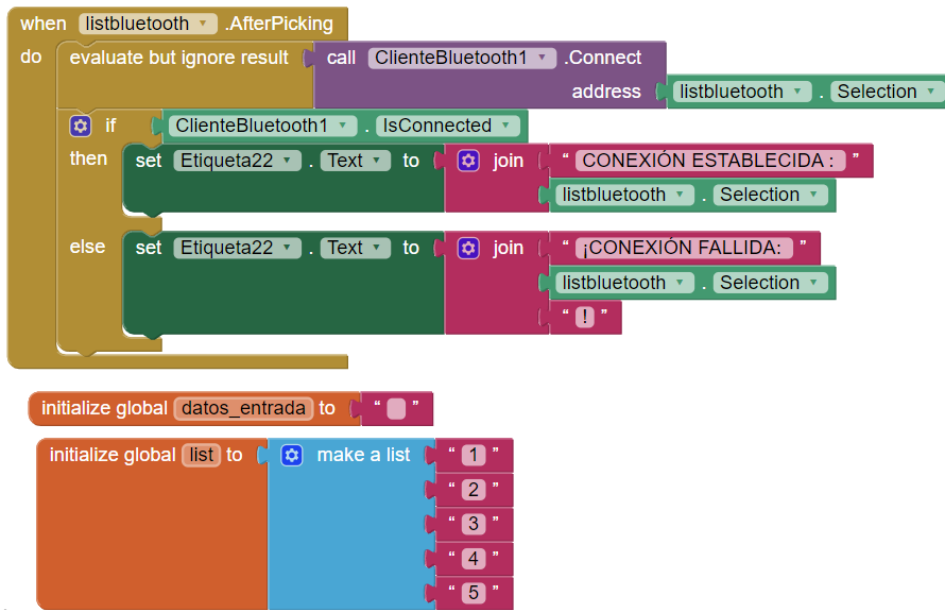
X.I- PROGRAMACIÓN POR BLOQUES DEL APPINVENTOR

Screen1:



Screen2:





X.II. LEPASYVISUALSTUDIO.INO

```
#define USE_ARDUINO_INTERRUPTS true

#include <PulseSensorPlayground.h>
#include <Wire.h>
#include <Adafruit_MLX90614.h>

Adafruit_MLX90614 mlx = Adafruit_MLX90614(); //Objeto mlx de tipo
Adafruit_MLX90614
PulseSensorPlayground pulseSensor; //Objeto pulseSensor de tipo
PulseSensorPlayground

int pulse;
float gsr;
int threshold = 550; //Definición del umbral de pico de tensión en la fotopletismografía
int media=0;
int suma;
int i = 0;

char mensaje;
int myBPM;
int fr;
int frecuenciaRespiratoria;
int respiraciones = 0;
const int tiempoEspera = 1000;
unsigned long tiempoAnterior = 0;
float gsra;
float tempC;
unsigned long previousMillis = 0; // Almacena el valor del tiempo anterior
const unsigned long interval = 2000; // Intervalo de tiempo deseado en milisegundos
(en este caso, 2000 ms para 2 segundos)

void setup() {

    pulseSensor.analogInput(1); //Almacena el valor de la lectura analógica del pin A1,
    sensor de frecuenciaCardiaca en el objeto pulseSensor.
    pulseSensor.setThreshold(threshold); //Se establece el valor del umbral definido.
    pulseSensor.begin(); //Comienzo del calculo de la frecuenciaCardiaca.
    mlx.begin(); //Comienzo del calculo de la temperaturaCorporal.
    pinMode(3, OUTPUT); //Definición del pin digital 3 como salida.
    pinMode(4, OUTPUT); //Definición del pin digital 4 como salida.
    pinMode(5, OUTPUT); //Definición del pin digital 5 como salida.

    Serial.begin(9600);
```

```
//Inicio de variables resultado a 0.
myBPM=0;
fr=0;
gsr=0;
tempC=0;
}

void loop() {
  int myBPM = pulseSensor.getBeatsPerMinute(); //Calculo de las pulsaciones por
  minuto.
  int fr = analogRead(A2); //Lectura del valor analógico del pin A2.
  float fra = fr * (5.0 / 1023.0); //Paso del valor analógico al rango de voltaje [0, 5].
  int myRPM = frEstimate(fra); //Función que estima el valor de la
  frecuenciaRespiratoria.
  float gsr = analogRead(A3); //Lectura del valor analógico del pin A3.
  float gsra = (gsr/1023)*5; //Paso del valor analógico al rango de voltaje [0, 5].
  float tempC = mlx.readObjectTempC(); //Calculo de la temperaturaCorporal.

  //.....Estructura que envía los datos cada 2 segundos.....
  unsigned long currentMillis = millis(); // Obtiene el tiempo actual

  if (currentMillis - previousMillis >= interval) { // Compara si ha pasado el intervalo de
  tiempo
    Serial.print(myBPM);
    Serial.print(",");
    Serial.print(myRPM);
    Serial.print(",");
    Serial.print(tempC);
    Serial.print(",");
    Serial.print(gsra);
    Serial.println(",");
    previousMillis = currentMillis; // Actualiza el valor del tiempo anterior
  }
}
//.....

//Función que estima el valor de la frecuenciaRespiratoria.....

bool haSoplado = false;
int frEstimate (float fra){
  unsigned long tiempoActual = millis();
  Serial.print(fra);
  Serial.print(", ");
  Serial.print(tiempoActual);
```

```
Serial.print(" ");
Serial.println(tiempoAnterior);
if (fra > 2.0 && haSoplado == false){} && ((tiempoActual - tiempoAnterior) >
tiempoEspera)) {
    //.....Captura de respiración. Ilumina el Led cuando se detecta
    respiracion.....
    haSoplado = true;
    digitalWrite(3,LOW);
    digitalWrite(4,HIGH);
    digitalWrite(5,LOW);
    //delay(50);          // Espera un tiempo para evitar la detección múltiple

    respiraciones++;      // Incrementa el contador de respiraciones

}
else if (fra <= 2.0 && haSoplado == true)
{
    haSoplado = false;

}
else
{
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(5,LOW);
}

delay(50); // Pequeña pausa antes de leer el sensor nuevamente

// Calcula la frecuencia respiratoria cada segundo
if (tiempoActual - tiempoAnterior >= tiempoEspera) {
    frecuenciaRespiratoria = (respiraciones / ((float)tiempoActual / 1000)) * 60; //
Cálculo de la frecuencia respiratoria en respiraciones por minuto
    tiempoAnterior = tiempoActual; // Actualiza el tiempo anterior
}
return frecuenciaRespiratoria;
}
```

X.III. LEPASYAPPINVENTOR.INO

```
#define USE_ARDUINO_INTERRUPTS true
```

```
#include <SoftwareSerial.h>
```

```
#include <PulseSensorPlayground.h>
```

```
#include <Wire.h>
```

```
#include <Adafruit_MLX90614.h>
```

```
SoftwareSerial BTSerial(10, 11); //RX , TX
```

```
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
```

```
PulseSensorPlayground pulseSensor;
```

```
int pulse;
```

```
float gsr;
```

```
int threshold = 550;
```

```
int media=0;
```

```
int suma;
```

```
int i = 0;
```

```
char buffer[10];
```

```
char mensaje;
```

```
int myBPM;
```

```
int fr;
```

```
int frecuenciaRespiratoria;
```

```
int respiraciones = 0;
```

```
const int tiempoEspera = 1000;
```

```
unsigned long tiempoAnterior = 0;
```

```
float gsra;
```

```
float tempC;
```

```
unsigned long previousMillis = 0; // Almacena el valor del tiempo anterior
```

```
const unsigned long interval = 2000; // Intervalo de tiempo deseado en milisegundos (en este caso, 2000 ms para 2 segundos)
```

```
void setup() {

    pulseSensor.analogInput(1);
    pulseSensor.setThreshold(threshold);
    pulseSensor.begin();
    mlx.begin();
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(10, OUTPUT);

    Serial.begin(9600);
    BTSerial.begin(9600);

    myBPM=0;
    fr=0;
    gsr=0;
    tempC=0;
}

void loop() {

    int myBPM = pulseSensor.getBeatsPerMinute();
    int fr = analogRead(A2);
    float fra = fr * (5.0 / 1023.0);
    int myRPM = frEstimate(fra);
```

```
float gsr = analogRead(A3);

float gsra = (gsr/1023)*5;

float tempC = mlx.readObjectTempC();

int lp = levelPainCalculate(myBPM, myRPM, gsr, tempC);


//verde
// digitalWrite(8, LOW);
// digitalWrite(9, HIGH);
// digitalWrite(10, LOW);


// delay(2000);


// //amarillo
// digitalWrite(8, HIGH);
// digitalWrite(9, LOW);
// digitalWrite(10, LOW);


// delay(2000);

// //rojo
// digitalWrite(8, LOW);
// digitalWrite(9, LOW);
// digitalWrite(10, HIGH);


// delay(2000);


unsigned long currentMillis = millis(); // Obtiene el tiempo actual

if (currentMillis - previousMillis >= interval) { // Compara si ha pasado el intervalo de tiempo
    BTSerial.print(myBPM);
    BTSerial.print(",");
}
```



```
BTSerial.print(myRPM);  
BTSerial.print(",");  
BTSerial.print(gsra);  
BTSerial.print(",");  
BTSerial.print(tempC);  
BTSerial.print(",");  
BTSerial.println(lp);  
previousMillis = currentMillis; // Actualiza el valor del tiempo anterior  
}  
}
```

int levelPainCalculate(int pulso, int frecresp, int gs, int temper){ //Lógica de implementación de mejor árbol del modelo Random Forest

```
int levelpain=0;  
if (frecresp<=22){  
if(pulso<=101){  
if(temper<=36.15){  
if(temper<=35.99){  
if(frecresp<=16){  
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;  
}else if (frecresp>16){  
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;  
}else if(temper>35.99){  
if(temper<=36.09){  
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;  
}else if(temper>36.09){  
if(gs<=0.05){  
if(temper<=36.13){  
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;  
}else if(temper>36.13){  
if(frecresp<=19){
```

```
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
    }else if(frecresp>19){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
    }
    }
    }else if(gs>0.05){
        digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
    }
    }
    }
    }else if(temper>36.15){
        digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
    }
    }
else if(pulso>101){
    if(frecresp<=18){
        if(gs<=0.06){
            if(gs<=0.04){
                if(gs<=0.03){
                    if(temper<=35.97){
                        digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
                    }else if(temper>35.97){
                        if(pulso<=184){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
                        }else if(pulso>184){
                            digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
                        }
                    }
                }else if(gs>0.03){
                    if(frecresp<=17){
```

```
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
    }else if(frecresp>17){
if(pulso<=185){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
}else if(pulso>185){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
    }
    }
    }
}else if(gs>0.04){
    if(temper<=36.09){
        digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
    }else if(36.09){
        digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
    }
    }
}else if(gs>0.06){
    if(temper<=36.13){
        if(frecresp<=12){
            digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
        }else if(frecresp>12){
            digitalWrite(3,LOW); digitalWrite(4,LOW); digitalWrite(5,HIGH); levelpain = 3;
        }
    }else if(temper>36.13){
        if(gs<=0.19){
            if(gs<=0.14){
                if(pulso<=150){
                    digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
                }else if(pulso>150){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
```



```
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
}
}
}else if(gs>0.03){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
}
}else if(gs>0.04){
if(frecresp<=20){

if(gs<=0.15){
if(temper<=36.65){
if(temper<=36.17){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
}else if(temper>36.17){
if(temper<=36.33){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
}else if(temper>36.33){
if(frecresp<=19){
if(gs<=0.1){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
}else if(gs>0.1){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
}
}else if(frecresp>19){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
}
}
}
}
}else if(temper>36.65){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
```

```
}  
}else if(gs>0.15){  
    if(pulso<=198){  
        digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;  
    }else if(pulso>198){  
        digitalWrite(3,LOW); digitalWrite(4,LOW); digitalWrite(5,HIGH);levelpain = 3;  
    }  
}  
}  
}else if(frecresp>20){  
    if(temper<=36.09){  
        digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;  
    }else if(temper>36.09){  
        digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;  
    }  
}  
}  
}  
}  
}  
}  
}  
}  
if(frecresp>22){  
    if(frecresp<=24){  
        if(temper<=36.15){  
            digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;  
        }else if(temper>36.15){  
            if(pulso<=102){  
                if(pulso<=65){  
                    digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;  
                }else if(pulso>65){  
                    if(gs<=0.03){  
                        if(temper<=36.23){
```

```
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
}else if(temper>36.23){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
}
}else if(gs>0.03){
if(temper<=36.23){
if(gs<=0.04){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
}else if(gs>0.04){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
}
}else if(temper>36.23){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
}
}
}
}else if(pulso>102){
if(temper<=36.29){
if(gs<=0.03){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;
}else if(gs>0.03){
if(temper<=36.21){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
}else if(temper>36.21){
if(gs<=0.11){
digitalWrite(3,LOW); digitalWrite(4,LOW); digitalWrite(5,HIGH);levelpain = 3;
}else if(gs>0.11){
digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,LOW); levelpain = 1;
}
}
}
}
```

```
}  
}else if(temper>36.29){  
    if(temper<=36.53){  
        digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;  
    }else if(temper>36.53){  
        digitalWrite(3,LOW); digitalWrite(4,LOW); digitalWrite(5,HIGH);levelpain = 3;  
    }  
}  
}  
}  
}  
}  
}else if(frecresp>24){  
    if(pulso<=82){  
        if(gs<=0.01){  
            digitalWrite(3,LOW); digitalWrite(4,LOW); digitalWrite(5,HIGH);levelpain = 3;  
        }else if(gs>0.01){  
            digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;  
        }  
    }else if(pulso>82){  
        if(temper<=36.29){  
            digitalWrite(3,LOW); digitalWrite(4,LOW); digitalWrite(5,HIGH);levelpain = 3;  
        }else if(temper>36.29){  
            if(pulso<=147){  
                digitalWrite(3,LOW); digitalWrite(4,HIGH); digitalWrite(5,HIGH); levelpain = 2;  
            }else if(pulso>147){  
                digitalWrite(3,LOW); digitalWrite(4,LOW); digitalWrite(5,HIGH);levelpain = 3;  
            }  
        }  
    }  
}  
}
```



```
    return levelpain;
}

int frEstimate (float fra){
    unsigned long tiempoActual = millis();

    if (fra > 2.0 && (tiempoActual - tiempoAnterior) > tiempoEspera) {
        digitalWrite(8, HIGH); // Enciende el LED indicador
        digitalWrite(9, LOW);
        digitalWrite(10, LOW);

        delay(500);           // Espera un tiempo para evitar la detección múltiple

        digitalWrite(8, LOW); // Apaga el LED indicador
        digitalWrite(9, LOW);
        digitalWrite(10, LOW);

        respiraciones++;      // Incrementa el contador de respiraciones
        tiempoAnterior = tiempoActual; // Actualiza el tiempo anterior
    }

    delay(10); // Pequeña pausa antes de leer el sensor nuevamente

    // Calcula la frecuencia respiratoria cada segundo
    if (tiempoActual % 1000 == 0) {
        frecuenciaRespiratoria = (respiraciones / ((float)tiempoActual / 1000)) * 60; // Cálculo de la
        frecuencia respiratoria en respiraciones por minuto
    }

    return frecuenciaRespiratoria;
}
```