



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

# **Sistema embebido para detección de emociones**

**TESIS**

Que para obtener el título de

**Ingeniero Eléctrico Electrónico**

**P R E S E N T A N**

Muñoz Melamed Rebeca

Ortiz Torres Daniel

Osorio Gutiérrez Samuel

**DIRECTOR DE TESIS**

Dr. Saúl de la Rosa Nieves



Ciudad Universitaria, Cd. Mx., 2021

## **Agradecimientos**

A nuestros padres quienes a lo largo de nuestras vidas nos han apoyado y motivado en nuestra formación académica, brindándonos de los valores para lograr ser buenos profesionistas y seres humanos, que además creyeron en nosotros en todo momento y que nunca dudaron de nuestras habilidades.

A nuestro asesor, quien nos guio a lo largo del presente trabajo y nos aconsejó de diversas maneras tanto de forma académica como personal.

A nuestros profesores a quienes les debemos gran parte de nuestros conocimientos, gracias a su paciencia y enseñanza.

A esta prestigiosa universidad la cual abre sus puertas a jóvenes como nosotros, preparándonos para un futuro competitivo y formándonos como profesionales con sentido de seriedad, responsabilidad y rigor académico.

¡Gracias!

## Resumen

En esta tesis se propone el desarrollo del prototipo de un detector de emociones a partir de señales fisiológicas implementado dentro de un sistema embebido. Para ello, se contempla la detección automática de cinco estados emocionales tomando como referencia el modelo dimensional de Valencia Excitación, siendo Excitación alta-Valencia positiva (EA-VP), Excitación baja-Valencia positiva (EB-VP), Excitación alta-Valencia negativa (EA-VN), Excitación baja-Valencia negativa (EB-VN) y Neutro.

El diseño creado consta de tres etapas: la adquisición de las señales fisiológicas, la extracción de características y la clasificación de la emoción. En la primera etapa, se opta por la adquisición de una señal eléctrica obtenida a partir de fotopleletismografía (PPG, por sus siglas en inglés), así como la actividad electrodérmica (EDA, por sus siglas en inglés). Una vez adquiridas, atraviesan la segunda etapa donde se les da un tratamiento específico con el propósito de obtener valores representativos de las mismas, empleando para ello distintos métodos de procesamiento digital de señales tanto en el dominio del tiempo como de la frecuencia, así como algunos métodos no lineales. Finalmente, dichos valores se introducen a una tercera etapa compuesta de algoritmos de Aprendizaje Automático, en la cual se realiza un proceso de selección de características, reducción dimensional y clasificación de emociones, empleando para ello los algoritmos denominados Selección secuencial hacia delante flotante (SFFS, por sus siglas en inglés), Discriminante lineal de Fisher (FLD, por sus siglas en inglés) y Análisis discriminante lineal (LDA, por sus siglas en inglés), respectivamente. A su vez, con el objetivo de mejorar la clasificación, se emplea un esquema de nombre Clasificación dicotómica multinivel específica a emociones (EMDC, por sus siglas en inglés). Para realizar las tareas anteriormente mencionadas, el proyecto se divide en dos elementos: un sistema embebido y un sistema en la computadora, cuya comunicación se realiza de forma inalámbrica mediante BLE. El sistema embebido se desarrolla empleando la tarjeta de desarrollo LAUNCHXL-F28379D, la cual contiene un procesador digital de señales TMS320F2837D con un procesador TMS320C28x de 32 bits. Dicho sistema realiza la adquisición y procesamiento de las señales fisiológicas, así como la implementación de la etapa de Aprendizaje Automático para la clasificación de emociones. Por otro lado, el sistema en la computadora es el encargado de realizar el entrenamiento de ésta última etapa, además de fungir como base de datos y proporcionar herramientas para la configuración del dispositivo a través de una interfaz gráfica de usuario.

Dentro de la adquisición de las señales fisiológicas fue posible comprobar la obtención adecuada de la señal proveniente de la PPG. Posteriormente, en la extracción de valores representativos, se realizó una comparación de aquellos valores adquiridos con el sistema embebido y otros utilizando software especializado, obteniendo errores por debajo del 0.02%. Para los algoritmos de selección, reducción y clasificación se obtienen exactitudes superiores al 95% empleando bases de datos ajenas a las emociones. Finalmente, se identifica el beneficio de la partición del detector de emociones en dos bloques de procesamiento, lo cual se ve reflejado al observar el tiempo de ejecución del entrenamiento del esquema EMDC en el sistema en la computadora, siendo de mínimo 4.02 s empleando un procesador de alto nivel. Por otro lado, con ayuda de los parámetros obtenidos en la etapa anterior, la simple implementación del esquema en el sistema embebido realiza la clasificación de una emoción en 2.3025 ms.

# Tabla de contenido

Agradecimientos.....	II
Resumen.....	III
Tabla de contenido.....	IV
Índice de acrónimos.....	VII
Lista de tablas.....	VIII
1 Introducción.....	1
1.1 Objetivos.....	2
1.1.1 Objetivo general.....	3
1.1.2 Objetivos específicos.....	3
2 Estado del Arte.....	4
3 Marco Teórico.....	11
3.1 Señales fisiológicas.....	11
3.1.1 Sistema Nervioso Autónomo (SNA).....	11
3.1.1.1 Sistema Nervioso Simpático (SNS).....	12
3.1.1.2 Sistema Nervioso Parasimpático (SNP).....	12
3.1.1.3 Funciones del Sistema Nervioso Autónomo.....	12
3.1.2 Electrocardiograma (ECG).....	12
3.1.3 Fotopletismografía (PPG) / Pulso de volumen de sangre (BVP).....	13
3.1.4 Actividad electrodérmica (EDA) / Respuesta Galvánica de la Piel (GSR) / Conductividad de la piel (SC).....	15
3.2 Extracción de características.....	18
3.2.1 Dominio del tiempo.....	18
3.2.1.1 Métodos estadísticos.....	18
3.2.2 Dominio de la frecuencia.....	19
3.2.2.1 Transformada rápida de Fourier.....	22
3.2.2.2 Densidad Espectral de Potencia.....	25
3.2.2.2.1 Método de Burg.....	29
3.2.2.2.2 Método de Welch.....	30
3.2.3 Atributos de las señales fisiológicas.....	32
3.2.3.1 Variabilidad de la Frecuencia Cardíaca (HRV)/ Variabilidad de la Frecuencia del Pulso (PRV).....	32
3.2.3.2 Atributos de la Respuesta Galvánica de Piel.....	35
3.3 Aprendizaje Automático en los sistemas de detección de emociones.....	36
3.3.1 Fundamentos de Aprendizaje Automático.....	36
3.3.2 Aprendizaje supervisado.....	38
3.3.3 Aprendizaje no supervisado.....	39
3.3.4 Subajuste y sobreajuste.....	40
3.3.5 Entrenamiento y validación.....	41
3.4 Algoritmos para la reducción de características.....	43

3.4.1	Selección secuencial hacia delante.....	44
3.4.2	Selección secuencial hacia delante flotante.....	46
3.4.3	Discriminante lineal de Fisher.....	46
3.4.3.1	Análisis discriminante lineal pseudoinverso.....	50
3.4.4	Algoritmos híbridos.....	50
3.5	Algoritmo de clasificación.....	51
3.5.1	Análisis discriminante lineal.....	51
3.6	Esquema EMDC.....	55
3.7	Emoción.....	56
3.7.1	Perspectivas de la emoción.....	56
3.7.2	Modelos de la emoción.....	57
3.7.3	Implicaciones de la emoción sobre los clasificadores.....	58
3.7.4	Estimuladores y evaluaciones de la emoción.....	59
4	Diseño del proyecto.....	61
4.1	Misión y visión.....	61
4.2	Desarrollo del concepto.....	62
4.2.1	Identificación de necesidades.....	62
4.2.2	Especificaciones objetivo.....	62
4.2.3	Generación de conceptos.....	63
4.2.4	Selección de concepto.....	66
4.3	Diseño en el nivel sistema.....	71
4.3.1	Configuración.....	72
4.3.2	Comunicaciones.....	73
4.3.3	Acondicionamiento de la señal.....	75
4.3.4	Extracción de características.....	75
4.3.5	Clasificación.....	76
4.3.6	Sistema en la computadora.....	77
4.3.6.1	Interfaz gráfica de usuario.....	78
4.3.6.2	Comunicación RF.....	79
4.3.6.3	Selección secuencial hacia delante flotante.....	79
4.3.6.4	Discriminante lineal de Fisher.....	80
4.3.6.5	Análisis discriminante lineal.....	81
4.4	Diseño de detalle.....	81
4.4.1	Configuración.....	81
4.4.2	Comunicaciones.....	88
4.4.2.1	Módulo HM10.....	92
4.4.2.2	Sensor biométrico AS7026GG.....	92
4.4.3	Acondicionamiento de la señal.....	97
4.4.4	Extracción de características.....	97
4.4.4.1	Generación de PRV.....	98
4.4.4.2	Obtención de SCR.....	99
4.4.4.3	Dominio del tiempo.....	99

4.4.4.4	Método no lineal de la gráfica de Poincaré.....	101
4.4.4.5	Dominio de la frecuencia.....	101
4.4.5	Clasificación.....	104
4.4.6	Selección del dispositivo comercial .....	104
4.4.7	Sistema en la computadora.....	105
4.4.7.1	Interfaz gráfica de usuario .....	105
4.4.7.2	Comunicación RF .....	107
4.4.7.3	Selección secuencial hacia delante flotante .....	108
4.4.7.4	Discriminante lineal de Fisher .....	109
4.4.7.5	Análisis discriminante lineal.....	109
5	Implementación.....	110
5.1	Configuración .....	110
5.2	Comunicaciones.....	111
5.2.1	Sensor biométrico AS7026GG.....	113
5.3	Acondicionamiento de la señal .....	115
5.4	Extracción de características.....	117
5.4.1	Generación de PRV .....	117
5.4.2	Dominio del tiempo y gráfica de Poincaré de PRV.....	117
5.4.3	Obtención de SCR.....	121
5.4.4	Dominio del tiempo de SCR .....	122
5.4.5	Dominio de la frecuencia .....	124
5.5	Clasificación .....	126
5.6	Ahorro de energía .....	128
5.7	Constitución final del sistema embebido .....	128
5.8	Sistema en la computadora .....	130
5.8.1	Interfaz Gráfica de Usuario .....	131
5.8.2	Comunicación RF.....	134
5.8.3	Selección secuencial hacia delante flotante.....	136
5.8.4	Discriminante lineal de Fisher.....	140
5.8.5	Análisis discriminante lineal .....	141
5.8.6	Esquema EMDC.....	142
6	Resultados y análisis de resultados .....	149
6.1	Comunicaciones.....	149
6.2	Acondicionamiento de la señal .....	151
6.2.1	Filtrado de la señal .....	151
6.2.2	Relación señal a ruido (SNR).....	153
6.3	Extracción de características.....	154
6.3.1	Dominio del tiempo y gráfica de Poincaré de PRV.....	154
6.3.2	Interpolación de datos PRV.....	156
6.3.3	Obtención de SCR.....	157
6.3.4	Dominio del tiempo de SCR .....	158
6.3.5	Densidad espectral de potencia .....	159
6.4	Clasificación .....	159

6.5	Sistema en la computadora .....	161
6.5.1	Comunicación RF.....	161
6.5.2	Selección secuencial hacia delante flotante.....	162
6.5.3	Discriminante lineal de Fisher.....	163
6.5.4	Algoritmo híbrido SFFS-FLD .....	165
6.5.5	Análisis discriminante lineal .....	166
6.5.6	Esquema EMDC.....	166
7	Conclusiones y trabajo a futuro.....	168
8	Referencias .....	171
9	Índice alfabético .....	178

## Índice de acrónimos

AB	AdaBoost	MD-VE	Modelo dimensional Valencia-Excitación
ADC	Convertidor analógico a digital	MLP	Perceptrón multicapa
ANOVA	Análisis de covarianza	OFE	Interfaz óptica
AR	Modelo autorregresivo	PD	Diámetro de pupila
BLE	Bluetooth de baja energía	PLDA	Análisis discriminante lineal pseudoinvertido
BR	Ritmo respiratorio	PPEM	Mapa psicofisiológico emocional
BVP	Pulso de volumen de sangre	PPG	Fotopletiografía
CART	árbol de clasificación y regresión	PRV	Variabilidad de la frecuencia del pulso
CV	Validación cruzada	PSD	Densidad espectral de potencia
DFT	Transformada discreta de Fourier	QDA	Análisis discriminante cuadrático
DT	Árbol de decisión	RESP	Respiración
EAFE	Interfaz eléctrica analógica	RF	Bosques aleatorios
ECG	Electrocardiograma	RRSV	Validación de submuestreo aleatorio repetido
EDA	Actividad electrodérmica	SAM	Maniquí de Auto Evaluación
EEG	Electroencefalograma	SAR	Registros de aproximaciones sucesivas
EMDC	Clasificación dicotómica multinivel específica a emociones	SBS	Selección secuencial hacia atrás
EMG	Electromiograma	SC	Conductividad de la piel
EOG	Electrooculograma	SCL	Nivel de conductividad de la piel
FFT	Transformada Rápida de Fourier	SCR	Respuesta de conductividad de la piel
FIFO	Cola primero en entrar-primero en salir	SFS	Selección secuencial hacia delante
FIR	Respuesta finita al impulso	SFFS	Selección secuencial hacia delante flotante
FLD	Discriminante lineal de Fisher	SKT	Temperatura de la piel
FP	Proyección Fisher	SNA	Sistema Nervioso Autónomo
GSR	Respuesta galvánica de la piel	SNP	Sistema Nervioso Parasimpático
HR	Ritmo cardíaco	SNR	Relación señal a ruido
HRV	Variabilidad de la frecuencia cardíaca	SNS	Sistema Nervioso Simpático

IAPS	Sistema internacional de imágenes afectivas	SOM	Mapa autoorganizado
IBI	Intervalo entre latido cardiacos	SVM	Máquina de vectores de soporte
IIR	Respuesta infinita al impulso	TIA	Amplificador de transimpedancia
IPP	Intervalo pulso a pulso	UART	Transmisor-receptor universal asíncrono
I2C	Circuito inter-integrado	UUID	Dirección única universal
KNN	K vecinos más cercanos	UART	Transmisor-receptor universal asíncrono
LDA	Análisis discriminante lineal	UUID	Dirección única universal
LED	Diodo emisor de luz		
LOSO	Dejar un sujeto fuera		
MAP	Máximo a posteriori		
MCU	Microcontrolador		

## Lista de tablas.

Tabla 1. Ventajas y desventajas de los métodos de audio, video y texto contra las señales fisiológicas.....	2
Tabla 2. Tabla comparativa de estudios .....	5
Tabla 3. Clasificadores con mejores resultados .....	9
Tabla 4. Mediciones electrodérmicas, definiciones y valores típicos. ....	17
Tabla 5. Desempeño de ventanas. Tabla tomada de [46] .....	21
Tabla 6. Parámetros de la HRV en el dominio del tiempo .....	33
Tabla 7. Parámetros de la HRV en el dominio de la frecuencia.....	35
Tabla 8. Características extraídas de la señal EDA.....	35
Tabla 9. Declaración de la misión .....	61
Tabla 10. Lista de necesidades involucradas con el diseño de un clasificador de emociones.....	62
Tabla 11. Métricas de las necesidades.....	63
Tabla 12. Modos de operación para el detector de emociones y la computadora externa.....	72
Tabla 13. Mapa de registros del detector de emociones.....	82
Tabla 14. Desglose de registros ilustrados en el mapa de la Tabla 13. ....	82
Tabla 15. Comandos AT para configuración del módulo HM10 .....	89
Tabla 16. Configuración temporal del secuenciador del sensor AS7026GG. ....	94
Tabla 17. Características correspondientes al dominio del tiempo para PPG. ....	100
Tabla 18. Características correspondientes al dominio del tiempo para EDA. ....	100
Tabla 19. Características y sus correspondientes ecuaciones estadísticas.....	100
Tabla 20. Propiedades del análisis en frecuencia de las señales PRV y SCR. ....	102
Tabla 21. Pseudocódigo de las funciones del bloque de configuración. ....	110
Tabla 22. Pseudocódigos de algunas ISR del subsistema de Comunicaciones. ....	112
Tabla 23. Pseudocódigo de algunas funciones utilizadas con el sensor AS7026GG. ....	114
Tabla 24. Pseudocódigo de función realizada en C para el filtrado de las señales fisiológicas. ....	116
Tabla 25. Pseudocódigo de la función realizada en ASM para ejecutar el filtro FIR en su forma directa y de fase lineal.....	117
Tabla 26. Pseudocódigo de las funciones para generar la PRV y calcular sus características .....	118
Tabla 27. Pseudocódigo para la obtención de la componente SCR de la señal EDA. ....	121
Tabla 28. Pseudocódigo de la función realizada en ASM para ejecutar la convolución de la señal diferenciada con la ventana Bartlett. ....	121
Tabla 29. Pseudocódigo para la obtención de las características provenientes de los SCR .....	122

Tabla 30. Pseudocódigo del método de interpolación por trazadores cúbicos. ....	124
Tabla 31. Pseudocódigo de la función para extracción de características en el dominio de la frecuencia..	126
Tabla 32. Pseudocódigo de las funciones del subsistema de clasificación.....	127
Tabla 33. Costo y dimensiones de los elementos del sistema embebido .....	128
Tabla 34. Pseudocódigo de los métodos de la clase RF_COMS para la configuración de registros del detector de emociones. ....	135
Tabla 35. Pseudocódigo de la función SFFS con validación simple.....	137
Tabla 36. Pseudocódigo de la función SFFS con validación cruzada dejando uno fuera. ....	139
Tabla 37. Pseudocódigo de la función FLD. ....	140
Tabla 38. Pseudocódigo de la función LDA. ....	141
Tabla 39. Formato de archivo “features.csv” .....	143
Tabla 40. Formato de archivo “Emotions.csv” .....	143
Tabla 41. Pseudocódigo de la función EMDC en la modalidad de entrenamiento rápido.....	143
Tabla 42. Pseudocódigo de la función EMDC en la modalidad de entrenamiento completo. ....	146
Tabla 43. Pseudocódigo de la función EMDC. ....	148
Tabla 44. IPP y características obtenidas a partir de MATLAB. ....	154
Tabla 45. Amplitudes y características obtenidas a partir de MATLAB. ....	158
Tabla 46. Características particulares del detector de emociones de esta tesis. ....	168

# 1 Introducción

Las emociones son aquellas sensaciones difusas y complejas que el ser humano suele experimentar en cada instante de su vida, estas lo motivan a empezar un nuevo día, ponerlo en un estado de alerta en caso de estar en peligro, provocarle conductas violentas que pueden afectar su entorno, o bien incapacitarlo mentalmente para realizar alguna de sus actividades diarias tras experimentar una pérdida o evento de alto impacto. Así pues, se entiende la gran influencia que poseen las emociones sobre la toma de decisiones de las personas, ya que, así como existen ocasiones en las que el individuo es capaz de identificar la emoción, lo cual le permite tomar acción respecto a una determinada circunstancia, en otras sucede que el individuo es incapaz de percibir la emoción que está experimentando, o bien no sabe la manera en cómo responder a la misma. Por esta razón, en la actualidad existe una tendencia en los campos de investigación relacionados con la psicología, psiquiatría, ciencia conductual y neurociencia hacia el desarrollo de sistemas autónomos que identifiquen los estados emocionales de los usuarios para anticipar las necesidades o cuidados que éstos requieran, a fin de asistirlos en su vida diaria. Por ende, el área del conocimiento que tiene por objetivo el desarrollo de estos sistemas capaces de reconocer, interpretar, procesar e incluso simular las emociones humanas recibe el nombre de Computación Afectiva (Affective Computing), término presentado en el libro “*Affective Computing*” de Rosalind W. Picard [1].

La mayoría de estos sistemas parten de la teoría propuesta por James-Lange [2] que dicta que la evocación de una emoción prosigue a las respuestas fisiológicas que experimenta el Sistema Nervioso Simpático. A pesar de no ser aceptada por toda la comunidad científica en el área de la computación afectiva, debido a que existen discusiones respecto a su universalidad en contextos sociales y experiencias personales, esta teoría ha terminado por otorgar resultados satisfactorios en la práctica.

Dentro de esta área, se han realizado diversos trabajos cuya aplicación se ha visto orientada a los sectores de la salud, seguridad, educación, marketing, entre otros. Algunas de las aplicaciones donde es posible la implementación de estos dispositivos puede ser:

- La telemedicina para diagnosticar pacientes sin necesidad del contacto directo con médicos.
- Tratamiento de fobias.
- Detección temprana y/o el apoyo en el tratamiento de personas con desórdenes psicológicos como ansiedad o depresión.
- Apoyo en la toma de decisiones bajo situaciones de peligro o crisis.
- Manejo de ataques de rabia o frustración en conductores.
- Asistencia inteligente en investigaciones criminales.
- Apoyo en el aprendizaje mediante el monitoreo de estados de frustración y/o ansiedad en alumnos para otorgar estímulos motivacionales que los contrarresten.
- Identificación de patrones en imágenes o vídeos publicitarios que generen mayor estimulación sobre los compradores.

Para llevar a cabo estas aplicaciones, los investigadores han optado por emplear técnicas audiovisuales, de escritura o técnicas basadas en señales fisiológicas que permiten reconocer la evocación de emociones en una persona. En el primer caso, podemos mencionar el uso de cámaras de video para capturar características visuales como el lenguaje corporal, tonalidad de la piel y gestos faciales, o bien micrófonos para identificar estructuras del lenguaje, variaciones en la entonación y volumen de voz. Respecto a las técnicas de escritura, la más popular consiste en la identificación de palabras clave en la redacción de un párrafo. Por otra parte, dentro de las técnicas que emplean señales fisiológicas se encuentran el electrocardiograma (ECG, por sus siglas en inglés), electroencefalograma (EEG, por sus siglas en inglés), fotopletismografía (PPG, por sus siglas en inglés), respuesta galvánica de la piel (GSR, por sus siglas en inglés), electromiograma (EMG, por sus siglas en inglés), respiración (RESP, por sus siglas en inglés), temperatura de la piel (SKT, por sus siglas en inglés) y electrooculograma (EOG, por sus siglas en inglés).

De las técnicas anteriores suelen emplearse un conjunto de ellas a fin de conformar lo que se conoce como un sistema multimodal que permite aumentar la efectividad del sistema. Entre cada una de estas técnicas es posible distinguir ciertas ventajas y desventajas enlistadas en la Tabla 1.

Tabla 1. Ventajas y desventajas de los métodos de audio, video y texto contra las señales fisiológicas.

<b>Audiovisuales y Texto</b>	<b>Señales fisiológicas</b>
<p><b>Ventajas</b></p> <ul style="list-style-type: none"> <li>• Los sistemas de monitoreo pueden resultar poco invasivos. Entendiendo por invasivo el no procurar la comodidad de uso para el usuario.</li> </ul>	<p><b>Ventajas</b></p> <ul style="list-style-type: none"> <li>• No son posibles de ocultarse o manipularse por el individuo.</li> <li>• Emplean periodos medianos de monitoreo para la detección de la emoción.</li> <li>• Permiten monitorear constantemente al individuo.</li> </ul>
<p><b>Desventajas</b></p> <ul style="list-style-type: none"> <li>• Permiten al individuo disimular o manipular sus respuestas corporales (reacciones, tono de voz, gestos, caligrafía).</li> <li>• Requieren de largos periodos de monitoreo para la detección de la emoción.</li> </ul>	<p><b>Desventajas</b></p> <ul style="list-style-type: none"> <li>• Les resulta difícil distinguir patrones fisiológicos similares en emociones distintas, ej. Miedo y Excitación ocasionan un aumento en el ritmo cardiaco.</li> <li>• La mayoría de los sistemas de monitoreo suelen resultar muy invasivos. Existen excepciones como la PPG.</li> <li>• Los sensores que hacen adquisición de ellas resultan sensibles a movimientos corporales pronunciados.</li> </ul>

Entre los anteriores, las técnicas que emplean señales fisiológicas resultan las más convenientes para la elaboración de dispositivos ergonómicos. Estos sistemas suelen conformarse por diversas etapas, una para la adquisición de las señales fisiológicas deseadas, acompañado de otra para su respectivo acondicionamiento, seguido de una etapa para el procesamiento digital dentro de la cual se obtienen parámetros representativos de las señales, y, finalmente, se implementa una última para la clasificación que asigna los parámetros mencionados a una emoción en específico.

No obstante, dentro de los trabajos realizados hasta el momento se han empleado procesadores de alto desempeño, normalmente con el uso de computadoras, que aprovechan su potencia de cómputo y uso de software especializado para implementar la mayoría de las etapas que conforman a un detector de emociones. Sin embargo, esto conlleva una carencia en practicidad para la realización de aplicaciones portátiles que solicitan un volumen reducido para su uso ergonómico en los usuarios. Por esta razón, en el siguiente trabajo se generan los módulos que permiten conformar a un sistema detector de emociones, basado en la técnica que emplea señales fisiológicas, integrando las etapas fundamentales anteriormente mencionadas dentro de un sistema embebido y dejando para futuros trabajos su caracterización en conjunto.

La tesis estará estructurada de la siguiente forma, en el Capítulo 2 se presenta el estado del arte del campo de la computación afectiva, enfocados primordialmente a las últimas técnicas empleadas con señales fisiológicas, para describir los métodos y tecnologías de vanguardia que se encuentran en desarrollo en este campo. El Capítulo 3 da una descripción detallada de los conceptos más importantes identificados en los artículos del capítulo previo, haciendo énfasis en su aplicación y utilidad para la detección de emociones. El Capítulo 4 describe el diseño del proyecto del cual se establece su misión y visión, el desarrollo del concepto a trabajar, su diseño a nivel sistema y su diseño a detalle. El Capítulo 5 compete a la implementación del detector de emociones, exponiendo los elementos conformados y describiendo los programas que comprenden a los diversos subsistemas del sistema embebido y del sistema en la computadora, así como los ajustes que fue necesario realizar en los elementos del diseño a detalle a causa de la obtención de resultados indeseados durante su implementación y puesta a prueba. El Capítulo 6 expone los resultados obtenidos por cada uno los sistemas y subsistemas diseñados, junto con el análisis de su efectividad en términos de funcionamiento. En el Capítulo 7 se presentan las conclusiones del trabajo junto con las observaciones y sugerencias que se tienen del mismo para aquellos individuos interesados en darle seguimiento o uso para sus propias aplicaciones.

## 1.1 Objetivos

A continuación, se presentan de forma resumida los objetivos de la presente tesis. Para ello se realiza una división de éstos en dos secciones, el Objetivo general y los Objetivos específicos, lo cual permite dar un panorama general del trabajo y, posteriormente, particularizarlo.

### **1.1.1 Objetivo general**

- Desarrollar la propuesta de un detector de emociones a partir de señales fisiológicas implementado dentro de un sistema embebido.

### **1.1.2 Objetivos específicos**

- Que el sistema desarrollado sea capaz de identificar cinco estados emocionales tomando como referencia el modelo dimensional de Valencia Excitación.
- Emplear elementos que permitan un diseño ergonómico.
- Evaluar el funcionamiento individual de cada componente del sistema desarrollado.
- Aportar las herramientas suficientes para que el prototipo pueda ser empleado en un estudio que permita validar su eficacia en la detección de emociones, así como brindar las bases para el desarrollo de dispositivos futuros.

## 2 Estado del Arte

Con el objetivo de identificar la estructura de un detector de emociones con el enfoque propuesto en la presente tesis, se analizaron 18 trabajos recientes relacionados al área de la Computación Afectiva, los cuales se centran en la detección de emociones a través del uso de señales fisiológicas. Por medio de la Tabla 2 se simplifica cada uno de los estudios dividiéndolos en apartados que permiten diferenciar sus principales rasgos. Dichos apartados son: los parámetros adquiridos de la(s) señal(es), el método de clasificación utilizado para reconocer las emociones, el número de categorías que se consideran en la clasificación, verificando de esta forma el modelo a seguir para identificar las emociones, la rúbrica bajo la cual se evalúan los resultados de la clasificación, la metodología para llevar a cabo el estudio, así como los elementos empleados para evocar las emociones en los individuos. Al final de esta sección se distinguen aquellas cualidades que implican un mejor rendimiento en la detección de emociones lo cual servirá como referencia para el planteamiento de este trabajo.

Tabla 2. Tabla comparativa de estudios

Estudio	Señal	Características extraídas	Clasificación	Estimulador	Descripción	Resultados
[3] (2001)	EMG, PPG, GSR, BR	En todas las señales se obtuvo promedio, desviación estándar, Prom. Abs. Primera diferencia, Prom. Abs. Primera diferencia, normalizada, Prom. Abs. Segunda diferencia, Prom. Abs. Segunda diferencia normalizada y otras 10 características.	<b>Clasificador:</b> KNN, MAP (Clasificación multi-clase) <b>Selector:</b> SFFS <b>Reductor:</b> FP <b>Selector/Reductor:</b> SFFS-FP	Sistema de estimulación "Sentograph"	Experimento sobre una sola persona con el sistema "Sentograph" durante varias semanas para clasificar ocho estados emocionales distintos.	MAP (SFFS-FP): 81.25% MAP (Fisher): 77.5% KNN (SFFS): 65%
[4] (2005)	EMG, ECG, GSR, RSP	Amplitud y frecuencia de señales, prom, $\sigma$ y otras 32 características.	<b>Clasificador:</b> KNN, LDF/LDA, MLP (Clasificación binaria y multiclase) <b>Selector:</b> ANOVA, SFS <b>Reductor:</b> Fisher <b>Validación:</b> Validación cruzada dejando uno fuera	Canciones	Experimento con una persona que consistía en escuchar canciones que provocaran colocarlo en cada uno de los cuatro estados emocionales a estudiar (alegría, ira, tristeza, placer) mediante el modelo MD-VE.	<b>4 emociones:</b> LDF/LDA (SFS): 92.05% 5NN (SFS/Fisher): 90.91% MLP6 (SFS/Fisher): 88.64% <b>Binario (Valencia):</b> LDF/LDA (SFS): 86.36% 5NN (SFS/Fisher): 86.36% MLP6 (SFS/Fisher): 88.64% <b>Binario (Excitación):</b> LDF/LDA (SFS): 96.59% 5NN (SFS/Fisher): 94.32% MLP6 (SFS/Fisher): 94.32%
[5] (2005)	HRV, GSR	SDNN, RMSSD, HR, LF/HF y GSR.	<b>Clasificador:</b> MLP (Clasificación multiclase)	Videos	Experimento con 6 personas mostrando videos como estímulo, para inducir 4 emociones (tristeza, placer tranquilo, placer interesante y miedo) representadas en el MD-VE.	Exactitud: 80.2%
[6] (2006), [7] (2007)	HRraw, HRV, GSRraw, SCR, SCL	GSR prom, GSR amplitud máxima, SCR RelativeNb, Tiempo de partida, Prom Tiempo Rizo, Prom Amplitud, SCL OnsetOffsetDiff, HR Prom, HR Min, HR Max, Prom Ei, Min Ei, Max Ei, Derivada Prom Ei, Potencia Relativa MF, Variabilidad Total, Balance Simpatovagal.	PPEM	Videos e imágenes	Experimento con 40 sujetos de prueba considerando 5 regiones del MD-VE.	No se proporcionan resultados
[8] (2008)	ECG, RESP, SC, EMG	Excitación: ECG_sd2_PoincareHRV, RESP_Energiaprom_SubEspectro, RESP_sd2_PoincareBRV, RESP_prom_MSE, Valencia: ECG_sd2_PoincareHRV, RESP_Energiaprom_SubEspectro, RESP_sd2_PoincareBRV, RESP_prom_MSE 4 emociones: ECG_sd2_Poincare_HRV, RESP_EnergiaProm_SubEspectro, RESP_sd2_PoincareBRV, RESP_prom_MSE	<b>Clasificador:</b> Distancias euclidianas (Clasificación binaria y multiclase) <b>Reductor:</b> pLDA <b>Selector:</b> SBS <b>Esquema:</b> EMDC <b>Validación:</b> Validación cruzada dejando uno fuera	Canciones elegidas por los participantes.	Experimento con 3 sujetos varones entre 25-38 años, considerando 4 regiones (EQ1, EQ2, EQ3 y EQ4) del MD-VE.	<b>LDA:</b> Clasif. Excitación: 90% Clasif. Valencia: 98% <b>4 clases:</b> Clasif. dependiente: 87% Clasif. independiente: 65% <b>EMDC:</b> Clasif. dependiente: 95% Clasif. independiente: 70%
[9] (2012)	GSR, PPG, RESP, SKT, EMG, EOG, EEG	En total son 106 características de las que destacan: GSR_prom, GSR_min_local, tiempo_levantamiento, PSD: (0-2.4 Hz) SCSR, SCVSR, SCSR y SCVSR_prom_magnitud_picos., RESP_prom, prom_derivada, RESP_ $\sigma$ , rango, RESP_ritmo, HR_prom, HR_ $\sigma$ , HRV, IBI, PSD: LF, MF, HF, entre otros.	<b>Clasificador:</b> Naïve Bayes <b>Selector:</b> a través de los valores propios del discriminante lineal de Fisher	40 videos	Experimento con 32 sujetos (16 mujeres), entre 19-37 años, considerando 3 regiones (valencia, excitación y dominancia)	<b>Clasif. Excitación (F1):</b> EEG 58.3%, Fisiológico 53.3%, MCA 61.8% <b>Clasif. Valencia (F1):</b> EEG 56.3%, Fisiológico 60.8%, MCA 60.5%

			<b>Validación:</b> Validación cruzada dejando uno fuera			
[10] (2012)	HR	Oscilación local dominante, función modal intrínseca (IMF)	<b>Clasificador:</b> LDA (Clasificación binaria y multiclase) <b>Validación:</b> Validación cruzada dejando uno fuera	Imágenes del IAPS	Experimento con 44 sujetos considerando 9 combinaciones de 5 estados emocionales (Erótica, Excitante, Miedo, Disgusto y Horror)	<b>Clasif. Excitación:</b> En 3 niveles 35.36% En 2 niveles 76.19% <b>Clasif. Valencia:</b> exactitudes menores al 70%
[11] (2013)	HR	HR prom, HR $\sigma$ , HRV prom, HRV $\sigma$ , PSD LF, PSD HF, PSD LF/HF, invariantes biespectrales, magnitud media del biespectro, fase de la entropía, entropía biespectral normalizada y raíz de la entropía biespectral normalizada	<b>Clasificador:</b> SVM (Clasificación binaria) <b>Validación:</b> Validación cruzada de 4 dobleces	20 imágenes del IAPS	Experimento con 4 sujetos entre 21-24 años, considerando 2 regiones (neuro y excitado) del MD-VE.	Emociones positivas-negativas: >58% Clasif. Excitación: >78% Clasif. Valencia: >68%
[12] (2014)	EDA, ECG, SKT, PPG	SCL, NSCR, meanSCR, HR, meanRRI, SD_RRI, RMSSD_RRI, NN50, pNN50, SDI, SD2, CSI, CVI, RRtri, TINN, LF, HF, HRV, RSA, Resp_R, Resp_A, Resp_Rsd, Resp_Asd, meanSKT, maxSKT, BVP y PPT.	<b>Clasificador:</b> LDA, Naive Bayes, CART, SVM (Clasificación multiclase) <b>Selector:</b> ANOVA <b>Validación:</b> CV y RRSV.	Videos	Experimento con 300 participantes, considerando 6 emociones (alegría, tristeza, enojo, miedo, sorpresa y neutral).	Naive Bayes: 53.9% LDA: 52.7% CART:42.7% SVM:39.2%
[13] (2015)	ECG	meanRR, CVRR, SDRR, SDSD, LF, HF, LF/HF, Coeficiente de curtosis y Skewness.	<b>Clasificador:</b> Algoritmo de separación simple (Clasificación multiclase)	Videos	Experimento con 21 participantes, considerando 4 emociones (tristeza, enojo, miedo y felicidad) para clasificarlas.	Exactitud máxima: 50%
[14] (2015)	ECG	RR prom, RR std, SDNNm RMSSD, pNN50, ind. tr. HRV, TINN, pot VLF/LF/HF, VLF/LF/HF pico, %pot VLF/LF/HF, pot nu LF/HF, relación LF/HF, ApEn, DFA, LPP.	<b>Clasificador:</b> QDA (Clasificación binaria y multiclase) <b>Validación:</b> LOSO	Sonidos	Experimento con 27 sujetos, empleando el MD-VE, con 4 niveles de excitación y dos niveles de valencia.	Exactitud excitación: 84.26% Exactitud valencia: 84.72%
[15] (2016)	IBI, RESP	pNN50, Welch PSD: LF/HF, Burg PSD: LF/HF, Lomb-Scargle PSD: LF/HF, SD <sub>2</sub> , SD <sub>2</sub> /SD <sub>1</sub> , SampEn <sub>1</sub> , SampEn <sub>2</sub> , DFAall	<b>Clasificador:</b> SVM (Clasificación multiclase) <b>Validación:</b> Validación cruzada dejando uno fuera	Elementos propios de los participantes	Experimento con 12 sujetos (6 M) entre los cuales 6 (3 M) eran actores con 3-7 años de experiencia, considerando 4 regiones (Tristeza, Enojo, Placer y Alegría), del MD-VE y un estado neutro	<b>5 clases:</b> Clasif. independiente: 72.3% Clasif. dependiente: 87.0%
[16] (2016)	SCL, SCR, BVP, HR, SKT, PD	Promedio y varianza.	<b>Clasificador:</b> SVM (Clasificación binaria) <b>Validación:</b> Validación cruzada de 4 dobleces	Comics	Se obtuvo los datos de 5 personas al leer un capítulo de un comic aleatorio. Se consideraron dos regiones del MD-VE (baja y alta excitación)	No brinda resultados numéricos.
[17] (2017)	HR, GSRr, RESP, BR, TEMP, HRV	Máximos, Mediana, Desviación estándar y otras 169 características.	<b>Clasificador:</b> SVM <b>Método de fusión de</b>	Base de datos MACHNOB-HCI	Uso de datos del MACHNOB-HCI trabajado sobre 24 sujetos, considerando 3 regiones (calmado, medianamente excitado y excitado) del MD-VE.	Clasif. Excitación: 54.73% Clasif. Valencia: 56.83%

			<b>características:</b> Level Feature Fusion			
[18] (2017)	EDA, ECG, SKT, PPG	SCR, NSCR, meanSKT, maxSKT, meanPPG, meanRR, STD, meanHR, RMSSD, NN50, pNN50, SD1, SD2, CVI, CSI, RR tri index, TINN, FFTap_LF, FFTap_HF, Arap_LF, Arap_HF, FFTnLF, FFTnHF, FFTL/Hratio, ARnLF, ARnHF, AR LF/HF ratio.	<b>Clasificador:</b> CART, kNN, SOM, Naïve Bayes, LDA, SVM.	Videos	Experimento con 20 estudiantes entre 19-25 años, mostrando videoclips como estímulo para inducir dos emociones negativas para clasificarlas mediante diversos algoritmos de machine learning.	LDA: 59.8% CART: 58.6% Naive Bayes: 55.4% KNN: 54.3% SOM: 53.7% SVM: 52.4%
[19] (2018)	BVP, ECG, SCR, SCL, EMG, RESP, SKT, ACC	ACC prom, ACC $\sigma$ , ACC picos de frecuencia, ECG/BVP PSD: ULF/LF/HLF/UHF, EDA prom, EDA $\sigma$ , EDA intervalo dinámico, EMG prom, EMG $\sigma$ , EMG PSD: ULF/LF/HLF/UHF, RESP max, RESP min, RESP prom, RESP $\sigma$ , TEMP prom, TEMP $\sigma$ , TEMP min, TEMP max, entre otras.	<b>Clasificador:</b> DT, RF, AB, LDA, KNN (Clasificación binaria y multiclase) <b>Validación:</b> CV y LOSO	Lecturas, videoclips, presentaciones públicas y tareas aritméticas.	Experimento que consistió en el análisis de 15 sujetos ante distintos estímulos con el objetivo de inducir tres estados afectivos (neutral, estrés, diversión) y posteriormente uno binario (estrés y no estrés).	<b>Tres estados afectivos:</b> AB: 73.62% (Muñeca), 80.34% (Pecho) LDA: 68.68% (Muñeca), 79.35% (Pecho) RF: 76.17% (Muñeca), 71.37% (Pecho) DT: 63.34% (Muñeca), 58.62% (Pecho) KNN: 58.54% (Muñeca), 57.31% (Pecho)  <b>Caso binario:</b> LDA: 86.46% (Muñeca), 93.12% (Pecho) AB: 88.05% (Muñeca), 89.76% (Pecho) RF: 84.88% (Muñeca), 84.18% (Pecho) KNN: 81.96% (Muñeca), 81.05% (Pecho) DT: 63.34% (Muñeca), 58.62% (Pecho)
[20] (2019)	IBI-ECG, IBI-PPG.	HF, LF, VLF, LF/HF, media, mediana, SDSD, NN20, pNN20, rMSSD, Entropía multiescala	<b>Clasificador:</b> Red Neuronal Profunda Bayesiana (Clasificación multiclase)	Videos	Experimento en 17 personas sometidas a vídeos cortos donde se usó el MD-VE y se clasificó en 5 regiones de valencia.	F1: 0.70

Haciendo un análisis de la Tabla 2 es posible identificar los sistemas con mejor desempeño a la hora de reconocer emociones, para esto se consideró aquellos que hayan obtenido exactitudes superiores al 80%. Cabe aclarar que las condiciones bajo las cuales se llevó a cabo cada estudio fueron distintas, ya sea por el número de participantes, la estimulación empleada o bien, la metodología en general, por lo que los resultados se presentarán tomando en cuenta distintas categorías para remarcar esta diferencia. A pesar de ello, gran parte de los estudios exitosos presentaron rasgos en común en determinados aspectos, por lo que es pertinente enlistarlos.

- **Señales fisiológicas y técnicas para obtenerlas:** las señales y técnicas con mejor desempeño para realizar la identificación de emociones son aquellas provenientes del corazón y la respiración obtenidas por medio del electrocardiograma (ECG) o fotopleletismografía (PPG). También es posible adquirir una señal que represente la respiración por medio de un acelerómetro en el pecho. Del mismo modo, son de gran importancia tanto la actividad electrodérmica (EDA) como aquellas señales provenientes de los músculos obtenidas por medio del electromiograma (EMG). En la sección 3.1 correspondiente al Marco Teórico se presentan con mayor detalle la descripción de las señales y técnicas empleadas en este trabajo.
- **Características obtenidas de las señales fisiológicas:** se entiende por “características” como aquellos datos representativos recabados de las señales por medio de técnicas de procesamiento digital de señales. En el caso de la clasificación de emociones, las características con el mejor rendimiento fueron aquellas obtenidas por medio de técnicas en el dominio del tiempo, la frecuencia, y métodos no lineales. Los artículos utilizados para dar esta valoración son [3], [4], [5], [8], [14], [15] y [19]. En la sección 3.2 se presentan con mayor detalle las características específicas para cada señal fisiológica considerada en este trabajo y la forma de obtenerlas.
- **Reducción de características:** en el Estado del Arte existen sistemas que implementan un algoritmo para reducir el número de características antes de ingresarlas a un clasificador. La Selección secuencial hacia delante flotante en conjunto con la Proyección Fisher (SFFS-FP, por sus siglas en inglés) incrementaron el desempeño del sistema clasificador de emociones. De igual manera, se utilizó una variante de la FP denominada pLDA. Los artículos utilizados para dar esta valoración son [3], [4] y [8]. Estos algoritmos son descritos a detalle en la sección 3.4.
- **Modelo de emociones:** un “modelo de emoción” es la forma de representar las emociones que están presentes en el ser humano. En la mayoría de los sistemas (incluyendo aquellos con mejor desempeño) se optó por el modelo dimensional de Valencia-Excitación. De la misma forma, aunque menos común, hubo casos que optaron por un modelo discreto. Estos modelos se presentan con mayor detalle en la sección 3.7.2 del Marco Teórico.
- **Estimulador:** se entiende por “estimuladores” a aquellos medios utilizados para evocar una emoción en el ser humano. En los estudios se utilizaron estimuladores de distintas clases, los cuales fueron propios de la investigación, o bien, se obtuvieron a partir de bases de datos ya existentes. Al visualizar los resultados se observó que los más utilizados fueron videos, sin embargo, aquellos estimuladores que tuvieron mejores resultados de forma global son aquellos donde se solicitaba al participante elegir una serie de objetos o canciones que le permitieran evocar alguna emoción. Los artículos utilizados para dar esta valoración son [4], [8] y [15]. En la sección 3.7.4 se presentan con mayor detalle los estimuladores descritos.
- **Validación:** la “validación” es una técnica propia de los algoritmos de clasificación que será abordada con mayor detalle dentro de la sección 3.3.5. El tipo de validación más recurrente en los estudios fue la Cruzada dejando uno afuera.

Como se mencionó, existe una diferencia en la metodología de cada uno de los estudios realizados para desarrollar el clasificador de emociones, sin embargo, el punto más diferencial es el tipo de clasificación, tal como puede verse a continuación:

- **Clasificador dependiente:** dentro de las investigaciones donde se analizaron datos de una sola persona se puede observar que los algoritmos LDA, 5NN y MLP6 obtuvieron los mejores resultados al momento de clasificar sus emociones. De igual forma, aunque no sea un clasificador, resulta importante destacar que el esquema EMDC fue bastante exitoso en sus pruebas utilizando el reductor pLDA y una clasificación por medio de las distancias euclidianas de las muestras y el centroide de las clases. Los artículos utilizados para dar esta valoración son [4] y [8].
- **Clasificador independiente:** en estos clasificadores se considera una universalidad de las emociones, implicando el análisis datos a partir de muestras recabadas con un gran número de participantes. Los mejores algoritmos fueron seleccionados tomando en cuenta tanto su exactitud como el número de participantes en el que fue implicado, siendo los clasificadores QDA y AB los más eficientes, seguidos de MLP, que si bien, cuenta con un menor número de estudiados, siguen destacando por el número de emociones que busca clasificar. Los artículos utilizados para dar esta valoración son [5], [14] y [19].
- **Clasificador con mayor número de emociones:** el clasificador más sobresaliente en este campo fue MAP pues considera la obtención de ocho emociones. El artículo utilizado para dar esta valoración es [3].
- **Clasificadores binarios:** son aquellos clasificadores que únicamente distinguen entre dos estados emocionales. Los mejores clasificadores dentro de este campo fueron LDA, RF y AB. El artículo utilizado para dar esta valoración es [19].

En la Tabla 3 se muestra un resumen de los algoritmos con mejor desempeño en la clasificación de emociones, así como las condiciones bajo las que fueron implementados.

Tabla 3. Clasificadores con mejores resultados

	Clasificador	Reductor	Número de emociones/regiones	Número de participantes	Exactitud
<b>Clasificador dependiente</b>	Distancias euclidianas usando esquema EMDC	pLDA	4	1	95%
	LDA	SFS	4	1	92.05%
	5NN	SFS-FP	4	1	90.91%
	MLP6	SFS-FP	4	1	88.65%
	SVM		4	1	87%
<b>Clasificador independiente</b>	QDA		4	27	84.26%
	AB		4	15	80.34%
	MLP		4	6	80.2%
<b>Clasificador con mayor número de emociones</b>	MAP	SFFS-FP	8	1	81.25% (dependiente)
<b>Clasificador binario</b>	LDA		2	15	93.12%
	RF		2	15	92.01%
	AB		2	15	89.76%

Dados los motivos que se expondrán en la sección 4.2.4, en la presente tesis únicamente se aborda el algoritmo de clasificación LDA, así como la estructura del esquema EMDC, mismos que se encuentran descritos dentro de la sección 3.5.

Para realizar un sistema detector de emociones es importante elegir un diseño que se adapte a las necesidades de la aplicación buscada. Dentro del Estado del Arte hubo un gran número de propuestas de diseño que, a pesar de perseguir una detección precisa y exacta de determinados estados de ánimo, tomaban caminos distintos dependiendo de su objetivo, tal como se muestra en la Tabla 3. En su mayoría, cada estudio aportó la exactitud de sus predicciones, lo cual ayuda en gran medida a averiguar qué tipo de diseños son más efectivos que otros en determinados campos, por lo que para realizar una propuesta es necesario definir en primer lugar el tipo de clasificador de emociones a desarrollar (si será dependiente, independiente, binario, etc.) y el número de estados emocionales que se busca detectar. Tomando esto en consideración y utilizando como respaldo los datos recabados, se puede sentar la base sobre la que se construirá la propuesta de diseño, en la que debe considerarse el tipo de sensores a utilizar, la interfaz entre el dispositivo y el usuario, el hardware de procesamiento, los algoritmos implementados para la manipulación de señales y detección de emociones, etc.

En un sistema detector de emociones basado en la adquisición de señales fisiológicas será necesario utilizar la mayoría, o un número considerable, de las señales empleadas en los diseños con mejor desempeño, tales como el ECG, GSR, RESP, EMG y PPG, de las cuales serán extraídas determinadas características a través de procesamientos digitales en el tiempo, la frecuencia, así como métodos estadísticos y no lineales. La selección de señales también dependerá en gran medida del grado de ergonomía buscado, pues mientras menor sea la cantidad de sensores implementados, más pequeño y cómodo será el detector. Además, hay que considerar que existen señales como el ECG y EMG que requieren de sensores bastante incómodos.

Después de la etapa de extracción de características es posible utilizar un algoritmo que realice una compresión y/o selección de los datos de modo que se disminuya la carga computacional del algoritmo de clasificación, sin embargo, su utilización será elección del diseñador, dado que la efectividad de dicho algoritmo dependerá del número de características extraídas de las señales fisiológicas, pues si se dispone de un número reducido de éstas es posible que dicho aditamento no sea necesario.

El algoritmo de clasificación deberá ser seleccionado de acuerdo con el tipo de clasificador requerido en el sistema. Esto puede verse más a detalle en la Tabla 3, la cual muestra los algoritmos con mejor desempeño en cada tipo de aplicación.

Finalmente, con el objetivo de evocar las emociones con mayor fidelidad en el o los participantes, será necesario el uso de objetos personales que los vinculen con los estados emocionales que se busca detectar. Además, el número de personas por evaluar será determinado por la dependencia o independencia del sistema, pues en el caso de un sistema dependiente será necesario contar con los datos de una sola persona y en el caso independiente habrá que maximizar el número de participantes. Esta diferencia es importante ya que definirá el algoritmo más apto para la clasificación. Asimismo, tanto el uso del modelo dimensional como del modelo discreto son buena opción para la detección. Los temas relacionados a la evocación de emociones se presentan en la sección 3.7 del Marco Teórico.

### 3 Marco Teórico

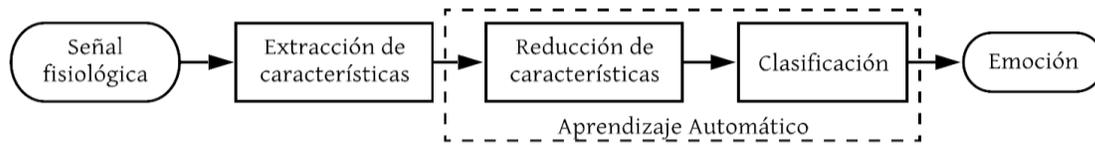
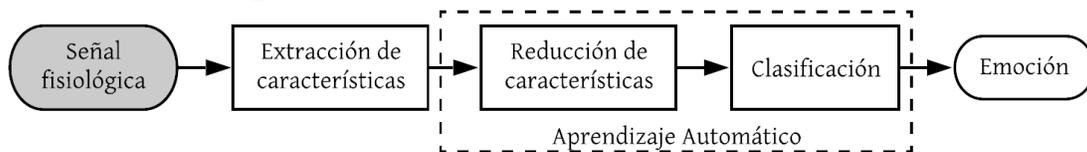


Figura 1. Etapas que conforman a un sistema detector de emociones.

Para el desarrollo de este capítulo se emplea el modelo de la Figura 1 con el objetivo de describir cada una de las etapas de un detector de emociones, las cuales son abordadas a continuación. Este es el esquema bajo el cual se rige la mayoría de los sistemas detectores de emociones analizados en el Estado del Arte. La sección 3.1 abarca la primera etapa del modelo, en la cual se realiza una descripción a detalle de las señales fisiológicas que, con base en el análisis del Estado del Arte, demostraron aportar información relevante acerca del estado afectivo de los participantes. En seguida, la fase de extracción de características se aborda en la sección 3.2, donde se presentan los fundamentos matemáticos empleados en la obtención de estimaciones, las cuales otorgan valores representativos de las señales fisiológicas analizadas. El bloque de Aprendizaje Automático abarca las secciones 3.3, 3.4 y 3.5, en donde se introducen los fundamentos teóricos del aprendizaje automático, los algoritmos dedicados a la reducción de características y los algoritmos de clasificación, respectivamente. Por último, el bloque de emoción se desarrolla en la sección 3.7, donde se exponen los modelos y perspectivas bajo las cuales se analizan las emociones a clasificar.

#### 3.1 Señales fisiológicas



Los estudios analizados en el Estado del Arte demostraron la relación que mantienen los estados afectivos con los cambios fisiológicos que experimenta el cuerpo humano, donde dichos cambios, a su vez, se encuentran estrechamente ligados al Sistema Nervioso Autónomo (SNA). A continuación, se describe la estructura y funciones del SNA que justifican la implementación de señales fisiológicas en la detección de emociones.

##### 3.1.1 Sistema Nervioso Autónomo (SNA)

El SNA es la parte del sistema nervioso periférico encargada de regular las funciones involuntarias del organismo. Es responsable de inervar los órganos internos como lo son las glándulas sudoríparas, el corazón, los vasos sanguíneos, el estómago, los pulmones, entre otros.

La relación estrecha que mantiene el SNA con las emociones ha impulsado grandes proyectos de investigación en los últimos años. Los múltiples textos que han profundizado en este estudio han causado controversia sobre el grado de coherencia y especificidad que presenta, entendiendo por coherencia a la organización y coordinación de la actividad del SNA por parte de las emociones, y por especificidad, a la medida en la cual sus respuestas difieren respecto a cada emoción. Además, su estructura compleja ha fomentado el planteamiento de múltiples teorías acerca de su funcionamiento. Dentro de la comunidad científica, algunos afirman que éste presenta únicamente dos estados (apagado y encendido), mientras que otros aseguran su capacidad para generar numerosos patrones de activación, es decir, que los cambios fisiológicos provocados por el SNA difieren en cada estado emocional [21]. Para analizar ambas teorías es prudente estudiar sus características tanto estructurales como funcionales, proporcionando así, una visión más amplia de su influencia en las emociones.

El SNA está conformado por dos ramas:

- a) Sistema Nervioso Simpático (SNS).
- b) Sistema Nervioso Parasimpático (SNP).

#### 3.1.1.1 Sistema Nervioso Simpático (SNS)

El SNS es responsable del estado “encendido” del SNA, pues controla las respuestas corporales durante situaciones estresantes tanto físicas como mentales. Produce la aceleración del ritmo cardíaco, dilatación de los bronquios, contracción muscular, dilatación de las pupilas, disminución del movimiento y secreciones estomacales, disminución de la producción de saliva y liberación de adrenalina [22].

#### 3.1.1.2 Sistema Nervioso Parasimpático (SNP)

El SNP es responsable del estado “apagado” del SNA, pues mantiene el medio corporal interno mientras se está en reposo (homeostasis), es decir, mientras no se encuentra bajo una situación de estrés. El SNP ralentiza la frecuencia cardíaca y disminuye la presión arterial, además de generar mayor actividad tanto en las glándulas salivales como en la actividad estomacal e intestinal [22].

#### 3.1.1.3 Funciones del Sistema Nervioso Autónomo

El SNA se encarga de gestionar múltiples funciones corporales, tanto reguladoras como coordinadoras, por lo que en determinadas situaciones resultaría difícil relacionar un estado afectivo con las respuestas fisiológicas. Sin embargo, se ha demostrado, a través de investigaciones y procedimientos experimentales, que la interacción entre el SNS y el SNP posibilita patrones de activación, es decir, cada sistema provoca cambios fisiológicos que, en conjunto, reflejan un estado emocional específico. Las metodologías analizadas en el Estado del Arte presentan resultados consistentes en la actividad del SNA, el cual responde con certeza a las emociones inducidas mediante estímulos. Además, el registro de señales fisiológicas resulta una técnica directa para el reconocimiento de emociones, ya que, a diferencia de las técnicas audiovisuales como las expresiones faciales o la voz, que en ocasiones pueden modificarse a voluntad, el estado emocional se refleja inherentemente en la actividad del SNA. Los resultados obtenidos en el laboratorio son alentadores y han sido clave para el desarrollo de la computación afectiva.

A continuación, se describen las señales fisiológicas más empleadas en el Estado del Arte, dada su efectividad para la detección de emociones, así como sus características eléctricas y los dispositivos específicos para su adquisición.

### 3.1.2 Electrocardiograma (ECG)

El electrocardiograma (ECG, por sus siglas en inglés) es una medición que representa por medio de deflexiones la actividad eléctrica del corazón. Se registra la diferencia de potencial entre los electrodos adheridos a la superficie de la piel y se grafica en función del tiempo. En la práctica, comúnmente son empleados electrodos de tipo Ag/AgCl y deben colocarse en puntos anatómicos específicos. Además, es recomendable aplicar pasta o gel conductor como medio de contacto entre el electrodo y la piel.

Como se estudió en el Estado del Arte, el ECG es ampliamente utilizado para obtener la variabilidad de la frecuencia cardíaca (HRV, por sus siglas en inglés), la cual se describe a detalle en la sección 3.2.3.1. Ésta es una herramienta clave para la obtención de características que permiten evaluar la actividad del SNA, y con ello, determinar el estado emocional. Por lo tanto, es indispensable el estudio y comprensión de los elementos que conforman el registro electrocardiográfico.

En la Figura 2 se muestra una señal típica de ECG, caracterizada por un ciclo repetitivo de tres componentes eléctricas:

- 1) Onda P (despolarización auricular)
- 2) Complejo QRS (despolarización ventricular)
- 3) Onda T (repolarización ventricular)

Como se observa en la Figura 2, el complejo QRS está compuesto por tres ondas (Q, R y S), donde la amplitud más alta le corresponde a la componente R. El intervalo entre los complejos QRS adyacentes es denominado intervalo de R a R (RR), ya que representa la distancia entre las componentes R de cada complejo. De igual forma, es común emplear el término “intervalo de normal a normal (NN)” para referirse al número de intervalos RR cuando se obtienen parámetros en el dominio del tiempo.

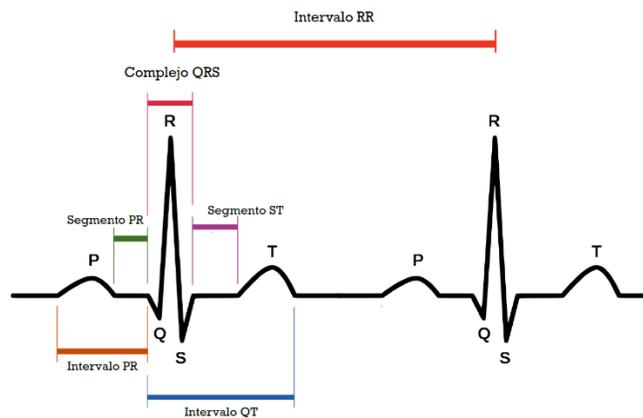


Figura 2. Señal de ECG ideal

Para calcular los intervalos RR se identifican los picos de cada complejo mediante la implementación de transformaciones que, explotando la pendiente pronunciada de los complejos QRS, generan una señal a partir de la cual es posible detectar las componentes R mediante el uso de umbrales. En la literatura han sido ampliamente utilizados los algoritmos de Pan and Tompkins [23], Hamilton [24], Christov [25], Engelse and Zeelenberg [26] y la función de suma de pendiente ECG [27]. En [28] se realiza un estudio comparativo de estos algoritmos.

### 3.1.3 Fotopleletismografía (PPG) / Pulso de volumen de sangre (BVP)

Para medir la presión sanguínea y otros parámetros cardiovasculares, se implementa una técnica no invasiva denominada fotopleletismografía (PPG, por sus siglas en inglés), la cual consiste en un sistema conformado por un LED como fuente luminosa y un fotodiodo como receptor, el cual se compone de un semiconductor sensible a la incidencia de fotones. Durante las últimas décadas se ha implementado la PPG con luz infrarroja y/o roja para el control de salud y práctica médica diaria, sin embargo, los dispositivos vestibles más novedosos, como pulseras y relojes inteligentes, han empleado la luz verde como fuente luminosa. Esto se debe al desarrollo de estudios comparativos que han demostrado el gran desempeño que presentan diversas longitudes de onda en la detección del ritmo cardíaco. En [29] se analizan longitudes menores a la infrarroja (800-960 nm), en donde los resultados demuestran que 633 nm y 825 nm son igualmente sensibles para registrar la frecuencia cardíaca correctamente. En [30] se concluye que no existen diferencias significativas entre la luz roja y la infrarroja cuando se mide el ritmo cardíaco. De igual forma, en [31] se obtuvo que, en dos entornos distintos, la luz verde (525 nm) detecta la frecuencia cardíaca con mayor grado de precisión que la infrarroja (880 nm). Esto demuestra que a menor longitud de onda, mayor es la absorción de la

sangre y menor la profundidad de penetración, lo cual permite obtener una indicación más clara del volumen de sangre en el tejido.

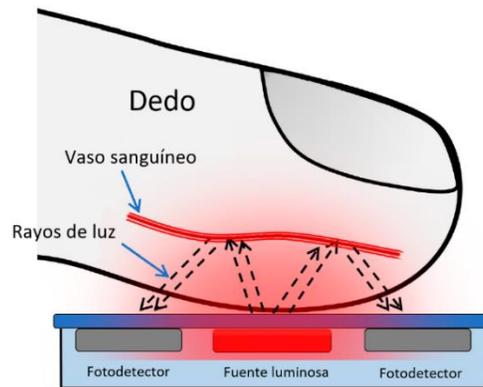


Figura 3. Adquisición de la señal PPG a partir del método de reflexión

La cantidad de luz que regresa o pasa a través del dedo es proporcional al volumen de sangre que fluye por el tejido. Por lo tanto, es posible detectar el pulso máximo y mínimo e inferir con ello la frecuencia cardíaca. Cuando una persona se encuentra en un estado de relajación los vasos sanguíneos se dilatan, mientras que, si se experimentan estados de ansiedad o miedo, ocurre lo contrario. Por dicho motivo, la PPG resulta una técnica viable para la detección de emociones. Esta técnica ha adquirido gran popularidad debido a la flexibilidad que presenta en cuanto a la ubicación del sensor. En múltiples estudios, los dispositivos son evaluados en puntos anatómicos diferentes para determinar la precisión del pulso cardíaco en cada zona, como lo es la frente, el cuello, el pecho, los brazos, etc. Actualmente, debido a los buenos resultados que proporcionan, la mayoría de los dispositivos son colocados ya sea en la muñeca, en la punta del dedo o en otros casos, aunque menos frecuente, en el lóbulo de la oreja. En [30] se concluye que para obtener la frecuencia cardíaca, oxigenación y respiración, la mejor ubicación es la punta del dedo, esto tanto en condiciones de reposo como en movimiento. Otro de los beneficios potenciales de la PPG es su implementación en sistemas empleados para registros ambulatorios o domiciliarios, ya que los avances tecnológicos han favorecido la reducción de sus componentes y han propiciado su integración en múltiples dispositivos portátiles, en los cuales se busca disminuir tanto el tiempo de procesamiento como el almacenamiento de datos. Para cumplir dichos objetivos es necesario reducir la tasa de muestreo a un valor tal que no se pierda precisión durante el registro de la señal. En la literatura se han obtenido buenos resultados empleando frecuencias de muestreo por debajo de los 100 Hz [32] [33].

La Figura 4 ilustra una señal típica obtenida mediante PPG, en la cual se indica la ubicación del pico sistólico, el pico diastólico e intervalo pulso a pulso (IPP), donde éste último es análogo al intervalo RR obtenido a partir de ECG.

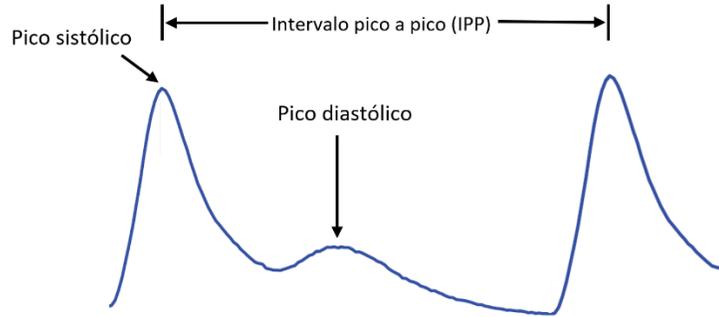


Figura 4. Señal de fotopleletismografía típica.

Para estimar la frecuencia cardíaca mediante esta técnica, es indispensable identificar los picos sistólicos de la señal. Es importante destacar que el espectro de ésta se encuentra en intervalos de menor frecuencia que el espectro de la señal electrocardiográfica, por lo que la lógica de decisión aplicada al comportamiento del complejo QRS es inapropiada para el análisis de PPG. Por dicho motivo, se han desarrollado diversos algoritmos que se acoplan a las características morfológicas de la señal, lo cual ha dado paso al desarrollo de dispositivos médicos basados en esta técnica. En [34] se emplean filtros no lineales y algoritmos de clasificación. Por otro lado, [35] calcula la primera derivada de la señal, en la cual se identifican los cruces por cero e inflexiones máximas. En [36] se desarrolla un algoritmo dirigido a aplicaciones en tiempo real, el cual implementa reglas de decisión y posteriormente, un resonador en frecuencia cero. De igual forma, se han desarrollado algoritmos basados en la transformada de Wavelet [37] y en la transformada de Hilbert [38]. En [39] se presenta un algoritmo sencillo y ligero denominado *Mountaineer's method*, el cual compara muestras sucesivas para detectar el flanco ascendente previo a un pico.

Como se analizó en el Estado del Arte, el ECG ha sido una herramienta ampliamente utilizada para determinar la HRV, sin embargo, debido a los avances tecnológicos, se han considerado otras técnicas que presentan resultados prometedores. Las desventajas que surgen a partir de la implementación del ECG, es que, como se mencionó en la sección 3.1.2, es necesario conectar al menos dos electrodos en puntos anatómicos específicos y es recomendable aplicar pasta o gel para electrodos. Estos inconvenientes pueden evitarse mediante el uso de la PPG, ya que proporciona más opciones en cuanto a la ubicación del sensor, además de no necesitar pasta o gel conductor, lo que favorece su aplicación en dispositivos portátiles. Diversos estudios han comparado el desempeño de la PPG con respecto al del ECG [40] [41] [42], por lo que, además de su practicidad, se ha confirmado su eficacia para calcular parámetros cardiovasculares, así como la HRV, la cual es denominada como variabilidad de la frecuencia del pulso (PRV, por sus siglas en inglés) cuando es extraída mediante fotopleletismografía. Por otro lado, los datos obtenidos a partir de la PPG presentan un mayor ruido que aquellos derivados del ECG debido a diversos factores externos como dispositivos luminosos, movimiento, ubicación del sensor y características fisiológicas que llegan a influir en las propiedades de reflexión y absorción de la piel. Por lo tanto, es de gran importancia aplicar métodos de acondicionamiento confiables, los cuales mejoren la calidad de los datos y, a su vez, no representen una carga computacional significativa.

### 3.1.4 Actividad electrodérmica (EDA) / Respuesta Galvánica de la Piel (GSR) / Conductividad de la piel (SC).

La Respuesta Galvánica de la Piel (GSR, por sus siglas en inglés) o también conocida como Actividad Electrodérmica (EDA, por sus siglas en inglés), es una medida que representa la variación de la conductividad de la piel.

Cuando una persona se encuentra en un estado de excitación, la actividad de las glándulas sudoríparas incrementa y secretan sudor en la superficie de la piel, provocando el aumento en la conductancia del tejido. La EDA es regulada únicamente por la rama simpática del SNA, por lo que es empleada

para detectar estados emocionales de alta excitación como el estrés, el miedo, la ira, entre otros. Por lo tanto, además de la HRV, la actividad electrodérmica es una herramienta de la cual se extraen características que auxilian en la detección de emociones.

La conductancia de la piel se mide a través de dos electrodos ubicados en partes del cuerpo con alta concentración de glándulas sudoríparas. Comúnmente se colocan en la segunda falange de los dedos medio e índice, pero también se pueden ubicar en la palma de la mano o la planta del pie.

Actualmente existen dos métodos para registrar la EDA:

- 1) *Exosomático*: se aplica una pequeña corriente que pasa a través de los electrodos colocados en la piel.
- 2) *Endosomático*: se mide el voltaje entre los electrodos sin aplicar una corriente externa.

De las técnicas descritas anteriormente, se ha demostrado que el método más empleado por estudios recientes es el *exosomático*. Para calcular la resistencia de la piel se aplica la ley de Ohm, que establece la relación entre el voltaje aplicado entre los electrodos y la corriente que pasa a través de la piel ( $R = V/I$ ). Cabe destacar que la corriente que recorre el tejido suele ser del orden de los  $\mu A$ .

Para la medición de la EDA es importante considerar diversos aspectos, tanto el tipo y ubicación de los electrodos, como la pasta que será el medio conductor entre los electrodos y la piel. En [43] se sugiere emplear pasta con salinidad semejante a la del sudor y colocar los electrodos en puntos anatómicos específicos, tal como se muestra en la Figura 5. Otro factor importante es la frecuencia de muestreo a la que será adquirida la señal electrodérmica. En [44] se obtienen resultados satisfactorios empleando una tasa de muestreo de 31 Hz.

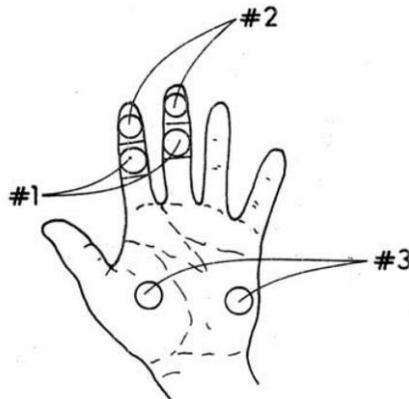


Figura 5. Tres colocaciones de electrodos para registrar la actividad electrodérmica. La #1 involucra superficie volar en falanges medias, la #2 involucra superficie volar de falanges distales y la #3 involucra las eminencias tenar e hipotenar de la palma [43].

La señal electrodérmica se divide principalmente en dos componentes:

- 1) *Nivel de conductancia de la piel (SCL, por sus siglas en inglés)*: es el componente tónico (0-0.16 Hz), el cual representa el nivel base de la conductancia de la piel y se relaciona con los cambios lentos de la señal EDA [45].
- 2) *Respuesta de conductancia de la piel (SCR, por sus siglas en inglés)*: representa el componente fásico (0.16-2.1 Hz), los cuales son los eventos transitorios contenidos en la señal EDA. Refleja las variaciones rápidas y abruptas ante un estímulo [45].

En la Figura 6 se muestran las características principales del componente fásico de EDA, las cuales son calculadas para extraer información relevante de la señal.

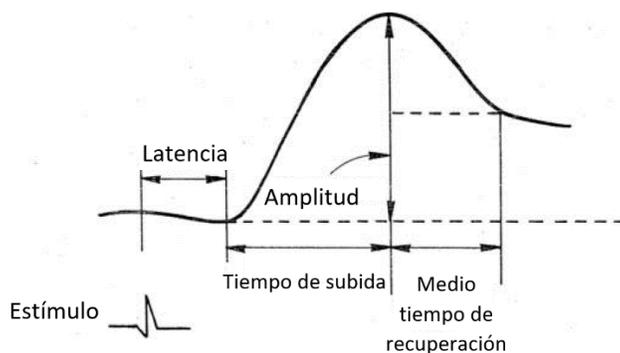


Figura 6. Representación gráfica del componente fásico de EDA [43].

Durante el registro electrodérmico se presentan respuestas que pueden estar o no relacionadas con un estímulo específico:

- a) *SCR No específico (NS-SCR, por sus siglas en inglés)*: SCRs que ocurren en la ausencia de un estímulo identificable. Pueden presentarse debido a cambios térmicos u otras regulaciones corporales.
- b) *SCR relacionado a un evento (ER-SCR, por sus siglas en inglés)*: SCRs que pueden ser atribuidos a un estímulo específico.

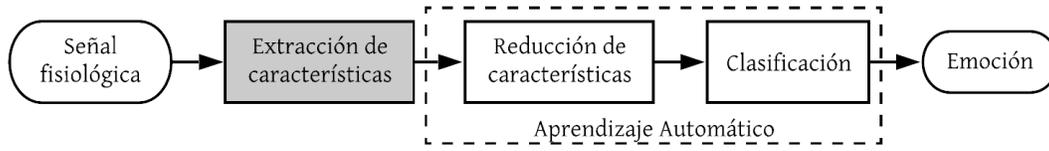
En la Tabla 4 se muestran las definiciones y valores típicos de las mediciones electrodérmicas tanto tónicas como fásicas.

Tabla 4. Mediciones electrodérmicas, definiciones y valores típicos.

Medida	Definición	Valores típicos
<b>Nivel de conductancia de la piel (SCL)</b>	Nivel tónico de la conductividad eléctrica de la piel.	2-20 $\mu S$
<b>Cambio en SCL</b>	Cambios graduales en SCL medidos en dos o más puntos en el tiempo.	1-3 $\mu S$
<b>Frecuencia de NS-SCR</b>	Número de SCRs en ausencia de un estímulo identificable.	1-3 por min
<b>Amplitud SCR</b>	Incremento fásico en la conductancia seguido de un estímulo.	0.1-1.0 $\mu S$
<b>Latencia SCR</b>	Intervalo temporal entre la aplicación del estímulo y el inicio del SCR.	1-3 s
<b>Tiempo de subida SCR</b>	Intervalo temporal entre el inicio de SCR y el pico SCR.	1-3 s
<b>Medio tiempo de recuperación SCR</b>	Intervalo temporal entre el pico SCR y el 50% de la recuperación de la amplitud SCR.	2-10 s
<b>Adaptación SCR (pruebas de adaptación)</b>	Número de estímulos presentados antes de dos o tres pruebas sin respuesta.	2-8 estímulos presentados
<b>Adaptación SCR (pendiente)</b>	Razón de cambio de la amplitud de un ER-SCR.	0.01-0.5 $\mu S$ por prueba

El estudio de la EDA se lleva a cabo analizando las componentes SCR y SCL por separado, por lo tanto, en la sección 3.2.3.2 se mencionan técnicas empleadas para su extracción, así como las características más relevantes para la detección de emociones.

## 3.2 Extracción de características



A partir de la adquisición de alguna o algunas de las señales fisiológicas mencionadas en el capítulo anterior, es necesario obtener de ellas una determinada cantidad de *características*, entendiéndose por éstas a todas aquellas estimaciones en el dominio del tiempo, la frecuencia o mediante métodos no lineales que otorgan un valor representativo de la señal. Mediante ellas es posible describir la evocación de una emoción, así como alimentar más adelante a un clasificador basado en Aprendizaje Automático que dictamine concretamente dicha emoción.

Por ello, con base en los resultados recabados durante el análisis del Estado del Arte reflejados en la Tabla 2, se identifica una serie de atributos para cada una de las señales ahí descritas. En cuanto a la PPG, se analiza la Variabilidad de la Frecuencia de Pulso (PRV, por sus siglas en inglés), la cual se describe a detalle en la sección 3.2.3.1, mientras que para la EDA se extraen características a partir de sus componentes fásico (SCR) y tónico (SCL), presentadas en la sección 3.2.3.2. Todos estos atributos parten de unos fundamentos básicos en los que se sustentan, donde en el dominio del tiempo resultan ser el tiempo de levantamiento y la derivada. Mediante métodos estadísticos se emplea el promedio, la raíz media cuadrática, la varianza, la desviación estándar y los cuantiles. Y en el dominio de la frecuencia se encuentran la transformada rápida de Fourier y la densidad espectral de potencia. De esta forma, se comienza por explicar los fundamentos necesarios de las herramientas matemáticas requeridas para identificar cada uno de estos atributos y posteriormente se indagará en ellos con el fin de comprender su funcionamiento e importancia para la clasificación de emociones.

### 3.2.1 Dominio del tiempo

Estos parámetros suelen emplearse para el análisis de señales cuyo comportamiento no es periódico, por lo que su aplicación suele verse en las señales fisiológicas EMG y EDA.

**Tiempo de levantamiento ( $T_r$ ):** corresponde al intervalo de tiempo dentro del cual la respuesta al escalón de un sistema pasa del 10% al 90% de su valor final. En el caso del EDA suele estimarse a partir de un valor *onset*, correspondiente a aquel valor 10% por encima de la señal base, hasta el valor máximo de la señal. Para este trabajo se requiere que el sistema pueda determinar tiempos de levantamiento con resolución de centenas de milisegundos.

**Derivada simétrica:** involucra la variación que experimenta una señal, denotada por  $x[n]$ , en un instante de tiempo, denotado por  $h$ , cuya definición se da mediante la ecuación (1), que difiere de la definición tradicional de la derivada dado que ésta otorga un término de error de truncamiento proporcional a  $\varepsilon(h^2)$  en lugar de  $\varepsilon(h)$  que presenta la definición tradicional. Por ello, en el dominio discreto este instante de tiempo yace estrechamente ligado a la frecuencia de muestreo de la señal  $f_s$ , concepto que se verá más adelante en la sección 3.2.2, ya que a mayor tasa de muestreo menor será el error proporcionado por esta expresión.

$$x'[n] = \lim_{h \rightarrow 0} \frac{x[n+h] - x[n-h]}{2h}; \quad n = 0, 1, \dots, N \quad (1)$$

#### 3.2.1.1 Métodos estadísticos

Estas estimaciones tienen aplicación tanto en las señales fisiológicas mencionadas anteriormente, como en los resultados del tratamiento en el dominio de la frecuencia que se analizan en la sección 3.2.2, para así obtener dos tipos de características distintas para cada uno de ellos.

**Promedio o media aritmética ( $m_x$ ):** es la suma de todos los elementos de un conjunto entre el tamaño  $N$  del mismo conjunto.

$$m_x = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \quad (2)$$

**Raíz de la media cuadrática (RMS):** la media cuadrática de un conjunto de valores es la raíz cuadrada de la media aritmética de los elementos al cuadrado. Es la medida de la magnitud de una cantidad variable, la cual puede ser una serie de valores discretos o una función matemática continua. Esta definición es expresada para valores discretos en la ecuación (3).

$$RMS = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} x[n]^2} \quad (3)$$

**Varianza ( $s^2$ ):** denota a una medida que describe el grado de dispersión que experimenta un conjunto de datos respecto a su media, por lo que resulta un valor de relevancia dentro de la distribución de los datos. De esta forma, la varianza se puede entender como una clase de promedio expresado mediante la diferencia de los datos de una señal con respecto a su media aritmética y cuyo valor cuadrático se deriva de la supresión de valores negativos que puede ocasionar la sustracción, de modo que se evite obtener una variación de valor nulo. Así pues, la definición de varianza para una muestra de datos de toda población resulta en la expresión (4).

$$s^2 = \frac{1}{N-1} \sum_{n=0}^{N-1} (x[n] - m_x)^2 = \frac{1}{(N-1)} \left( \sum_{n=0}^{N-1} x[n]^2 - \frac{1}{N} \left( \sum_{n=0}^{N-1} x[n] \right)^2 \right) \quad (4)$$

**Desviación estándar ( $s$ ):** corresponde a la raíz cuadrada de la varianza. Este parámetro ayuda a evitar la existencia de valores muy elevados obtenidos en la varianza.

**Cuantiles ( $C_i$ ):** es una medida de posición que, una vez ordenados los datos de forma ascendente, indica el valor de la variable por debajo del cual se encuentra un porcentaje dado de observaciones. Los cuantiles se representan del 1% al 99%, por lo que el cuantil 0% y 100% no se consideran válidos. La ecuación (5) indica el cálculo de la posición de la variable correspondiente el cuantil  $i$ .

$$c = \frac{N * i}{100} \quad (5)$$

donde  $N$  es el número total de muestras e  $i$  el cuantil. El resultado de esta operación es un número real con parte entera  $n$  y parte decimal  $d$ . Teniendo en cuenta estos dos valores, la ecuación (6) es aplicada:

$$C_i = \begin{cases} x[n+1] & d \neq 0 \\ \frac{x[n] + x[n+1]}{2} & d = 0 \end{cases} \quad (6)$$

### 3.2.2 Dominio de la frecuencia

Las características estimadas mediante este análisis permiten identificar las periodicidades que presentan las señales fisiológicas dada la descripción que existe de éstas como una combinación lineal de funciones armónicas seno y coseno. De esta forma, resulta una herramienta muy útil en la detección de emociones, dado que permite realizar la comparación de dichas estimaciones en varios instantes de tiempo a fin de identificar algún cambio en ellas que pueda corresponder a la evocación de alguna emoción en específico.

Esta clase de análisis parte del concepto de la transformada de Fourier (FT, por sus siglas en inglés), que a su vez resulta una generalización de las series exponenciales de Fourier descritas en (7) que permiten describir a una señal continua como la suma de una serie de señales senoidales y cosenoidales con una determinada amplitud y frecuencia, denotadas por  $c_k$  y  $\omega_0$  respectivamente, que en conjunto reciben el nombre de *componentes espectrales* de la señal.

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \quad (7)$$

Sin embargo, la expresión correspondiente a señales discretas que muestrean a una señal continua en  $N$  puntos se encuentra descrita por la transformada discreta de Fourier (DFT, por sus siglas en inglés) cuyas ecuaciones de análisis y de síntesis están dadas por (8) y (9) respectivamente, donde  $x[n]$  corresponde a la señal en el dominio del tiempo y  $X[k]$  la señal en el dominio de la frecuencia. Estas expresiones pueden considerarse el caso particular de la transformada de Fourier de tiempo discreto (DTFT, por sus siglas en inglés) cuyo dominio continuo en la frecuencia sobre el intervalo  $0 \leq \omega < 2\pi$  es muestreado en  $N$  puntos dentro de la DFT en la forma  $\omega_k = 2\pi k/N$ .

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N}, \quad n = 0, 1, \dots, N-1 \quad (8)$$

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \quad (9)$$

De este modo es posible evaluar las componentes espectrales de funciones periódicas o no periódicas, bajo la cual la DFT asume una periodicidad de ambas a pesar de la naturaleza que tengan. Tome como ejemplo la Figura 7 en la cual se evalúa una función no periódica en el intervalo  $[-M, M]$ , la acción de la DFT sobre ésta es asumir una periodicidad de la señal al extender la misma a lo largo de los intervalos adyacentes superiores a  $M$  e inferiores a  $-M$ . Para el análisis de una señal de esta naturaleza, como es el caso de algunas señales fisiológicas mencionadas en la sección 3.1, es conveniente acotarla adecuadamente, evitando cambios abruptos en los límites de su intervalo, ya que de lo contrario es posible identificar componentes espectrales erróneas que no reflejen fielmente la naturaleza de la señal. Un ejemplo de este caso se da en la Figura 8 cuya señal en rojo secciona a la señal en negro de 13 [Hz] en una ventana de  $13 \frac{1}{2}$  ciclos de la señal en negro, su respectivo análisis en frecuencia resulta equivocado ya que demuestra una componente fundamental de 13.5 [Hz], en cambio un ventaneo de 13 ciclos de la señal azul logra identificar correctamente la componente fundamental de 13 [Hz].

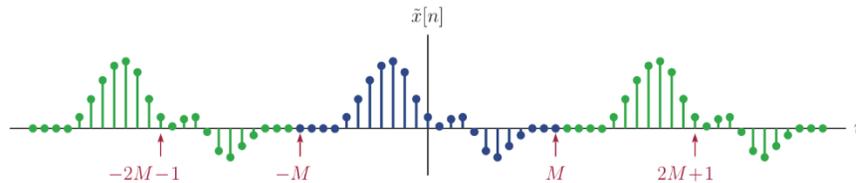


Figura 7. Extensión periódica de una función no periódica por medio de la DFT.

Asimismo, deben cuidarse los efectos como el derrame espectral “*spectral leakage*” que sucede al momento de seccionar un conjunto de datos de la señal (ventaneo) para analizarlo en frecuencia, que es el caso más frecuente en la práctica, ya que en realidad esto corresponde a un ventaneo con una señal rectangular, por lo que la señal a analizar se expresaría en la forma:

$$x_w[n] = w[n]x[n] = \begin{cases} x[n], & 0 \leq n \leq N-1 \\ 0, & \text{de otra forma} \end{cases}$$

Sin embargo, dado que el producto de dos señales en el dominio del tiempo corresponde a una operación de convolución en el dominio de la frecuencia, el espectro adquirido de  $x_W[n]$  presenta una dispersamiento de la potencia de la señal  $x[n]$  hacia todas las componentes de la respuesta en fre-

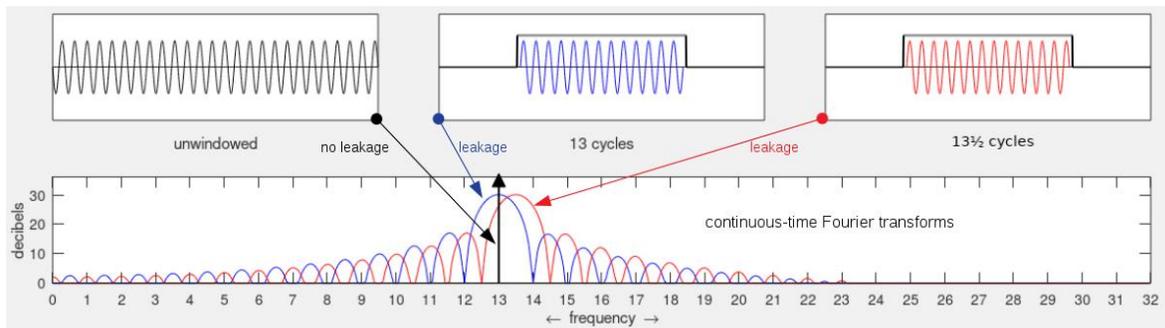


Figura 8. Ejemplificación (no real) de los problemas ocasionados por derrame espectral y por la captura de la señal en un valor no múltiplo de las frecuencias que lo componen. Imagen adquirida de [108].

cuencia de la señal cuadrada. Tome el caso de la señal en negro de la Figura 8 cuyo espectro es una única espiga a los 13 [Hz] y en cambio la señal azul, que captura una porción de la señal mediante un ventaneo rectangular, entrega un espectro que se alinea a los 13 [Hz] pero su magnitud se ve dispersada a lo largo de todo el espectro debido a los lóbulos contiguos que presenta la respuesta en frecuencia de la señal rectangular.

A fin de evitar dicho esparcimiento, es recomendable el uso de otras ventanas que presenten una mayor atenuación en sus lóbulos laterales, como podría ser una ventana Triangular, Hamming, Hanning o Blackman, por mencionar algunos. No obstante, este aumento en la atenuación de los lóbulos laterales se realiza a cambio de una pérdida en la resolución de la señal dado que el lóbulo central de la ventana aumenta en grosor, véase la comparación entre ventanas en la Tabla 5, dificultando la distinción entre componentes espectrales contiguas que yazcan contenidas en ella. Tome los casos de la Figura 9 donde se expone el ventaneo de una señal con frecuencias de 0.1 [Hz], 0.11 [Hz] y 0.3 [Hz] por una ventana rectangular y Hamming con longitudes de  $P=25$ ,  $P=65$  y  $P=100$ . Es importante mencionar que  $P$  corresponde al número de puntos que conforman a la ventana y que puede resultar en  $P \neq N$ . En todos los casos es posible distinguir la frecuencia de 0.3 [Hz], pero la situación cambia respecto a las frecuencias de 0.1 y 0.11 [Hz]. En el primer caso para ambas ventanas es indistinguible la diferencia entre las señales, mientras que, en el segundo caso es apenas perceptible para la ventana rectangular pero aún imperceptible para la ventana Hamming, y finalmente, en el último caso es evidente su distinción en ambas ventanas a pesar de que en la ventana rectangular se presenta la dispersión mencionada debido a los lóbulos adyacentes de la ventana rectangular. De esta forma, es importante considerar estos criterios al momento de escoger el tipo de aplicación que se desea implementar para garantizar una resolución adecuada de los datos con la menor interferencia posible de lóbulos adyacentes.

Tabla 5. Desempeño de ventanas. Tabla tomada de [46]

Ventana	Ancho del lóbulo principal de $W(\omega)$	Lóbulo principal a 2do. lóbulo	$A_s$ Mínima
Rectangular	$4\pi/N$	-13	-21
Triangular	$8\pi/N$	-26	-25
Hanning	$8\pi/N$	-31	-44
Hamming	$8\pi/N$	-41	-53
Blackman	$12\pi/N$	-57	-74

Otro factor importante por considerar dentro de la evaluación en frecuencia de una señal es la resolución de la DFT dada por (10) donde  $f_s$  es la frecuencia de muestreo de la señal y  $N$  el número de puntos en los que se realizará la DFT, ya que normalmente se busca un valor reducido de este factor para permitir la visualización de múltiples componentes espectrales de la señal. Convenientemente, dado que las señales fisiológicas no suelen manejar tasas de muestreo tan elevadas, no se requiere de un gran número de puntos para realizar este análisis, por lo que es posible adquirir buenas resoluciones de éstas sin requerir de largas cantidades de memoria.

$$\Delta f = \frac{f_s}{N} \quad (10)$$

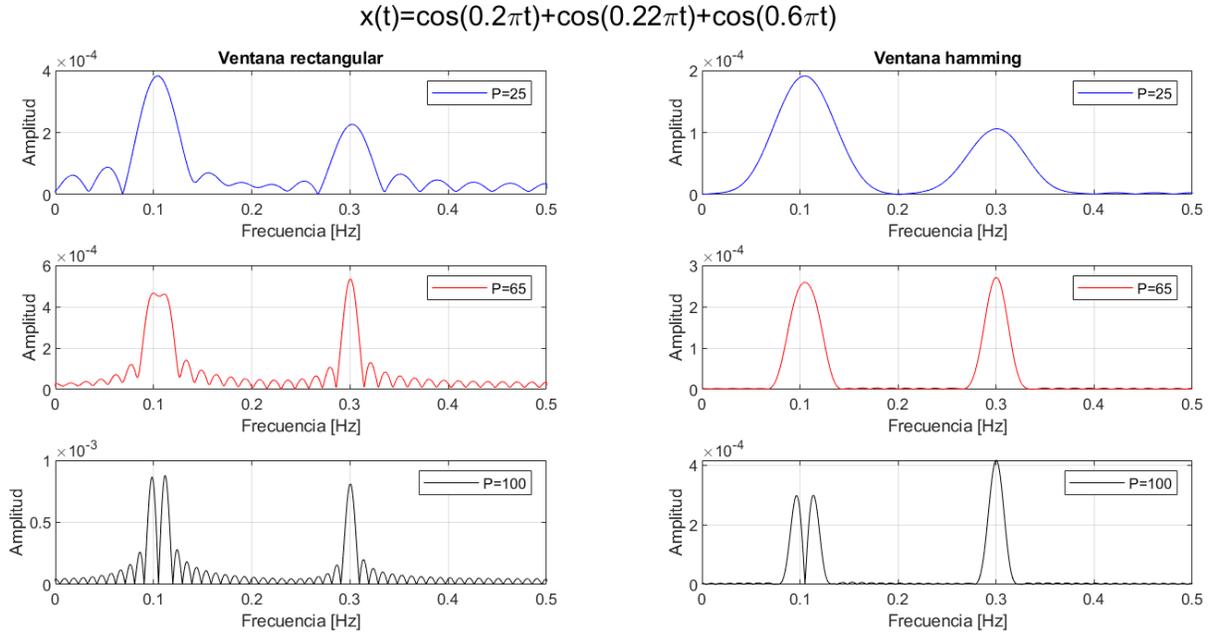


Figura 9. Relación entre la resolución y la longitud de una ventana rectangular y Hamming.

Asimismo, existe la posibilidad de incrementar el número de componentes en el espectro al extender el vector de datos mediante la adición de ceros, esta técnica recibe el nombre de “zero padding”. Sin embargo, esta solución conlleva una interpolación de las componentes espectrales para facilitar la visualización de más características del espectro, por lo que no añade información al conjunto de datos inicial y, por lo tanto, no aumenta la resolución de la señal.

### 3.2.2.1 Transformada rápida de Fourier

Pese a que la DFT permite identificar las componentes espectrales que conforman a las señales fisiológicas, en la práctica no se implementa directamente la definición de la ecuación (9) dada la gran carga computacional que requiere su implementación, por ello existe lo que se conoce como la Transformada Rápida de Fourier (FFT, por sus siglas en inglés). Este término se le atribuye al conjunto de algoritmos computacionalmente eficientes que permiten la implementación de la DFT al considerar las propiedades de simetría y periodicidad, descritas en las ecuaciones (11) y (12) respectivamente, de un factor  $W_N^{kn} = e^{-\frac{j2\pi kn}{N}}$  conocido como factor “twiddle”, así como el algoritmo divide-y-vencerás que consiste en subdividir la DFT de una extensión de  $N$  puntos a DFTs reducidas en un factor  $r$  de modo que se cumpla que  $N = r_1 r_2 r_3 \dots r_n$  siendo  $r \in \mathbb{N}$  el mismo valor para todas las  $r_n$ , este factor se le denomina el “radix” del algoritmo FFT.

$$W_N^{k+N/2} = -W_N^k \quad (11)$$

$$W_N^{k+N} = W_N^k \quad (12)$$

El *radix* más popular para la implementación de la FFT es el *radix 2* como se describe en el algoritmo Cooley-Tukey [47], éste parte por solicitar una extensión de  $N$  valores que sea potencia de dos o, en su caso, la correspondiente extensión de las muestras de la señal a ese valor mediante “*zero padding*”. De esta forma la expresión de la DFT de  $N$  puntos, dada por la ecuación (9), puede dividirse en dos DFTs correspondientes a los elementos con índice par e índice impar como se comprueba en el siguiente desarrollo:

$$X[k] = \sum_{r=0}^{N/2-1} x[2r]W_N^{(2r)k} + \sum_{r=0}^{N/2-1} x[2r+1]W_N^{(2r+1)k}$$

$$X[k] = \sum_{r=0}^{N/2-1} x[2r]W_N^{(2r)k} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1]W_N^{2rk}$$

Considerando la propiedad de  $W_N^{pq} = e^{-\frac{j2\pi pq}{N}} = e^{-\frac{j2\pi q}{N/p}} = W_{N/p}^q$  se obtiene:

$$X[k] = \sum_{r=0}^{N/2-1} x[2r]W_{N/2}^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1]W_{N/2}^{rk}$$

$$X[k] = X[2r] + W_N^k X[2r+1] \quad (13)$$

A partir de la expresión (13) no sólo se comprueba la posibilidad de dividir una DFT de extensión  $N$  en dos DFTs de longitud  $N/2$  correspondientes a los elementos con índice par e impar, sino que además es posible extender esta misma idea a cada una de las DFTs obtenidas  $X[2r]$  y  $X[2r+1]$  para subdividirlas en dos DFTs pares e impares. Esta generalización consecutiva finalizará hasta obtener una simple DFT con dos elementos de entrada para su operación, un ejemplo de esta idea puede observarse en la Figura 10 mediante un arreglo con extensión  $N = 8$ .

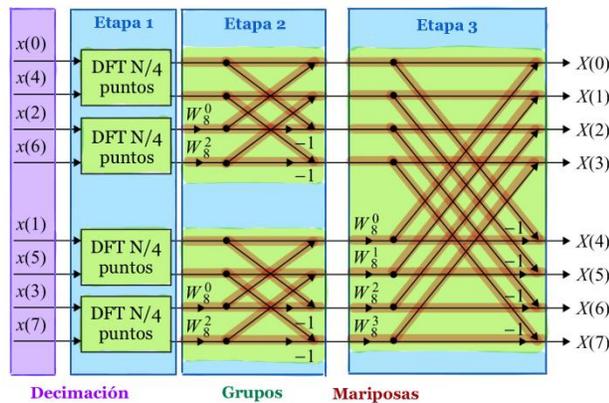


Figura 10. FFT por decimación en el tiempo de 3 Etapas para  $N=8$ .

Asimismo, dentro de la FFT de la Figura 10 se identifican unas estructuras similares a la denotada en la Figura 11, estas reciben el nombre de “operaciones mariposa” cuyo significado, dentro de la *decimación en el tiempo*, refiere a una operación aritmética que yace conformada por una multiplicación, una adición y una sustracción de números complejos. Esta operación resulta uno de los elementos clave dentro del algoritmo FFT dado que permite la reducción del número de multiplicaciones complejas realizadas a un valor  $O\left(\frac{N}{2} \log_2 N\right)$ , muy distante del valor  $O(N^2)$  necesario para una implementación directa de la DFT.

Del algoritmo FFT *radix 2* de la Figura 10 es posible identificar cuatro elementos principales, el primero de ellos es el número de etapas en las que se realiza la partición de la DFT cuyo valor se obtiene como  $m = \log_2 N$ , el segundo es el número de grupos de operaciones mariposa que existen dentro de una etapa, partiendo de un valor  $N/2$  en la primera etapa hasta ser un único grupo en la última, el tercero elemento es el número de operaciones mariposa que se realizan dentro del grupo de operaciones mariposa, partiendo de una única operación por grupo en la primera etapa y terminando con  $N/2$  operaciones en el grupo de la última etapa, y finalmente un ordenamiento de los valores con índice par e impar al comienzo de la FFT para conformar una “*decimación en tiempo*”. Cabe mencionar que también es posible realizar un ordenamiento de los valores a la salida de la FFT en lugar de la entrada, en cuyo caso recibe el nombre de “*decimación en frecuencia*”, sin embargo, para ello se sigue una estructura ligeramente distinta a la decimación en el tiempo y unos cambios en las operaciones de la operación mariposa.

Otro aspecto que presenta la FFT es la posibilidad de guardar los resultados de las operaciones

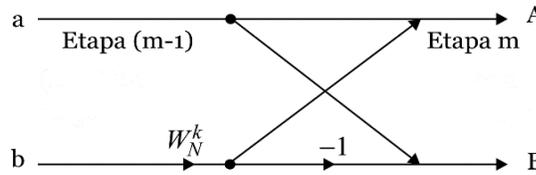


Figura 11. Operación mariposa para la *decimación en el tiempo* descrita como  $A = a + W_N^k * b$  y  $B = a - W_N^k * b$

mariposa de cada etapa sobre los mismos valores de donde se partió, de forma que permite realizar “operaciones en el mismo lugar”. La cualidad anterior, junto con el manejo que se da de elementos con parte real e imaginaria dada la naturaleza de la transformada con números complejos, dan lugar a que la cantidad de localidades de memoria necesarias para realizar la implementación de la FFT sea de  $2N$ , o en su caso  $2N + N/2$  si se deciden guardar los coeficientes de los factores “*twiddle*” previo a la operación de la FFT, lo cual incrementa el uso de memoria, pero a la vez aumenta la velocidad de ejecución del algoritmo.

De acuerdo con los estudios realizados en el Estado del Arte para la detección de emociones, a través de este análisis que se realiza sobre las señales fisiológicas, se secciona todo el intervalo de frecuencias que entrega la FFT en intervalos de interés, de forma que dentro de cada uno de ellos se estime un promedio de la magnitud de las componentes espectrales que lo integran, dichos promedios, o bien alguna razón entre ellos, resultan ser ciertas de las *características* que alimentarán a alguno de los clasificadores que se verán más adelante en la sección 3.5.

### 3.2.2.2 Densidad Espectral de Potencia

Para captar la verdadera naturaleza de las señales fisiológicas, cuyo comportamiento en realidad es aleatorio, es necesario realizar otro tipo de estudio espectral denominado Densidad Espectral de Potencia (PSD, por sus siglas en inglés). Este estudio parte por considerar a las señales aleatorias con media constante y con autocorrelación dependiente sólo en una diferencia de tiempo, es decir como un proceso aleatorio en sentido amplio estacionario (WSS, por sus siglas en inglés). A su vez, normaliza la amplitud de la señal respecto a la resolución de las frecuencias  $\Delta f$ , resultando en unidades de *Potencia/frecuencia*, para así otorgar un espectro que resulta independiente de la longitud de los datos por analizar. La idea anterior se ilustra con los resultados de la Figura 12 mediante la estimación del espectro de dos señales senoidales con ruido a través de la FFT y la PSD considerando diferentes longitudes de  $N$ . De esta forma, se adquiere un espectro cuya amplitud denota la potencia de la señal con respecto a la frecuencia.

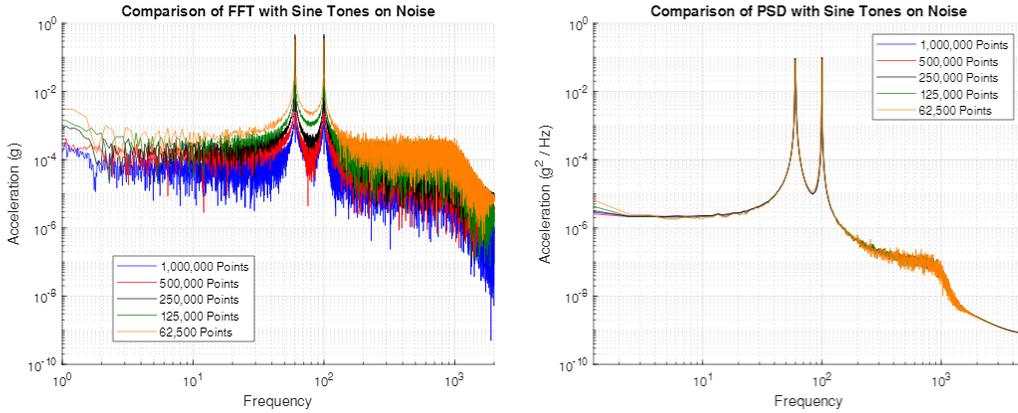


Figura 12. Diferencia entre la FFT y PSD de dos señales senoidales con ruido.  
Imágenes adquiridas de [84]

Por esta razón, al verificar las *características* empleadas en el Estado del Arte se observa que, en su mayoría, se utilizan estimaciones de la PSD por encima de los análisis con FFT dado que las señales fisiológicas suelen ser de naturaleza aleatoria estacionaria y la FFT resulta únicamente una de las herramientas para adquirir dicho espectro.

Este tipo de estudio parte de lo dictado en el Teorema de Parseval con la ecuación (14) que relaciona la energía total de una señal  $x[n]$  con la energía de su espectro a lo largo de la frecuencia, esta última se expresa mediante la suma de todas sus componentes espectrales adquiridas a través de su DFT.

$$E = \sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2 \quad (14)$$

De esta relación se identifica a  $|X[k]|^2$  como la densidad espectral de energía denotada como  $S_{xx}[k]$ , sin embargo, la definición de la densidad espectral de potencia, denotada como  $P_{xx}[k]$  y que maneja señales de energía infinita pero potencia finita, queda descrita por la ecuación (15), considerando que el proceso aleatorio es ergódico en su primer y segundo momento como se explica en [48], que además corresponde a lo que se le conoce como un *periodograma*. Donde  $N$  denota la longitud de datos a evaluar y  $E$  corresponde al valor esperado.

$$P_{xx}[k] = \lim_{N \rightarrow \infty} E \left\{ \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn} \right|^2 \right\}, \quad k = 0, 1, \dots, N-1 \quad (15)$$

Existen múltiples técnicas para estimar la densidad espectral de potencia de una señal, las cuales generalmente quedan clasificadas en dos grupos:

- **Métodos no paramétricos:** estos métodos corresponden normalmente a una implementación cercana de la definición matemática de la PSD expresada en (15) haciendo uso de la FFT. Sin embargo, pese a que no suelen presentarse picos espurios y resultan ser de fácil computación gracias a la FFT, requieren de un gran número de datos para obtener una buena resolución en frecuencia y presentan distorsiones por derrame espectral de acuerdo con el tipo de ventana empleada para su estimación.
- **Métodos paramétricos:** asumen un conocimiento previo de la señal, por lo que definen la existencia de una estructura definida para el proceso aleatorio que es posible de describir mediante un modelo lineal que yace conformado por un determinado número de parámetros.

Estos métodos parten del teorema de Stone-Weierstrass que dicta la posibilidad de aproximar cualquier PSD continua a través de una función racional de polinomios con un grado lo suficientemente elevado. Donde  $\gamma_k$  y  $\delta_k$  corresponden a las magnitudes del polinomio del numerador y denominador respectivamente.

$$S_{xx}(\omega) = \frac{\sum_{k=-m}^m \gamma_k e^{-j\omega k}}{\sum_{k=-p}^p \delta_k e^{-j\omega k}}$$

A su vez, el teorema de factorización espectral descrito en [49] permite expresar dicha función racional en el dominio de la frecuencia complejo mediante la transformada  $Z$  como:

$$S_{xx}(z) = S_{ww}(z) \frac{D(z)D(z^{-1})}{A(z)A(z^{-1})}$$

En el caso particular donde la señal  $w$  corresponda a ruido blanco, su densidad espectral de energía  $S_{ww}$  equivale a una constante  $\sigma_w^2$ , por lo que la ecuación anterior se vuelve (16).

$$S_{xx}(z) = \sigma_w^2 \frac{D(z)D(z^{-1})}{A(z)A(z^{-1})} \quad (16)$$

Así pues, se observa que la PSD de la señal  $x[n]$  puede asociarse a una función de transferencia  $H(z) = D(z)/A(z)$  cuya entrada es ruido blanco  $w[n]$ , por medio de la forma (17).

$$\begin{aligned} X(z) &= W(z)H(z) \\ H(z) &= \frac{X(z)}{W(z)} = \frac{D(z)}{A(z)} = \frac{\sum_{k=1}^p d_k z^{-k}}{1 + \sum_{k=1}^m a_k z^{-k}} \end{aligned} \quad (17)$$

Por lo que la estimación de la PSD de una señal  $x[n]$  se convierte en un problema de modelado que suele resolverse a través de un modelo autorregresivo (AR, por sus siglas en inglés), un modelo de media móvil (MA, por sus siglas en inglés) o autorregresivo de media móvil (ARMA, por sus siglas en inglés), estos dos últimos se analizan a profundidad en [50]. Entre estos el modelo AR resulta el más utilizado dada la representación que ofrece para picos delgados, es decir, datos cuya PSD es elevada en determinadas frecuencias (resonancias), además de resultar en ecuaciones lineales muy sencillas para la obtención de los parámetros AR.

#### ○ **Modelo Autoregresivo**

El modelo AR, también conocido como *método de máxima entropía (MEM)*, por sus siglas en inglés, cuenta con una estructura  $H(z) = 1/A(z)$ , partiendo de la ecuación (17), que puede interpretarse como la salida de un filtro IIR todo polo cuya entrada es ruido blanco  $w[n]$ , del cual las constantes  $a_m$  pasan a ser *coeficientes del modelo AR*, donde el valor  $m$  indica el

orden del modelo. Su descripción mediante su relación de recurrencia lineal se describe en (18).

$$\hat{x}[n] = - \sum_{k=1}^m a_m[k]x[n-k] + w[n], \quad n = 0, 1, \dots, N-1 \quad (18)$$

Normalmente la metodología que siguen los métodos paramétricos se conforma por tres pasos: 1) selección del modelo y orden adecuados, 2) estimación de los parámetros del modelo y 3) estimación del espectro usando dichos parámetros.

Dentro del Estado del Arte no todos los trabajos describen el método que utilizan para realizar la estimación de la PSD, sin embargo, dentro de los pocos que lo mencionan, destaca el uso de los métodos paramétricos o el uso conjunto de un método paramétrico y un método no paramétrico, esto último es posible realizarse para garantizar una estimación adecuada del espectro, aprovechando las ventajas que ofrece cada uno de los métodos por separado.

Ahora bien, antes de comenzar a describir la técnica paramétrica del método de Burg y la técnica no paramétrica del método de Welch, siendo ambos métodos empleados en algunos trabajos del Estado del Arte, es necesario introducir los conceptos de predicción lineal, estructura Lattice y el algoritmo Levinson-Durbin que serán empleados más adelante.

### **Predicción lineal hacia adelante**

Este elemento puede ser observado de forma muy similar al modelo AR mencionado anteriormente, dado que permite predecir el valor futuro  $\hat{x}[n]$  de un proceso aleatorio estacionario a partir de la observación de los valores previos del mismo. Se encuentra descrito a través de la ecuación (19) como la combinación lineal de los valores pasados  $x[n-k]$  ponderados por los pesos  $a_m$  que se denominan *coeficientes de predicción hacia adelante*, que resultan equivalentes a los *coeficientes del modelo AR*. De la misma ecuación, el orden del predictor se indica con el valor  $m$  y el signo negativo es utilizado por conveniencia matemática dada la estructura Lattice que se verá más adelante.

$$\hat{x}[n] = - \sum_{k=1}^m a_m[k]x[n-k], \quad n = 0, 1, \dots, N-1 \quad (19)$$

Asimismo, es posible atribuir un *error de predicción hacia adelante*  $f_m$  mediante la comparación del valor real adquirido de la señal  $x[n]$  con el valor predicho  $\hat{x}[n]$ , conformando la expresión (20).

$$f_m[n] = x[n] - \hat{x}[n]$$

$$f_m[n] = x[n] + \sum_{k=1}^m a_m[k]x[n-k], \quad n = 0, 1, \dots, N-1 \quad (20)$$

### **Predicción lineal hacia atrás**

En forma similar a la predicción hacia adelante, este elemento permite predecir el valor pasado  $\hat{x}[n]$  de un proceso aleatorio estacionario a partir de la observación de valores futuros del mismo, el cual se describe a través de la ecuación (21) como la combinación lineal de valores pasados  $x[n-k]$ , ponderado por los pesos  $q_m$  que se denominan *coeficientes de predicción hacia atrás*. De la misma ecuación, el orden del predictor se indica con el valor  $m$  y el signo negativo es utilizado nuevamente por las razones que se mencionan en el predictor hacia adelante.

$$\hat{x}[n - m] = - \sum_{k=1}^m q_m[k] x[(n - m) + k], \quad n = 0, 1, \dots, N - 1 \quad (21)$$

Asimismo, es posible atribuir un *error de predicción hacia atrás*  $b_m$  mediante la comparación del valor real adquirido de la señal  $x[n - m]$  con el valor predicho  $\hat{x}[n - m]$ , conformando la expresión (22).

$$b_m[n] = x[n - m] - \hat{x}[n - m]$$

$$b_m[n] = x[n - m] + \sum_{k=1}^m q_m[k] x[(n - m) + k], \quad n = 0, 1, \dots, N - 1 \quad (22)$$

### Estructura Lattice

Corresponde a una estructura muy utilizada en los campos de predicción lineal para procesamiento de voz gracias a que permite obtener al mismo tiempo los errores de predicción hacia adelante  $f_m$  y hacia atrás  $b_m$ . De igual forma, la estructura Lattice es bastante útil para la implementación de filtros FIR e IIR dado que ofrece una estructura modular invariante conforme se incrementa el orden del filtro y, además, brinda coeficientes de magnitud menor a la unidad que permiten adquirir menores errores por redondeo durante las operaciones realizadas en cada una de las etapas del filtro, al manejarse una aritmética de punto fijo. Esta estructura parte de los errores de predicción hacia adelante (20) y hacia atrás (22) mencionados anteriormente para conformar las ecuaciones descritas en (23).

$$f_0[n] = b_0[n] = x[n]$$

$$f_m[n] = f_{m-1}[n] + K_m b_{m-1}[n - 1], \quad m = 1, 2, \dots, M \quad (23)$$

$$b_m[n] = K_m f_{m-1}[n] + b_{m-1}[n - 1]$$

Donde  $f_m$  corresponde al *error de predicción hacia adelante*,  $b_m$  corresponde al *error de predicción hacia atrás* y  $K_m$  son los *coeficientes de reflexión* provenientes del criterio de estabilidad de Schür. Estos últimos poseen una relación directa con los coeficientes de predicción  $a_m$  dada la recursión Levinson Durbin que se describe más adelante. En la Figura 13 es posible apreciar una representación gráfica de la estructura Lattice de la cual se observa que el orden del modelo AR, el orden del predictor y el número de etapas de la estructura Lattice es la misma  $m$ .

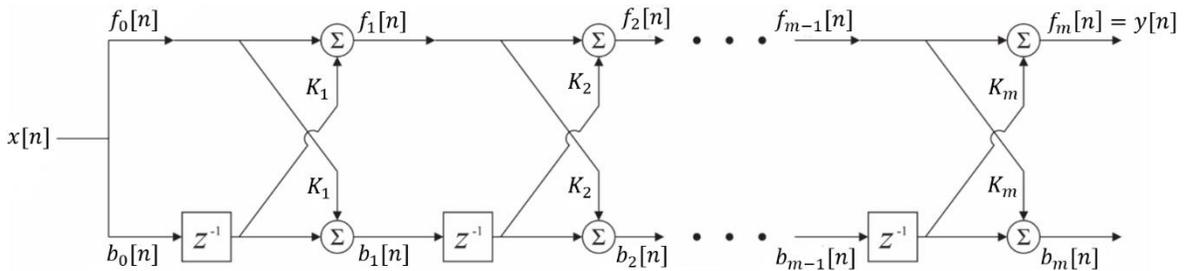


Figura 13. Estructura Lattice. Imagen modificada de [85].

### Recursión Levinson-Durbin

Esta recursión permite conocer la relación que guardan los *coeficientes del modelo AR* con los *coeficientes de reflexión* mediante la ecuación (24) que permite obtener los coeficientes  $a_m$  de una etapa  $m$  a partir de sus respectivos valores en la etapa anterior  $m - 1$  y la ecuación (25) que permite calcular la energía del error  $E_m$  de una etapa  $m$  a partir de sus respectivos valores en la etapa anterior  $m - 1$ . El origen de esta recursión parte de la solución de las ecuaciones normales aumentadas para un predictor lineal hacia adelante y hacia atrás cuyo desarrollo matemático puede verificarse en [46].

$$a_m[n] = \begin{cases} a_{m-1}[n] + K_m a_{m-1}[m-n], & n = 0, 1, \dots, M-2 \\ K_m, & n = M-1 \end{cases}, m = 0, 1, \dots, M-1 \quad (24)$$

$$E_m = E_{m-1}(1 - |K_m|^2), \quad m = 1, 2, \dots, M \quad (25)$$

### 3.2.2.2.1 Método de Burg

Este método paramétrico estima los coeficientes de reflexión del modelo AR de una señal a través del uso de una estructura Lattice de orden recursivo, en cuya recursión se busca la minimización de los errores medios cuadráticos en las predicciones hacia adelante  $f_m$  y hacia atrás  $b_m$  de la señal. Los coeficientes  $K_m$  óptimos adquiridos permiten obtener los parámetros  $a_m$  del modelo AR al satisfacerse la recursión Levinson-Durbin. De esta forma, es posible estimar la PSD de una señal a través del uso de los parámetros AR.

Este método ofrece una alta resolución en frecuencia para señales con baja relación señal-a-ruido (SNR), buenas estimaciones para señales con pocos datos, no presenta derrame espectral y garantiza la conformación de un modelo estable. Sin embargo, la localización de los picos se encuentra estrechamente ligado a la fase inicial de la señal, además puede presentar picos espectrales espurios no propios de la señal, y la precisión del método decrece para modelos con ordenes elevados, longitud de datos extensa y señales con SNR elevados.

Como se mencionó, el corazón de este método parte de la obtención de los *coeficientes de reflexión*  $\hat{K}_m$  mediante la reducción del error medio cuadrático de los errores de predicción hacia adelante  $f_m$  y hacia atrás  $b_m$  expresándose como:

$$\varepsilon_m = \frac{1}{2} \sum_{n=m}^{N-1} \{|f_m[n]|^2 + |b_m[n]|^2\}$$

De esta forma, se busca el valor del coeficiente  $K_m$  que minimice el error  $\varepsilon_m$ , obteniéndose a través de la derivada de ésta respecto a  $\hat{K}_m$  con igualación a cero, de cuyo desarrollo se obtiene la expresión (26).

$$\begin{aligned} \frac{\partial \varepsilon_m}{\partial K_m} &= \frac{1}{2} \sum_{n=m}^{N-1} \{|f_{m-1}[n] + K_m b_{m-1}[n-1]|^2 + |K_m f_{m-1}[n] + b_{m-1}[n-1]|^2\} = 0 \\ K_m &= \frac{-\sum_{n=m}^{N-1} f_{m-1}[n] b_{m-1}[n-1]}{\frac{1}{2} \sum_{n=m}^{N-1} [|f_{m-1}[n]|^2 + |b_{m-1}[n-1]|^2]}; \quad m = 1, 2, \dots, M \end{aligned} \quad (26)$$

La minimización de  $K_m$  implica que los coeficientes AR calculados pueden predecir la señal hacia adelante y hacia atrás. Además, se observa que el denominador de (26) corresponde a una estimación mediante mínimos cuadrados de los errores de predicción hacia adelante y hacia atrás, la cual puede adquirir una forma recursiva como se expresa en (27).

$$DEN_m = DEN_{m-1} \left(1 - |\hat{K}_{m-1}|^2\right) - |f_{m-1}[m]|^2 - |b_{m-1}[N-1]|^2; \quad m = 1, 2, \dots, M \quad (27)$$

Los pasos para llevar a cabo la estimación de la PSD mediante el método de Burg son:

1. Inicializar con un orden  $m = 1$  y con los valores dados por (28).

$$\begin{aligned}
f_0[n] &= b_0[n] = x[n] \\
E_0 &= \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2 \\
DEN_0 &= 2E_0 \\
K_0 &= 0
\end{aligned} \tag{28}$$

2. Calcular el coeficiente de reflexión  $K_m$  mediante la ecuación (26) y aprovechando la expresión recursiva (27).
3. Obtener el parámetro  $\hat{a}_m[n]$  del modelo AR y el error  $\hat{E}_m$  mediante las expresiones de la recursión Levinson-Durbin (24) y (25).
4. Determinar los errores de predicción hacia adelante  $f_m[n]$  y hacia atrás  $b_m[n]$  para  $n = m + 1, \dots, N$  a través de las expresiones (23) de una estructura Lattice.
5. Regresar al paso 2 y aumentar el orden  $m$  en una unidad hasta alcanzar el orden  $M$  del modelo AR. Nótese que el arreglo que contiene a los parámetros  $\hat{a}_m[n]$  va incrementando de tamaño con cada iteración, quedando al final un arreglo  $\hat{a}_M[n]$  de tamaño  $M$ .
6. Calcular el PSD de la señal  $P_{xx}[k]$  mediante la expresión (29) que resulta similar a las expresiones (16) y (17) considerando el modelo AR. La adición del término  $F_s$  se realiza para representar la verdadera potencia de una señal analógica [48].

$$P_{xx}[k] = \frac{1}{F_s} \frac{E_M}{\left| 1 + \sum_{n=0}^{M-1} \hat{a}_M[n] e^{-j2\pi kn/N} \right|^2}, \quad k = 0, 1, \dots, N-1 \tag{29}$$

De esta forma, se observa que en el denominador de la expresión se realiza una operación muy similar a la DFT, salvo que la señal de entrada en este caso serán los coeficientes  $a_m$  del modelo AR. Asimismo, la carga computacional que involucra este algoritmo resulta ser de  $3NM - M^2 - 2N - M$  adiciones complejas, donde  $M$  es el orden del predictor AR, y  $3NM - M^2 - N + 3M$  multiplicaciones complejas, por lo que se requiere un mínimo espacio de memoria de aproximadamente  $3N + M + 2$  [48].

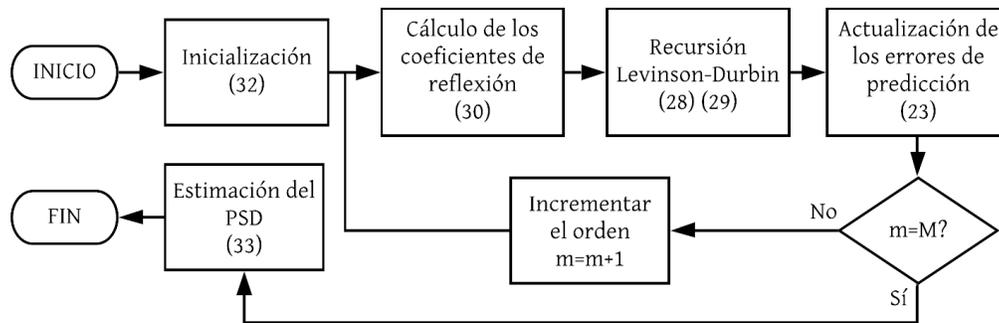


Figura 14. Diagrama de flujo del funcionamiento del método de Burg.

### 3.2.2.2.2 Método de Welch

La estimación no paramétrica de la PSD mediante el método de Welch resulta una implementación barata a nivel computacional que permite reducir el ruido en los espectros a cambio de una reducción en la resolución en frecuencia, esto gracias a la realización del promedio de periodogramas modificados.

Este método resulta similar al método de Bartlett, a diferencia de que este permite tanto el traslape de segmentos de datos como realizar un proceso de ventaneo previo al cómputo del periodograma, de forma que es posible describir su desarrollo a partir de los siguientes cuatro pasos [51] :

1. Partición de la secuencia de  $N$  datos en  $G$  segmentos o ramas.

$$x_g[n] = x[m + gD], \quad \begin{matrix} m = 0, 1, \dots, M - 1 \\ g = 0, 1, \dots, G - 1 \end{matrix}$$

Donde  $M$  es el número de puntos en cada segmento,  $D$  es un factor que determina el traslape entre segmentos y  $G$  es el número de segmentos. Nótese que en el caso de  $D = M$  no existe ningún traslape entre segmentos y, en cambio, si  $D = M/2$  se ocasiona un traslape del 50% entre segmentos de datos sucesivos.

2. Para cada segmento  $G$  se aplica una DFT con un ventaneo discreto  $w[n]$ . Este paso suele implementarse a través del algoritmo FFT para reducir el número de operaciones complejas por realizar.

$$X_g[k] = \sum_{n=gD}^{(M+gD)-1} w[n]x[n] e^{-j2\pi kn/N}, \quad g = 0, \dots, G - 1$$

3. Para cada segmento  $K$  se conforma el valor del periodograma modificado  $P_k(\Omega)$  de la DFT.

$$P_g[k] = \frac{|X_g[k]|^2}{W} \quad \text{donde } W = \sum_{n=0}^{M-1} w^2[n]$$

4. Finalmente, el promedio del periodograma  $P_g[k]$  entrega la estimación de Welch de la PSD  $P_{xx}[k]$ .

$$P_{xx}[k] = \frac{1}{G} \sum_{g=0}^{G-1} P_g[k]$$

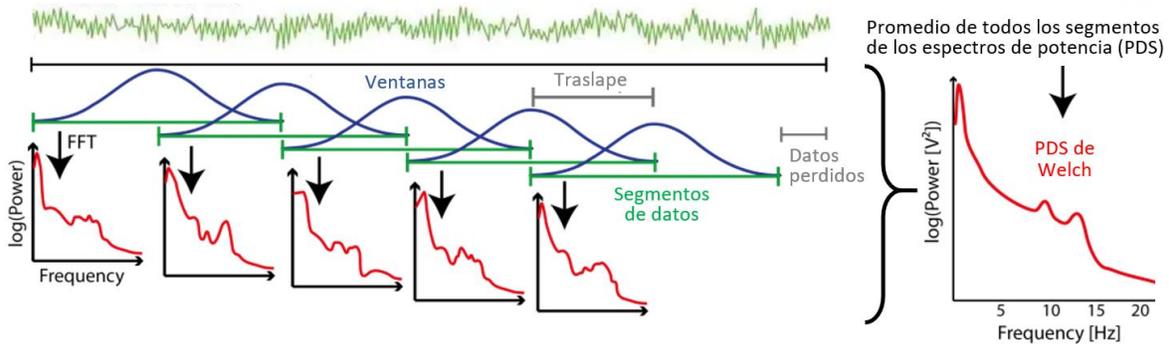


Figura 15. Ejemplo gráfico de la segmentación de una señal para la estimación de la densidad espectral de potencia a través del método de Welch. Imagen modificada de [86].

El número de operaciones que requiere este método se encuentra ligado al número de segmentos en los que se desea dividir al conjunto de datos  $N$  y el porcentaje de traslape que se busca realizar entre ellos. De esta forma, dado que segmento involucra la operación de una FFT cuyo número de operaciones son del orden  $O(G/2 \log_2 G)$ , el número de FFTs por realizar se expresa como  $\frac{2N}{G} - 1$  y el número total de operaciones resultaría en  $\left(\frac{2N}{G} - 1\right) \left(\frac{G}{2} \log_2 G\right)$  [50].

Semejante al análisis mediante la FFT, la estimación de la PSD de la señal, mediante cualquiera de los métodos mencionados previamente, permitirá conformar intervalos de frecuencia de interés cuyos promedios en las magnitudes de las componentes que los conforman, o bien una razón entre dichos promedios, servirán como *características* para cualquiera de los clasificadores de la sección 3.5.

### 3.2.3 Atributos de las señales fisiológicas

Con base en los fundamentos estudiados en la sección anterior, se procede al análisis de los atributos de cada una de las señales fisiológicas descritas en la sección 3.1, los cuales han demostrado, a lo largo de todos los estudios del Estado del Arte, que constituyen elementos clave para la detección de una emoción.

#### 3.2.3.1 Variabilidad de la Frecuencia Cardíaca (HRV)/ Variabilidad de la Frecuencia del Pulso (PRV)

El tiempo que transcurre desde que se presenta un latido cardíaco hasta que ocurre otro, no siempre es igual, y dependiendo de diversos factores, tanto externos como internos, este lapso puede variar. Dicho fenómeno fisiológico es denominado variabilidad de la frecuencia cardíaca (HRV) cuando se analiza a partir del ECG (sección 3.1.2) y variabilidad de la frecuencia de pulso (PRV) cuando es determinada a través de la PPG (sección 3.1.3). Esta herramienta proporciona información relevante para evaluar el estado y la actividad del SNA. Durante la implementación del ECG son empleadas técnicas específicas con el propósito de detectar los picos R de cada complejo QRS y posteriormente, determinar los intervalos RR. Por otro lado, al emplear la PPG se realiza la detección de los picos sistólicos con los cuales se calculan los intervalos pulso a pulso (IPP). Una vez obtenidos dichos parámetros, el análisis empleado es el mismo para ambas técnicas, por lo que, por cuestiones de simplicidad se hará referencia únicamente a HRV y los intervalos RR. A partir de los intervalos RR es posible determinar la frecuencia cardíaca de un individuo, la cual llega a adquirir un valor mínimo en reposo de 55 bpm para hombres y 58 bpm para mujeres según [52] junto con un valor máximo bajo situaciones de estrés o ejercicio de 210 bpm para hombres y 206 bpm para mujeres según [53], sin embargo, estos últimos resultados varían de acuerdo con la edad y la fisiología de la persona según su índice de masa corporal. Por ende, para este trabajo se considera una frecuencia cardíaca mínima de 40 bpm y máxima de 200 bpm.

El análisis de la HRV se realiza sobre una gráfica, la cual representa el valor de cada intervalo RR en función del tiempo. Este diagrama se denomina *tacograma* y se ilustra en la Figura 16. Cabe resaltar que la irregularidad de los intervalos implica una tasa de muestreo variable de la señal. Para su estudio se emplean diversos dominios de análisis, incluidos el tiempo y la frecuencia, así como métodos gráficos (no lineales). Estos procedimientos permiten obtener las características más relevantes para la detección de emociones.

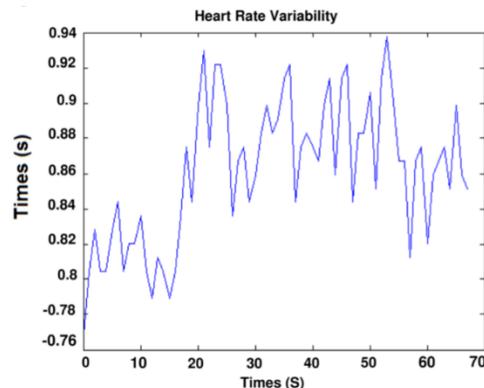


Figura 16. Tacograma

El análisis del *dominio del tiempo* consiste en el cálculo de parámetros que pueden obtenerse de dos formas distintas:

- a) Mediciones directas de los intervalos RR o la frecuencia cardíaca instantánea.
- b) Diferencias entre los intervalos RR.

En un registro continuo del ECG se determina el ritmo cardíaco en cualquier instante, lo cual refleja el intervalo entre complejos QRS adyacentes (intervalo RR). Realizando un registro prolongado, típicamente en un período de 24 h, es posible obtener parámetros estadísticos más complejos, sin embargo, para dispositivos portátiles y aplicaciones en tiempo real no es implementado. Por lo tanto, diversos parámetros se obtienen analizando segmentos más pequeños, los cuales permiten comparar el comportamiento de la HRV en situaciones o actividades diferentes.

Tabla 6. Parámetros de la HRV en el dominio del tiempo

Parámetro	Unidades	Descripción
<b>RR media</b>	ms	Es la media de los intervalos RR.
<b>NN50</b>		Número de intervalos RR adyacentes que difieren por más de 50 ms en el registro completo.
<b>pNN50</b>	%	NN50 dividido por el número total de intervalos RR.
<b>MAX-MIN</b>	ms	Diferencia entre el intervalo RR más corto y el más largo.
<b>SDNN</b>	ms	Desviación estándar de todos los intervalos RR.
<b>SDNN index</b>	ms	Media de las desviaciones estándar de todos los intervalos RR en todos los segmentos (5 min) de un registro completo (> 24 h).
<b>SDANN</b>	ms	Desviación estándar de los promedios de los intervalos RR en todos los segmentos (5 min) de un registro completo (> 24 h).
<b>RMSSD</b>	ms	Raíz de la media cuadrática (RMS) de las diferencias entre intervalos RR adyacentes.
<b>SDSD</b>	ms	Desviación estándar de las diferencias entre intervalos RR adyacentes.

El cálculo del promedio y de las desviaciones estándar es realizado mediante las ecuaciones (2) y (4) descritas en la sección 3.2.1.1, respectivamente. Para la RMSSD se implementa una variación de la ecuación (3), la cual se expresa a continuación en la ecuación (30). Por otro lado, el parámetro pNN50 es calculado con (31).

$$RMSSD = \sqrt{\frac{1}{N-1} \sum_{j=1}^{N-1} (RR_{j+1} - RR_j)^2} \quad (30)$$

$$pNN50 = \frac{NN50}{N-1} 100\% \quad (31)$$

Dentro del dominio del tiempo también se encuentra el análisis de la gráfica de Poincaré, el cual consiste en un método geométrico y no lineal para evaluar la dinámica de la HRV. Cada intervalo RR se representa en función del intervalo RR anterior, donde cada punto en la gráfica es definido por un par de intervalos sucesivos. La naturaleza no lineal del sistema cardíaco posiciona a la gráfica de

Poincaré como una herramienta sumamente útil en el análisis de la HRV. Además de una interpretación visual simple, proporciona información resumida y detallada sobre el comportamiento del corazón [54].

Hay dos formas de evaluar una gráfica de Poincaré:

- a) *De forma cualitativa*: utilizando el patrón visual definido por la forma, tamaño y posición de los grupos de puntos [55].
- b) *De forma cuantitativa*: calculando los índices de desviación estándar (SD, por sus siglas en inglés) del diagrama [54].
  - **SD1**: variabilidad entre latidos a corto plazo.
  - **SD2**: variabilidad entre latidos a largo plazo.
  - **SD1/SD2**: relación de la variación del intervalo corto y la variación del intervalo largo.

En la Figura 17 se muestra la gráfica de Poincaré de una persona saludable. El eje horizontal representa el primer intervalo RR ( $RRI_n$ ), mientras que el eje vertical representa el intervalo consecutivo ( $RRI_{n+1}$ ).

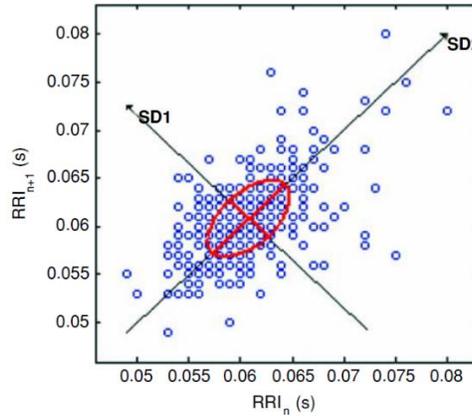


Figura 17. Gráfica de Poincaré de una persona saludable [54].

Como se mencionó anteriormente, el análisis cuantitativo se lleva a cabo mediante el cálculo de las desviaciones estándar de las distancias entre los puntos graficados y las rectas mostradas en la Figura 17, cuyas ecuaciones se definen a continuación:

$$RRI_{n+1} = RRI_n \quad (32)$$

$$RRI_{n+1} = -RRI_n + 2(m_{RR}) \quad (33)$$

donde  $m_{RR}$  es la media de todos los intervalos RR.

La ecuación (32) representa la pendiente del eje longitudinal, mientras que la ecuación (33) define la pendiente transversal, la cual es perpendicular a la recta definida por (32).

En el *dominio de la frecuencia* se calcula la PSD de la HRV. Este análisis permite comprender la naturaleza de las oscilaciones presentes en el registro de intervalos RR y, además, refleja la distribución de potencia en función de la frecuencia. No obstante, a raíz del requerimiento de emplear una tasa de muestreo fija para utilizar varias de las técnicas que estiman la PSD y dado que la naturaleza de la señal HRV no cumple con ésta, es necesario utilizar procedimientos de interpolación a fin de conformar una nueva señal que cumpla con dicho criterio. Las técnicas más utilizadas para cumplir

este propósito son la técnica de interpolación por trazadores cúbicos y por polinomios de Lagrange cuyo fundamento teórico puede consultarse en [56] y [57].

Cada intervalo de frecuencias presente en la HRV es asociado con determinados comportamientos fisiológicos. La actividad del SNS se refleja en el intervalo de alta frecuencia, mientras que la del SNP se asocia con el intervalo de frecuencias bajas.

Tabla 7. Parámetros de la HRV en el dominio de la frecuencia

Parámetro	Descripción	Frecuencia
<b>Medidas absolutas</b>		
Potencia total	Varianza de todos los intervalos RR	Aproximadamente $\leq 0.4$ Hz
ULF	Ultra baja frecuencia	$\leq 0.003$ Hz
VLF	Muy baja frecuencia	0.003 Hz – 0.04 Hz
LF	Baja frecuencia	0.04 Hz – 0.15 Hz
HF	Alta frecuencia	0.15 Hz – 0.4 Hz
<b>Medidas relativas</b>		
LF Norm	Baja frecuencia normalizada	$LF / (Potencia\ Total - VLF) \times 100$
HF Norm	Alta frecuencia normalizada	$HF / (Potencia\ Total - VLF) \times 100$
LF/HF	Relación de baja y alta frecuencia	--

En la Tabla 7 se dividen las componentes espectrales en dos grupos, medidas absolutas y medidas relativas. Las primeras se representan con valores absolutos de potencia en milisegundos cuadrados ( $ms^2$ ) y las relativas en unidades normalizadas (n.u.). Estas últimas resaltan el comportamiento equilibrado de las dos ramas del SNA y es reducida su variación cuando se presentan cambios en la potencia total.

Cuando se registran períodos a corto plazo (2 a 5 min), las componentes espectrales presentes en el registro son los de muy baja frecuencia (VLF, por sus siglas en inglés), baja frecuencia (LF, por sus siglas en inglés) y alta frecuencia (HF, por sus siglas en inglés). Únicamente cuando el registro es en un período de 24 h, se incluye la componente de frecuencia ultrabaja (ULF, por sus siglas en inglés), además de las componentes VLF, LF y HF.

### 3.2.3.2 Atributos de la Respuesta Galvánica de Piel

De acuerdo con la información recopilada a partir del Estado del Arte, las características comúnmente extraídas de la señal EDA se presentan en la Tabla 8.

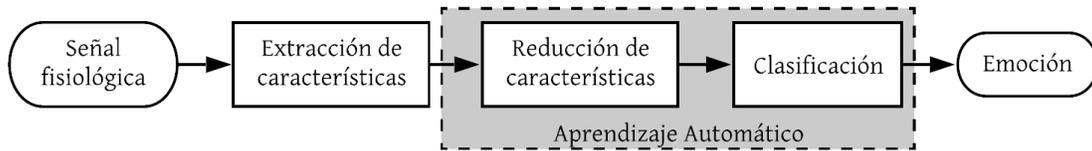
Tabla 8. Características extraídas de la señal EDA.

	Características
<b>Dominio del tiempo</b>	Características estadísticas: promedio, desviación estándar, min, max, pendiente, tiempo de subida promedio, media de la derivada, etc.
<b>Dominio de la frecuencia</b>	10 espectros de potencia en las bandas de 0-2.4 Hz.
<b>Características SCL</b>	Características estadísticas, grado de linealidad.
<b>Características SCR</b>	Número de segmentos SCR identificados, suma de magnitudes de sobresalto y duraciones de respuesta SCR, área bajo los SCR identificados.

Existen diversas técnicas dedicadas a la extracción de las componentes SCL y SCR, dentro de las más empleadas se encuentra el método presentado por [58], en el cual se reduce la señal a 20 muestras/s, se realiza la diferenciación y posterior convolución con una ventana de Bartlett de 20 puntos. La ocurrencia de un SCR se detecta al identificar dos cruces por cero consecutivos, de negativo a positivo y de positivo a negativo. La amplitud de SCR se obtiene registrando el valor máximo entre

los cruces por cero. Aquellos SCR con amplitudes menores al 10% de la amplitud máxima del segmento son excluidos.

### 3.3 Aprendizaje Automático en los sistemas de detección de emociones



Una vez extraídas las características de diversas señales fisiológicas a través de técnicas de procesamiento digital de señales, es necesario darles un tratamiento por medio de un sistema que detecte la emoción evocada. Si bien, existe una relación entre los datos fisiológicos y las emociones expresadas al momento de registrarlos, no se reporta un algoritmo que pueda relacionar ambos con exactitud. Para solventar este problema, en el Estado del Arte se propone el uso de técnicas de reducción de características y clasificación, pues estas proveen un sistema que aprende por sí mismo a resolver una tarea de la mejor forma posible sin necesidad de implementar una serie de reglas específicas que lo hagan, estos temas son englobados en un área de la inteligencia artificial llamada *Aprendizaje Automático* (ML, por sus siglas en inglés).

Para introducirse al Aprendizaje Automático es preciso conocer algunos conceptos fundamentales que ayudarán a la comprensión del tema. Los algoritmos de ML son abarcados dentro de una gran variedad de áreas con enfoques diferentes, las cuales definirán el tipo de tareas que pueden resolver, la forma en que lo harán y, en algunos casos, el tipo de entradas que necesitan. Por este motivo es necesario conocer, de forma general, los distintos tipos de aprendizaje existentes y sus aplicaciones, así como las ramas del conocimiento abarcadas dentro de éstos. Además, es importante definir aquellas herramientas que son útiles para la implementación del sistema, como es el caso de las técnicas utilizadas para entrenamiento y validación de un algoritmo. De igual manera, al momento de realizar pruebas de desempeño, es preciso tener una valoración numérica que lo refleje, por lo que se darán a conocer el término de exactitud, la cual es una herramienta ampliamente utilizada en ML para la evaluación de algoritmos.

#### 3.3.1 Fundamentos de Aprendizaje Automático

Cuando es necesario resolver un problema específico, el código utilizado en los modelos de programación clásicos debe contener una serie de reglas que dicten el preprocesamiento que se debe dar a los datos de entrada para obtener la salida deseada. En el Aprendizaje Automático se busca que una máquina sea capaz de aprender y establecer dichas reglas por sí sola, es decir, no deberá ser programada específicamente para resolver una tarea, en su lugar, se le presentarán distintos ejemplos de ésta y posteriormente ajustará los parámetros de un grupo de operaciones predefinidas. A dichos procesos se les denomina *entrenamiento* y *aprendizaje*, respectivamente. Finalmente, con dicho ajuste, el sistema será capaz de realizar la tarea para la cual fue programado. Cabe destacar que el funcionamiento específico de cada sistema dependerá del tipo de aprendizaje y el algoritmo por sí mismo, por lo que una descripción concreta será abordada más adelante en su respectiva sección.

Existen diversos algoritmos que están considerados dentro del Aprendizaje Automático, estos necesitan una serie de datos de entrada, normalmente llamados *características* y, en algunos casos, ejemplos de la salida esperada. De igual manera, es necesario contar con una medición que indique qué tan bien se está realizando una determinada tarea, esto se realiza por medio de una *función de costo*, la cual será una señal de retroalimentación que ayuda a elegir y ajustar el modelo.

Antes de proseguir, es importante ahondar más en la naturaleza de los datos de entrada, además de definir un concepto que será requerido para comprender de mejor manera cada tema: la *dimensionalidad*. Cada uno de los datos de entrada de los algoritmos de Aprendizaje Automático estarán compuestos por vectores  $\vec{x}$  con un determinado número de características en su interior, los cuales

definirán la dimensión de dicho vector. Por ejemplo, para predecir un estado afectivo por medio de señales fisiológicas, podrían solicitarse  $D$  características distintas como el HRV, pNN50, etc. por lo que la dimensión de los vectores de entrada será de igual manera  $D$ , tal como se muestra en la ecuación (34).

$$\vec{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \quad \text{donde } d = 0, 1, 2, \dots, D - 1 \quad (34)$$

Cabe mencionar que las características  $x_d$ , si bien pueden ser completamente independientes, en algunas ocasiones es conveniente agregar al vector  $\vec{x}$  distintas versiones de una misma característica o combinaciones de ellas, como productos, formas cuadráticas, etc. Esto tiene como objetivo mejorar el proceso de aprendizaje del sistema tomando en consideración que las características no tendrán un comportamiento necesariamente lineal.

Como anteriormente se mencionó, los algoritmos con un enfoque en Aprendizaje Automático han sido abordados desde distintas ramas del conocimiento, o inclusive, combinaciones de ellas. Esto definirá la forma en que desempeñará la tarea que le será asignada. A continuación, se enlistan algunos modelos con el objetivo de dar una perspectiva más general de la variedad de áreas existentes desde las que se pueden abordar los algoritmos de Aprendizaje Automático.

- **Modelos probabilísticos y estadísticos:** se utilizan los principios de la probabilidad y estadística para el análisis de los datos.
- **Modelos lineales:** tratamiento de los datos a través de funciones lineales con parámetros ajustables.
- **Métodos de Kernel:** dentro de este campo se hace uso de funciones Kernel, las cuales son funciones matemáticas que proyectan los datos en un espacio de características de alta dimensión para incrementar la eficiencia computacional. Al realizar ese mapeo de la información se tiene un límite de decisión expresada en un hiperplano, donde se busca maximizar la distancia entre dicho hiperplano y los puntos de datos en cada clase.
- **Árboles de decisión:** este tipo de modelos cuenta con una estructura de diagrama de flujo, la cual busca clasificar los datos de entrada.
- **Redes neuronales:** modelo computacional inspirado en su contraparte biológica. Consiste en grupos de unidades interconectadas entre sí a través de capas, donde cada una somete a sus datos de entrada a operaciones para obtener un valor de salida, el cual será el valor de entrada para otras unidades.

Cabe destacar que dentro de los sistemas de detección de emociones abordados en el Estado del Arte no hubo un área del conocimiento predilecta, pues se utilizaron algoritmos pertenecientes a todos los campos mencionados. En cuanto a desempeño, se mostró un mejor rendimiento en los modelos probabilísticos, estadísticos, lineales y redes neuronales.

Los algoritmos de Aprendizaje Automático no se dividen únicamente por las ramas del conocimiento que abordan, pues eso únicamente definirá la forma en que realizarán la tarea que le sea asignada. El aprendizaje es otra manera de categorizar dicha área, pues ayuda a comprender, de forma general, el tipo de entradas y salidas del algoritmo y, salvo algunas excepciones, la función que desempeñará dentro de un sistema. Los principales tipos de aprendizaje son el *aprendizaje supervisado* y el *aprendizaje no supervisado*.

### 3.3.2 Aprendizaje supervisado

El aprendizaje supervisado adopta su nombre por la forma en que se realiza el entrenamiento del algoritmo, pues consiste en presentarle ejemplos de tareas cuyas entradas y salidas son conocidas, es decir, su aprendizaje es guiado por el diseñador. El objetivo de estos sistemas es que, una vez entrenados, al ingresar nuevas características de las cuales se desconoce la salida, dichos sistemas sean capaces de predecirla. Existen excepciones como es el caso del Discriminante lineal de Fisher abordado en la sección 3.4.3, pues a pesar de pertenecer a este aprendizaje, su funcionamiento es más semejante al descrito en el aprendizaje no supervisado.

Dependiendo de la naturaleza de la salida, los algoritmos de aprendizaje supervisado pueden ser utilizado para realizar *regresiones* (si sus salidas son continuas) o *clasificaciones* (si sus salidas son discretas).

- **Regresión:** En esta clase de algoritmos, dado un vector de características de entrada, se tiene una predicción expresada como un valor escalar continuo, tal como se ejemplifica en la Figura 18. En ésta se muestra una serie de datos por medio de puntos azules dispersos en un plano, donde el eje de las abscisas representa el valor de una sola componente del vector de características y el eje de las ordenadas su resultado escalar. El algoritmo de regresión ajusta el modelo de modo que, al tener una nueva entrada de la que se desconoce su salida, se tenga una predicción asociada a la tendencia de los datos con los que fue entrenado, la cual queda expresada gráficamente por una línea negra.

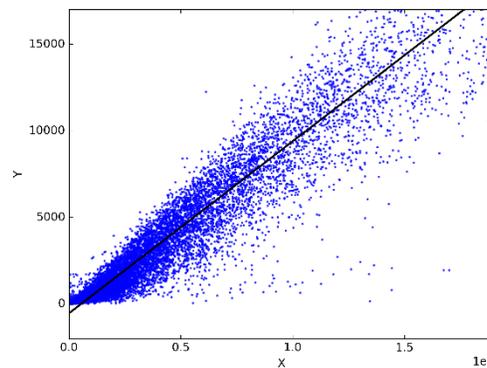


Figura 18. Ejemplo de regresión lineal. Imagen obtenida de [91].

- **Clasificación:** En esta clase de sistemas se busca que, a través de un vector de características de entrada, el algoritmo sea capaz de predecir una salida discreta como resultado del ajuste de sus parámetros durante el entrenamiento. A cada salida posible se le denomina *clase*, *etiqueta*, *patrón* u *objetivo*. Es importante señalar que existe una diferencia al realizar clasificaciones *binarias* (dos clases) o *multiclase* (mayor a dos clases), pues existen algoritmos que no les es posible realizar directamente este último tipo de clasificación. En la Figura 19 se ejemplifica gráficamente la distribución de datos en una clasificación, pues se muestran puntos cuyas coordenadas representan dos de las componentes de un vector de características y, por otro lado, la clase a la que pertenecen está definida por un color. Este tipo de algoritmo ajusta sus parámetros de forma que, al tener una nueva entrada de la que se desconoce su clase, ésta pueda asociarse a una de las ya existentes.

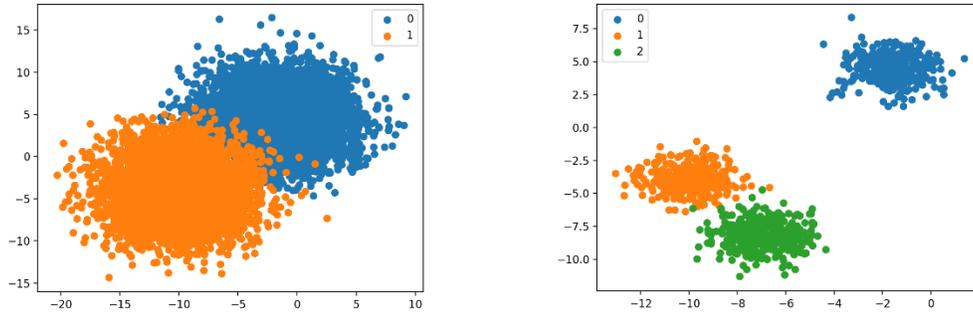


Figura 19. Clasificación binaria y multiclase. Imágenes obtenidas de [89] y [90].

Para la detección de emociones utilizando métodos de Aprendizaje Automático, los algoritmos de aprendizaje supervisado enfocados a clasificaciones son el eje principal de la etapa de detección, pues es necesario un sistema que tome las características provenientes de las técnicas de procesamiento digital de señales y las adjudique a algún estado emocional en específico, siendo este precisamente la clase que se busca predecir. Es por este motivo que se dedica la sección 3.5 para detallar el algoritmo denominado *Análisis Lineal Discriminante*, el cual es elegido de entre aquellos clasificadores con mejor desempeño demostrado en el Estado del Arte para implementarse en la presente tesis.

### 3.3.3 Aprendizaje no supervisado

La diferencia entre el aprendizaje supervisado y el aprendizaje no supervisado radica en la forma en que se realiza el entrenamiento, pues en este último se utilizan vectores de características cuyas salidas no son conocidas, por lo que, en esta ocasión, no se le informa al algoritmo el resultado que debería arrojar y, por tanto, su aprendizaje no está supervisado. Una de sus tareas se basa en identificar rasgos en común dentro de las características de entrenamiento y, con la llegada de nuevos vectores de características, realizar la identificación de dichos atributos para asociarlas a un determinado grupo (parecido a la tarea de clasificación del aprendizaje supervisado) o detectar anomalías (datos que se salgan de las condiciones normales establecidas por los vectores de entrenamiento). Debido a la capacidad que tiene este aprendizaje de extraer información de las características de entrada, también se incluyen en él las funciones de *reducción dimensional* y *selección de características*, las cuales son técnicas que aprovechan esta cualidad para reducir la información de los vectores.

- **Agrupamiento:** Al igual que la clasificación en el aprendizaje supervisado, en el agrupamiento se busca crear precisamente grupos a los cuales asociar los vectores de características. Para ello, el algoritmo ajusta sus parámetros identificando rasgos en común entre los vectores utilizados en el entrenamiento, para que, al momento de ingresar nuevas entradas, puedan asociarse a dichos grupos, tal como se ejemplifican en la Figura 20. En ésta se muestra una serie de puntos que representan a los vectores de características en la etapa de entrenamiento, que, como puede observarse, no tienen una clase asociada que permita discriminarlos, por lo que el algoritmo deberá encontrar una relación entre dichos puntos y agruparlos, lo que se traduce a realizar un ajuste de sus parámetros internos. Una vez realizado, cuando se ingrese

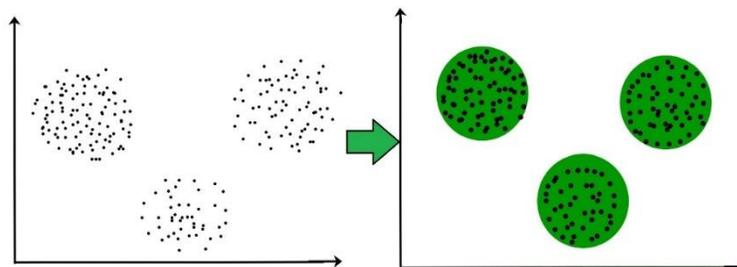


Figura 20. Agrupamiento de datos. Imagen obtenida de [92].

un nuevo vector de características se tendrá como salida el grupo al que más afinidad tenga. La desventaja de esta clase de algoritmos surge de la falta de información que permita validar los resultados, pues como se menciona previamente, no se cuenta con una salida conocida que permita realizar un contraste.

- **Detección de anomalías:** Se utiliza una serie de características de entrenamiento consideradas ordinarias para establecer un margen que permita detectar si algún nuevo dato de entrada no está dentro de los parámetros normales. Existe cierto parecido entre este tipo de algoritmos y la clasificación binaria del aprendizaje supervisado, pues puede considerarse que todo el conjunto de entrenamiento pertenece a una misma clase y, posteriormente, se realiza una clasificación para identificar si el nuevo dato pertenece o no a dicha clase. Sin embargo, es necesario resaltar que para realizar la detección no se utiliza información sobre la clase a la que pertenecen las características, pues en caso contrario se necesitarían también ejemplos de las características consideradas anómalas para asociarlas a dicha clase al momento de la clasificación. Dada esta cualidad, se considera a la detección de anomalías dentro del aprendizaje no supervisado.
- **Reducción dimensional y selección de características:** Ambas técnicas consisten en reducción de datos, pues su tarea es disminuir el tamaño de los vectores de características de entrada y evitar que se pierda información útil en la medida de lo posible, ya sea desechando características dada su redundancia o debido a la poca información que aportan para resolver la tarea en cuestión, o bien, comprimiéndolas por medio de transformaciones lineales. De esta manera, se busca disminuir el costo computacional que involucraría el tratamiento de todas las características extraídas, además, puede ser utilizado para facilitar la representación visual de éstas. Recordando que los datos de entrada se componen de vectores de características  $\vec{x}$  de dimensión  $D$ , las salidas de esta clase de algoritmos serán vectores  $\vec{y}$  de dimensión  $K$ , donde  $K < D$ , en los que se mantendría la mayor parte de la información de los datos originales, tal como se muestra en la ecuación (35). Posteriormente, dichos vectores de salida servirán como entrada de algún algoritmo de clasificación y/o podrán representarse en alguna gráfica en caso de que la reducción sea menor o igual a tres dimensiones.

$$\vec{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \rightarrow \vec{y} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} \text{ donde } \begin{matrix} d = 0, 1, 2, \dots, D - 1 \\ k = 0, 1, 2, \dots, K - 1 \end{matrix} \quad (35)$$

Los algoritmos de reducción dimensional y selección de características son muy útiles para los sistemas de detección de emociones, pues al existir una gran variedad de características que pueden ser extraídas de las señales fisiológicas, no existe garantía de que todas en conjunto aporten un mejor desempeño al sistema, por lo que es importante garantizar que la información aportada al algoritmo de clasificación sea útil y comprimida, pues esto ayudará a disminuir la carga computacional y aumentar el rendimiento del mismo. Debido a que algunos de los sistemas en el Estado del Arte utilizan esta clase de algoritmos, se dedicó la sección 3.4 a describir los más utilizados.

### 3.3.4 Subajuste y sobreajuste

Es necesario que un sistema con Aprendizaje Automático sea *óptimo* y *generalizado*, es decir, debe realizar su tarea de manera satisfactoria incluso con vectores de características con los que no haya sido entrenado previamente. Para ello deben tenerse en cuenta dos problemas comunes: el *subajuste* y *sobreajuste*.

- Cuando un modelo está subajustado presenta un rendimiento pobre al momento de trabajar con las características que se le proporcionen, ya sean aquellas con las que haya sido entrenado

o con características completamente nuevas, pues al no estar ajustado correctamente, no desempeña su tarea de forma adecuada.

- Cuando se presenta un sobreajuste del modelo, este comienza a aprender patrones específicos del conjunto de entrenamiento, lo que puede ser engañoso, ya que trabajaría perfectamente con aquellos valores con los que fue entrenado, pero provocaría resultados erróneos cuando se tratara de nuevos datos. Una solución es contar con un gran número de ejemplos de entrenamiento con el objetivo de mantener un modelo generalizado, sin embargo, no siempre se cuenta con tal cantidad de información y no siempre es la mejor solución. Lo común es aplicar técnicas que permitan maximizar el ajuste del modelo sin que llegue a excederse, tomando únicamente los patrones más representativos de los datos de entrenamiento.

En la Figura 21, puede observarse un ejemplo de estos problemas en la clasificación, pues si se intenta entrenar a un modelo que trabaja con funciones lineales mediante un conjunto de entrenamiento con un umbral de tendencia cuadrática, por más que se ajuste a dicho conjunto, no podrá clasificar los nuevos datos de forma correcta. Por otro lado, si se excede en el ajuste, el modelo aprenderá patrones específicos del conjunto de entrenamiento y no será generalizado, lo que resultará en un pobre rendimiento con datos que no pertenezcan al conjunto con el que fue entrenado.

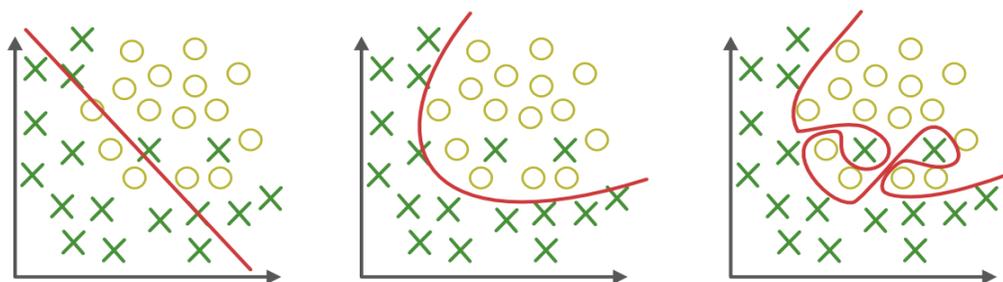


Figura 21. Subajuste, ajuste ideal y sobreajuste. Imagen adaptada de [97].

### 3.3.5 Entrenamiento y validación

Como se ha visto anteriormente, un sistema de Aprendizaje Automático consta de dos etapas: entrenamiento, donde se realizarán ajustes en los parámetros del algoritmo y la puesta en funcionamiento o prueba, donde se ejecutará la tarea para la que fue entrenado. En la etapa de entrenamiento es necesario utilizar un conjunto de características obtenido con antelación, sin embargo, para corroborar su funcionamiento, en la etapa de prueba se debe evaluar al sistema con características completamente desconocidas, es decir, con las que no haya sido entrenado previamente. Esto se realiza por medio de una comparación entre la clase de los vectores de características de entrada y las predicciones realizadas por el algoritmo. Para realizar la evaluación existen algunas técnicas de valoración como lo puede ser la *exactitud*, la cual nos indica el porcentaje de clasificaciones realizadas de forma exitosa.

En caso de utilizar la misma información para entrenar y comprobar los resultados de un modelo, se tendría una percepción del rendimiento bastante sesgada, pues es probable que el sistema aparente tener un buen desempeño y, al momento de probarlo con nuevos datos, suceda lo contrario, lo que indicaría un posible sobreajuste del modelo. Es por este motivo que para realizar la evaluación es necesario separar la información en tres categorías: el conjunto de entrenamiento, el conjunto de validación y el conjunto de prueba.

La razón por la que es necesario dividir los datos en tres conjuntos (entrenamiento, validación y prueba) en lugar de solo dos (entrenamiento y prueba) es por un proceso post entrenamiento para elegir el modelo a implementar. Cuando se comienza a diseñar un sistema no existe un modelo único de aprendizaje, pues no hay reglas específicas para saber cuál tendrá un mejor desempeño para resolver la tarea en cuestión. Por este motivo, es necesario contar inicialmente con distintos algoritmos y/o

múltiples configuraciones de un mismo algoritmo, donde éstas últimas se obtienen modificando sus respectivos parámetros de diseño, también llamados *hiperparámetros*, como lo puede ser el valor de  $k$  en el algoritmo kNN o la dimensión a la que se busca reducir los vectores por medio de FLD. Posteriormente, cada sistema será entrenado por el conjunto de entrenamiento y después evaluado con los datos del conjunto de validación, eligiendo así el sistema con mejor rendimiento. Es importante destacar que, a pesar de no ajustar directamente los parámetros de los algoritmos con este conjunto de validación, la selección del modelo sigue siendo parte del proceso de aprendizaje, por lo que puede resultar en un sobreajuste a dicho conjunto en caso de utilizarlo para evaluar el modelo de forma definitiva. Es por este motivo que existe el conjunto de prueba, el cual servirá para tener una medida *generalizada* del desempeño del modelo seleccionado. Es una práctica común que para este último paso se realice de nuevo el entrenamiento del modelo tomando todos los datos que no pertenezcan al conjunto de prueba.

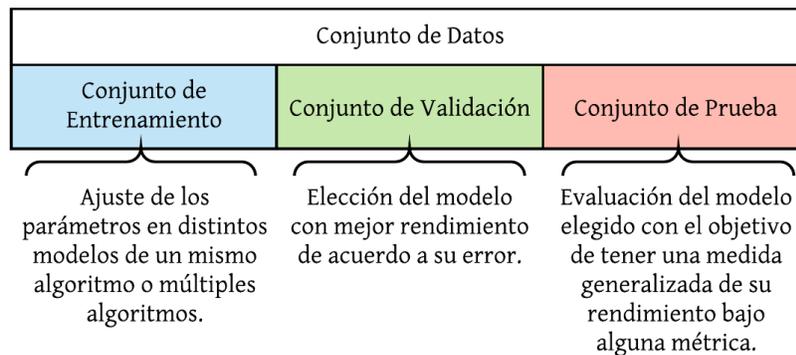


Figura 22. Divisiones del conjunto de datos y sus funciones

Existen distintas formas en las cuales pueden ser divididos los datos con los que será entrenado y validado el sistema. Cada una de ellas varía su utilidad dependiendo de la cantidad de información disponible al momento de entrenar un algoritmo:

- **Validación simple o validación por método de retención (simple hold-out validation):** Se dividen los datos en un conjunto de entrenamiento (60% de los datos), un conjunto de validación y un conjunto de prueba (20% de los datos cada uno). Esta validación no es recomendable cuando se cuenta con un reducido número de datos, pues los conjuntos de prueba y validación serán muy pequeños y no serán del todo representativos de la información que se busca procesar.
- **Validación cruzada de  $k$  iteraciones (k-fold cross-validation):** Se particionan los datos disponibles en  $k$  partes de igual tamaño. Para cada partición  $i$  (conjunto de validación), se entrena el modelo con las  $k - 1$  particiones restantes (conjunto de entrenamiento) y posteriormente se evalúa con la partición  $i$ . Este proceso se realiza de manera iterativa, tal como se muestra en la Figura 23. La puntuación final será el promedio de las  $k$  puntuaciones obtenidas. Este método es útil cuando el rendimiento del modelo muestra una varianza significativa en el conjunto de validación. Al igual que la validación simple, es preferible utilizar un conjunto de prueba independiente de los datos.
- **Validación cruzada de  $k$  iteraciones con barajeo (iterated k-fold cross-validation with shuffling):** Consiste en aplicar el mismo principio que la validación de  $k$  iteraciones, sin embargo, en cada iteración se hace un reordenamiento aleatorio de los datos antes de dividirlos. El puntaje nuevamente es el promedio de todas las puntuaciones obtenidas en las  $k$  iteraciones. Esta validación es útil cuando se tiene una muy pequeña cantidad de datos y se quiere tener una evaluación del modelo lo más precisa posible, sin embargo, este proceso puede implicar que haya datos con los que no se evalúe el modelo.

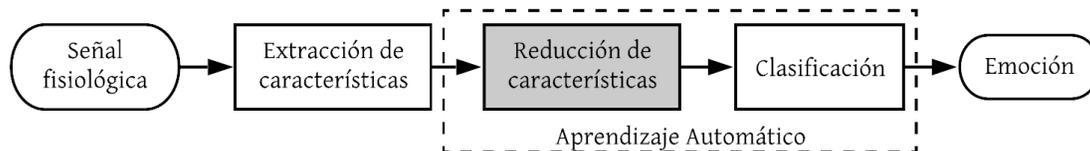
- **Validación cruzada dejando uno afuera (leave-one-out cross-validation):** Es muy parecida a la validación de  $k$  iteraciones, sin embargo, no se realiza una partición del conjunto de datos en grupos, en su lugar el entrenamiento se realiza con todo el conjunto exceptuando un dato. Dicho de otra manera, de  $k$  datos totales se realiza el entrenamiento con  $k - 1$  y se evalúa con un solo dato a lo largo de  $k$  iteraciones. Este método evita que haya varianza al momento de evaluar el modelo con el conjunto de validación, pues se analizará su rendimiento con prácticamente la totalidad del conjunto de datos (exceptuando el conjunto de prueba).

	Conjunto de Datos				Conjunto de Prueba
iteración 1	Validación	Entrenamiento	Entrenamiento	Entrenamiento	
iteración 2	Entrenamiento	Validación	Entrenamiento	Entrenamiento	
iteración 3	Entrenamiento	Entrenamiento	Validación	Entrenamiento	
iteración 4	Entrenamiento	Entrenamiento	Entrenamiento	Validación	

Figura 23. Validación cruzada de  $k$  iteraciones.

En los estudios contenidos en el Estado del Arte, la validación cruzada dejando uno afuera fue la más utilizada.

### 3.4 Algoritmos para la reducción de características



Como anteriormente se mencionó, el algoritmo de clasificación es una parte fundamental de un sistema de detección de emociones, pues por medio de éste será posible predecir la emoción evocada utilizando un vector de características obtenidas de las señales fisiológicas, las cuales son valores que describen dicha señal. Es común creer que mientras mayor sea el número de características obtenidas, mayor será el rendimiento del algoritmo de clasificación, pues dispondrá de más información para realizar su tarea, sin embargo, esto no es del todo cierto, pues un gran número de características de entrada puede involucrar una mayor variabilidad en los datos y, por tanto, empeorar la efectividad del clasificador. De igual manera, un incremento en el número de entradas puede disminuir el rendimiento en tiempo del algoritmo, pues habrá necesidad de realizar más operaciones tanto en su entrenamiento como en su puesta en funcionamiento. Es por este motivo que existen las técnicas de reducción de características, cuya función es disminuir la cantidad de datos sin perder, en la medida de lo posible, la información que aportarán al clasificador. Aun así, es necesario aclarar que su implementación es opcional y se recomienda realizar pruebas preliminares sin él, pues si se cuenta inicialmente con un espacio de características adecuado, el clasificador podría tener un buen rendimiento utilizando los vectores completos y, por tanto, la implementación del algoritmo de reducción podría traer una mejora imperceptible que no justificaría su tiempo de desarrollo.

Estas técnicas se dividen en dos clases de algoritmos, la *reducción dimensional* y la *selección de características*.

- Los métodos de *reducción dimensional* consideran toda la información de los vectores de características del conjunto de entrenamiento (incluida la más “ruidosa”) y crean un nuevo espacio de características de menor dimensión por medio de alguna técnica, como lo puede ser

una transformación lineal. Una vez obtenidos los parámetros necesarios para realizar dicha reducción, será posible comprimir los vectores de características con los que será entrenado el clasificador, así como los nuevos datos que se busquen clasificar durante su puesta en funcionamiento.

- En la *selección de características* se crea un nuevo conjunto de datos de entrenamiento haciendo omisión de aquellas características que puedan resultar menos útiles de acuerdo con la valoración realizada por alguna función objetivo, también llamada función de criterio. Este nuevo conjunto de entrenamiento con vectores de características de menor dimensión será utilizado para entrenar al clasificador. Una vez que se busque clasificar nuevos datos, solo será necesario desechar aquellas características que fueron indicadas por el algoritmo de selección.

Los algoritmos de selección de características pueden dividirse en dos grupos dependiendo de la función objetivo que se utilice:

- **Filtros:** evalúa las características de los vectores del conjunto de entrenamiento utilizando medidas de distancia o de correlación. Esto lo hace independiente del algoritmo de clasificación. Generalmente son de rápida ejecución, sin embargo, es necesario seleccionar arbitrariamente el tamaño de los nuevos vectores de características, pues suele escoger el tamaño original como la solución óptima.
- **Envoltorios:** se utiliza un clasificador para evaluar los subconjuntos resultantes de la reducción, por ejemplo, mediante la exactitud del algoritmo obtenida en el conjunto de prueba. Tienden a lograr mejores resultados, pero son de lenta ejecución.

Los sistemas expuestos en el Estado del Arte utilizaron reductores de características para optimizar su proceso de clasificación, siendo los más efectivos aquellos señalados en la Tabla 3. Es por este motivo que resulta conveniente analizar cada uno de ellos y considerar su integración al diseño del detector de emociones. En la Figura 24 se muestran de forma simplificada la integración de cada algoritmo al sistema, siendo SFS un selector de características, SFFS una variante de SFS que mejora la selección de características mediante una modalidad de eliminación, pLDA un algoritmo que permite reducir dimensionalmente un espacio mediante una transformación lineal y, finalmente, SFFS-FP, un híbrido entre un reductor dimensional y un selector de características. La teoría aquí expuesta fue obtenida de [59] y [60].

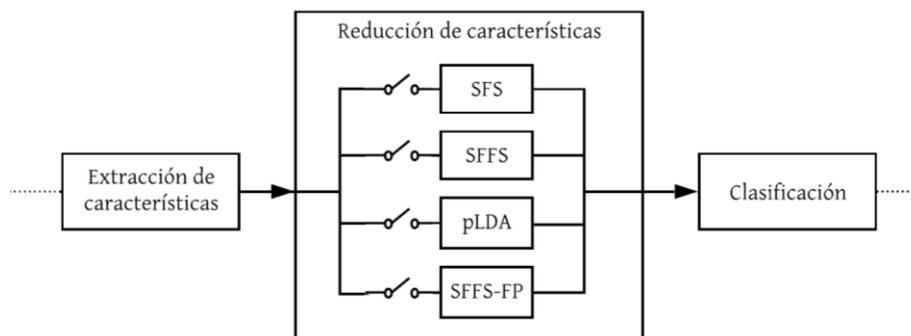


Figura 24. Etapa de reducción de características.

### 3.4.1 Selección secuencial hacia delante

La *Selección secuencial hacia adelante* (SFS, por sus siglas en inglés) es un algoritmo de selección de características y, dada su naturaleza, éste es considerado un algoritmo de aprendizaje no supervisado. SFS comienza con una matriz vacía donde serán agregadas gradualmente las características seleccionadas por una función objetivo desde la matriz de características original de dimensión  $D \times N$ ,

hasta alcanzar el nuevo espacio de características de dimensión  $K \times N$ , donde  $N$  es el número de datos del que se disponga (vectores),  $D$  la dimensión de los vectores de características originales y  $K$  la dimensión que se busca en los nuevos vectores de características.

En primer lugar, se toma como entrada al conjunto total de datos  $Y$  que contiene  $N$  vectores de dimensión  $D$ . Cada vector de  $Y$  contiene los valores de las características correspondiente a un dato obtenido del estudio.

$$Y = \begin{bmatrix} y_{00} & y_{01} & \cdots & y_{0n} \\ y_{10} & y_{11} & \cdots & y_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{d0} & y_{d1} & \cdots & y_{dn} \end{bmatrix} \quad \text{donde } \begin{matrix} d = 0, 1, 2, \dots, D - 1 \\ n = 0, 1, 2, \dots, N - 1 \end{matrix} \quad (36)$$

Sin embargo, por cuestiones de visualización, es mejor expresar los vectores como se muestra a continuación.

$$Y = \{\vec{y}_0, \vec{y}_1, \dots, \vec{y}_d\}$$

Recordando que  $\vec{y}_0, \vec{y}_1, \dots, \vec{y}_d$  son vectores que representan los renglones de la matriz mostrada en (36).

Posteriormente se busca retornar un conjunto de datos  $X_k$  compuesto por  $N$  vectores de características de dimensión  $K$ , de modo que  $K < D$ .

$$X_k = \begin{bmatrix} x_{k0} & x_{k1} & \cdots & x_{kn} \\ x_{10} & x_{11} & \cdots & x_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k0} & x_{k1} & \cdots & x_{kn} \end{bmatrix} \quad \text{donde } \begin{matrix} k = 0, 1, 2, \dots, K - 1 \\ n = 0, 1, 2, \dots, N - 1 \end{matrix} \quad (37)$$

$$X_k = \{\vec{x}_0, \vec{x}_1, \dots, \vec{x}_k\} \quad \text{donde } X_k \in Y$$

Inicialmente el conjunto de salida  $X_k$  es un conjunto vacío de vectores de dimensión  $k = 0$ .

$$X_0 = \emptyset, k = 0$$

Después, se busca un vector renglón  $\vec{y}_d$  dentro del conjunto original  $Y$  de manera que, agregándolo temporalmente al subconjunto  $X_k$ , maximice el resultado obtenido por la función objetivo  $J(\cdot)$ . En este proceso, el vector que demuestre incrementar el valor de dicha función será expresado como  $\vec{x}^+$ , tal como se muestra en la ecuación (38).

$$\vec{x}^+ = \arg \max J(X_k + \vec{y}_d), \text{ donde } \vec{y}_d \in Y \wedge \vec{y}_d \notin X_k \quad (38)$$

Cabe destacar que el *argumento máximo* ( $\arg \max$ ) indica el punto del dominio de una función que maximiza el valor de ésta, por lo que se puede interpretar que el vector  $\vec{x}^+$  es aquel vector  $\vec{y}_d$  que, al ser agregado a  $X_k$ , maximizó el valor de la función de costo.

Una vez que se establezca qué vector aporta un mejor desempeño, se incluirá definitivamente al subconjunto  $X_k$  como se muestra en la ecuación (39) y se realizará el mismo procedimiento para las características restantes.

$$X_{k+1} = X_k + \vec{x}^+ \quad (39)$$

$$k = k + 1$$

El algoritmo termina hasta que se alcance el número de características  $K$  que se especifican inicialmente, es decir,  $k = K$ .

### 3.4.2 Selección secuencial hacia delante flotante

La *Selección secuencial hacia delante flotante* (SFFS, por sus siglas en inglés) puede considerarse una extensión del algoritmo anterior. Este cuenta con un paso adicional de exclusión de características una vez que se incluyeron, de modo que se puedan muestrear un mayor número de combinaciones de subconjuntos de características. Dicho paso es condicional, pues ocurre solo cuando el subconjunto resultante es mejor una vez evaluado por la función objetivo después de eliminar la característica particular.

Al igual que en la ecuación (38) y (39), el proceso inicia incluyendo un vector renglón correspondiente a una característica del conjunto original que incremente el rendimiento del subconjunto con base en la evaluación de la función objetivo.

$$\begin{aligned}\vec{x}^+ &= \arg \max J(X_k + \vec{y}_d), \text{ donde } \vec{y}_d \in Y \wedge \vec{y}_d \notin X_k \\ X_{k+1} &= X_k + \vec{x}^+ \\ k &= k + 1\end{aligned}$$

Posteriormente se busca un vector dentro de  $X_k$  que al eliminarlo se incremente el valor de la función objetivo, tal como se observa en (40).

$$\begin{aligned}x^- &= \arg \max J(X_k - \vec{x}_k), \text{ donde } \vec{x}_k \in X_k \\ \text{Si } J(X_k - \vec{x}_k) &> J(X_k) \\ X_{k-1} &= X_k - x^- \\ k &= k - 1\end{aligned} \tag{40}$$

En caso de que  $k \leq 2$  o no exista un vector de alguna característica que al descartarlo maximice  $J(\cdot)$ , se vuelve al paso de adición de características.

La limitación de  $k \leq 2$  se debe a que, al momento de agregar el primer dato al subconjunto ( $k = 1$ ) se realiza el recorrido sobre todas las características, por lo que no es posible eliminar una de ellas de modo que mejore el rendimiento, pues el mismo algoritmo ya agotó todas las posibilidades. Cuando  $k = 2$  solo se puede eliminar el vector de la característica que se acaba de agregar o el que se agregó en la primera iteración. El primer caso no puede suceder, pues dicho vector fue escogido debido a que su adición maximiza la función objetivo, en caso de eliminarlo se caería en un ciclo infinito donde sería eliminado y agregado constantemente, pues no hay otro vector que maximice aún más la función objetivo. De igual manera, es imposible la eliminación del vector que se agregó inicialmente, pues fue escogido en la primera iteración dado que maximizaba la función objetivo aún más que el vector agregado en la segunda iteración, por lo que su eliminación no mejoraría el desempeño.

Dicho procedimiento se repite hasta que se llegue al número de características deseadas, es decir,  $k = K$ .

### 3.4.3 Discriminante lineal de Fisher

El *Análisis discriminante lineal* (LDA, por sus siglas en inglés) es un algoritmo de aprendizaje supervisado que puede ser utilizado tanto para reducción dimensional como para tareas de clasificación. Sin embargo, para que sea efectivo, es necesario asumir que los datos provienen de una distribución Gaussiana Multivariante y que las covarianzas de las  $K$  clases son iguales. Para ver más a detalle este algoritmo como clasificador, revisar la sección 3.5.1.

En LDA existe una reducción de dimensión inherente derivada de la regla bajo la que trabaja el algoritmo, la cual se describe con mayor detalle en [61]. En dicha regla se realiza una comparación de distancias sobre un espacio abarcado por las medias de las diferentes clases, el cual como máximo

es de  $C - 1$  dimensiones, donde  $C$  es el número de clases. Este puede ser aprovechado para facilitar la tarea de clasificación, pues es posible reducir la dimensión de los datos de entrada mediante una proyección ortogonal sobre aquel espacio sin perder información. Fisher propuso que una proyección de datos es óptima cuando las medias de cada clase se dispersan lo mayormente posible y se minimiza

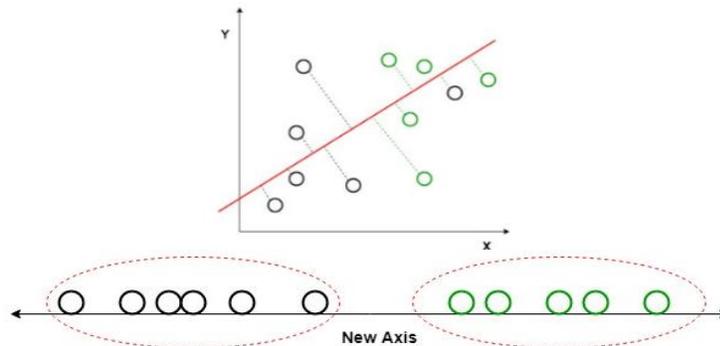


Figura 25. Reducción de espacio 2D a 1D por medio del Discriminante lineal de Fisher. Imagen adaptada de [93].

la varianza de los datos dentro de ellas, lo que da lugar a un enfoque ligeramente distinto llamado *Discriminante lineal de Fisher* (FLD, por sus siglas en inglés) o *Proyección Fisher* (FP, por sus siglas en inglés), el cual, a diferencia de LDA, no asume respecto a la distribución de los datos.

Existen lecturas que se refieren a ambos algoritmos de manera indiscriminada, pues existe una gran relación entre FLD y LDA, considerándose a este último como una generalización del primero. Debido a que explorar dicho vínculo no es el enfoque del presente trabajo, se invita al lector a encontrar información en [61] y [62], donde se aborda más a fondo el desarrollo de estos algoritmos y su conexión. Para evitar confusiones a lo largo de la presente tesis, se utilizará FLD cuando se habla de reducción dimensional y LDA cuando se habla del clasificador utilizando el teorema de Bayes.

El Discriminante lineal de Fisher es un algoritmo de aprendizaje supervisado enfocado a reducción dimensional. Este se basa en utilizar la información del conjunto de entrenamiento para obtener una matriz de transformación  $W$ , que permita reducir la dimensión de los vectores de características originales por medio de una proyección a un espacio de menor dimensión, procurando que se mantenga una mínima varianza dentro de cada clase y maximizando la distancia entre sus medias, tal como se ilustra en la Figura 25 y la Figura 26. De esta manera se logra reducir un espacio de características de dimensión  $D$  a un subespacio de dimensión  $K$  (donde  $K < D$ ) manteniendo la información discriminativa de la clase. Cabe destacar que, dada la naturaleza del FLD, es recomendable, mas no obligatorio, que la dimensión de los vectores de características originales sea mayor al número de clases, es

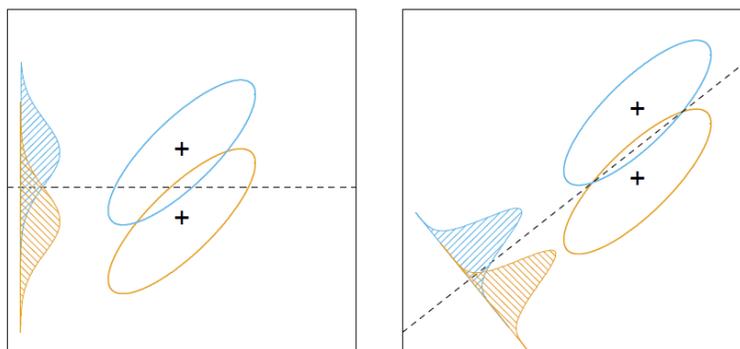


Figura 26. Diferencia entre una proyección de datos convencional (izquierda) y una realizada por el método de Fisher (derecha). Imagen obtenida de [62].

decir,  $C < D$ . La reducción dimensional deberá obedecer que  $K \leq \min(D, C) - 1$ . A continuación, se detallará paso a paso el desarrollo del algoritmo.

1. Primero es necesario recordar la estructura de los datos que entrarán al algoritmo y realizarán su entrenamiento. En este conjunto se tiene un total de  $N$  vectores  $\vec{x}_n$  de dimensión  $D$ , que en su interior contiene las características obtenidas durante su respectivo estudio. Cada vector pertenece a una clase  $c$ , tal como se muestra en la Figura 27.

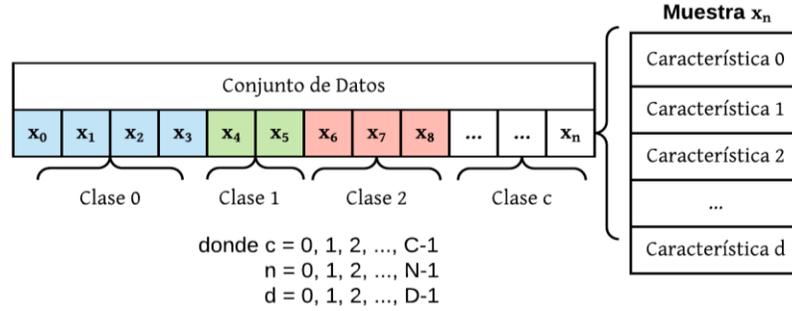


Figura 27. Estructura del conjunto de datos

2. Calcular los vectores  $\vec{m}_c$  y  $\vec{m}$  de dimensión  $D$  que contengan las medias de los vectores de características en cada clase y la media total, respectivamente. Al ser este un algoritmo de aprendizaje supervisado, se tienen los datos de la clase a la que pertenece cada uno, así que será posible obtener  $\vec{m}_c$  correspondientes a cada clase  $c$  como se indica en la ecuación (41). De igual manera  $\vec{m}$  puede ser obtenido como se expresa en la ecuación (42).

$$\vec{m}_c = \frac{1}{N_c} \sum_{n \in \Omega_c} \vec{x}_n, \quad c = 0, 1, 2, \dots, C - 1 \quad (41)$$

$$\vec{m} = \frac{1}{N} \sum_{n=0}^{N-1} \vec{x}_n = \frac{1}{N} \sum_{c=0}^{C-1} N_c \vec{m}_c \quad (42)$$

Donde  $C$  es el número de clases,  $N_c$  es el número de vectores de características en cada clase y  $N$  el número total de vectores de características.  $\Omega_c$  es un conjunto que contiene los índices de los vectores de características pertenecientes a la clase  $c$ . Por lo que  $n \in \Omega_c$  indica que sólo se usarán aquellos vectores pertenecientes a dicha clase.

3. Calcular la matriz de covarianza para cada clase  $c$  haciendo uso de su media expresada por la ecuación (41). Esta matriz tendrá dimensión  $D \times D$ . La operación  $(\cdot)^T$  indica que se pasa de un vector renglón a un vector columna y viceversa.

$$S_c = \sum_{n \in \Omega_c} (\vec{x}_n - \vec{m}_c)(\vec{x}_n - \vec{m}_c)^T \quad (43)$$

4. Calcular la matriz de covarianza dentro de las clases (de dimensión  $D \times D$ ) mediante la suma de las matrices obtenidas en la ecuación (43).

$$S_W = \sum_{c=0}^{C-1} S_c \quad (44)$$

5. Calcular la matriz de covarianza entre clases utilizando las medias calculadas en la ecuación (41) y (42). De igual manera dicha matriz tendrá una dimensión de  $D \times D$ .

$$S_B = \sum_{c=0}^{C-1} N_c (\vec{m}_c - \vec{m})(\vec{m}_c - \vec{m})^T \quad (45)$$

Como antes se mencionó, la idea del Discriminante lineal de Fisher es encontrar una matriz de transformación  $W$  que maximice la dispersión entre clases expresada en la ecuación (45) mientras se mantenga una mínima varianza dentro de cada clase, la cual es expresada mediante la matriz obtenida por la ecuación (44). Esto puede verse expresado por medio de la ecuación (46).

$$J(W) = \operatorname{argmax} \frac{|W^T S_B W|}{|W^T S_W W|} \quad (46)$$

Esta matriz de transformación puede ser obtenida mediante la descomposición en valores propios de la matriz  $A$  indicada en la ecuación (47). Cabe destacar que  $S_W^{-1} S_B$  se realiza si  $S_W$  es no singular y  $S_B^{-1} S_W$  si  $S_B$  es no singular.

$$A = S_W^{-1} S_B \quad \text{ó} \quad A = S_B^{-1} S_W \quad (47)$$

6. Calcular la matriz de vectores propios  $v$  y la matriz identidad que contenga los valores propios  $\lambda$  (ambas de dimensión  $D \times D$ ) utilizando la matriz de transformación  $A$  de la ecuación (47). Es posible obtener  $\lambda$  por medio del sistema de ecuaciones expresado en la ecuación (49) y, posteriormente, calcular la matriz  $v$  mediante la ecuación (48). La obtención de estos valores resulta muy importante, pues proveen información sobre la distorsión de una transformación lineal, donde los vectores propios son la dirección de dicha distorsión y los valores propios su magnitud.

$$Av = \lambda v$$

$$(A - \lambda I)v = \vec{0} \quad (48)$$

$$\det(A - \lambda I) = 0 \quad (49)$$

Si bien, la forma en que se calculan los vectores y valores propios es relativamente sencilla, en la práctica puede resultar computacionalmente costoso, pues implica extracción de raíces exactas en polinomios de alto grado.

La forma de estimar el número adecuado de dimensiones  $K$  al que debe ser reducido nuestro espacio de características de dimensión  $D$  es por medio de los valores propios. Únicamente debe conservarse aquellos vectores cuyos valores propios son de mayor magnitud, pues estos tienen más información sobre la distribución de los datos, a diferencia de aquellos con los valores más pequeños, los cuales son menos informativos.

7. Ordenar los vectores propios de forma decreciente basándose en su magnitud expresada por sus respectivos valores propios y, posteriormente, elegir aquellos con los  $K$  mayores. Esto finalmente resultará en una matriz  $W$  de dimensiones  $D \times K$  (donde cada columna representa los vectores propios seleccionados), que servirá para reducir la dimensión del conjunto de datos con el que será entrenado el algoritmo de clasificación, tal como se muestra en la ecuación (50). La matriz resultante  $Y$  tendrá una dimensión de  $N \times K$ , a diferencia del conjunto original  $X$  de dimensión  $N \times D$ . Del mismo modo, la matriz  $W$  será utilizada para transformar los nuevos vectores de características que se busque clasificar.

$$Y = X x W \quad (50)$$

En caso de contar con un conjunto de datos donde únicamente existan dos clases, la matriz  $W$  puede ser calculada simplemente multiplicando la inversa de la matriz  $S_W$  con la diferencia de las medias de las dos clases existentes, tal como se observa en la ecuación (51).

$$W = S_W^{-1}(\vec{m}_1 - \vec{m}_0) \quad (51)$$

Como puede observarse, FLD en realidad no aporta una *discriminante* que permita diferenciar los datos entre clases. Sin embargo, la proyección resultante puede utilizarse para generarla modelando una Distribución Gaussiana Multivariante y posteriormente usando el concepto de probabilidad condicional para realizar clasificaciones. Otro método utilizado es mediante la obtención de distancias euclidianas, tomando como referencias las características y de los centroides de cada clase, tal como se realiza en [8]. Cabe mencionar que, bajo los términos de Aprendizaje Automático, una *discriminante* es una función que toma un vector de entrada  $\vec{x}$  y la asigna a una clase  $c$  de las  $C$  existentes. Para realizar esta acción se pretende encontrar una relación lineal que discrimine las distintas clases utilizando la información de las características del conjunto de entrenamiento, de modo que, al momento de ingresar nuevos vectores de características, pueda tenerse un límite de decisión que permita clasificarlos adecuadamente.

#### 3.4.3.1 Análisis discriminante lineal pseudoinverso

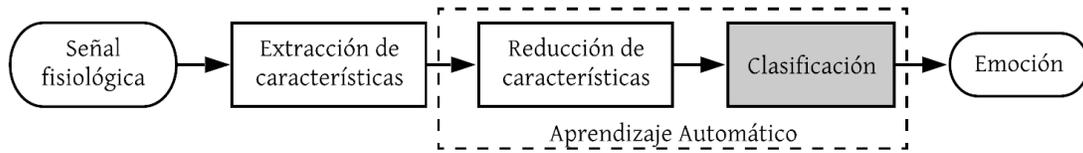
Como ya se había mencionado, hay una estrecha relación entre FLD y LDA, por lo que hay ocasiones en que se nombran de la misma manera. Existe una variación de FLD cuyo nombre es *Análisis discriminante lineal pseudoinverso* (pLDA, por sus siglas en inglés), el cual da solución a algunas limitantes del algoritmo original.

En el Discriminante lineal de Fisher es necesario que una de las matrices de covarianza ( $S_W$  o  $S_B$ ) sea no singular (invertible). De este problema surge pLDA, el cual se realiza obteniendo los vectores y valores propios de la matriz  $S_W^+ S_B$  o  $S_B^+ S_W$ , donde  $S_W^+$  y  $S_B^+$  son matrices pseudoinversas, en lugar de la expresada en la ecuación (47). La matriz pseudoinversa es una generalización de la matriz inversa y existe para toda matriz de dimensión  $m \times n$ . La forma más simple de obtenerla es por medio de la descomposición en valores singulares (SVD, por sus siglas en inglés), la cual queda expresada para cualquier matriz  $A$  como  $A = U \Sigma V^T$ , donde  $\Sigma$  es una matriz diagonal cuyos elementos no nulos son los valores singulares de la matriz  $A$ , mientras que  $U$  y  $V$  se conocen como vectores singulares por la izquierda y por la derecha, respectivamente. En el caso de su matriz pseudoinversa, la descomposición en valores singulares será  $A^+ = U \Sigma^+ V^T$ , donde la matriz diagonal  $\Sigma^+$  se obtiene tomando el recíproco de cada elemento distinto de cero de la diagonal principal de  $\Sigma$ .

#### 3.4.4 Algoritmos híbridos

Es posible utilizar algoritmos de reducción dimensional en conjunto con algoritmos de selección, aunque esto depende de la cantidad de características con que se cuente, pues si hay un número reducido probablemente no sea necesaria la implementación en conjunto o individual de los algoritmos. El uso de ambos sistemas es beneficioso al contar con un gran número de características, pues como resultado se contará con la mejor versión de los vectores. Una de las combinaciones más populares dentro del Estado del Arte es la Selección secuencial hacia delante (también en su variante flotante) y la Proyección Fisher. Dicho híbrido resultaría en una Proyección Fisher alimentada de los vectores de características resultantes del algoritmo SFFS o SFS, es decir, SFFS-FP o SFS-FP (de igual forma puede escribirse como SFFS-FLD o SFS-FLD).

### 3.5 Algoritmo de clasificación



La última etapa de un sistema de detección de emociones es el algoritmo de clasificación, el cual provee de una herramienta que, una vez extraídas las características provenientes de las señales fisiológicas, permitirá estimar la emoción que se está experimentando en ese instante. Dicha clasificación se puede realizar por medio de una gran cantidad de métodos, tal como se describieron en la sección 3.3.1.

La mayoría de los clasificadores consta de dos etapas: el entrenamiento y la puesta en funcionamiento o prueba. En la primera etapa será necesario aportar ejemplos de los datos a clasificar, es decir, de vectores de características ligados con la información sobre el estado emocional en el que se encontraba el individuo al extraerlas, el cual representará su respectiva *clase*. Para recordar la estructura de dichos datos, puede visualizarse la Figura 27, ya que ésta se utilizó para la explicación de un método de aprendizaje supervisado al igual que lo son los algoritmos enlistados en esta sección. Una vez recolectada esta información dentro de un conjunto de entrenamiento tomando en cuenta la metodología descrita en la sección 3.3.5, se podrá realizar el ajuste de los parámetros propios del algoritmo con el objetivo de predecir a qué clase pertenecen los nuevos vectores de características que se ingresen en la segunda etapa.

En el Estado del Arte fue utilizada una gran cantidad de algoritmos de clasificación, los cuales variaron su desempeño dependiendo del tipo de clasificador que se buscaba, tal como puede observarse en la Tabla 3. En la presente tesis se opta por abordar el algoritmo LDA, el cual es uno de los clasificadores con mayor desempeño en el ámbito dependiente, evaluando así su practicidad dentro del sistema detector de emociones planteado. Los motivos de dicha elección son abordados con mayor detalle en la sección 4.2.4.

En la Figura 28 se muestra, de forma simplificada, la posible integración al sistema de los algoritmos de clasificación más relevantes en el ámbito dependiente (de acuerdo con la Tabla 3), siendo kNN, un clasificador basado en la obtención de distancias; LDA, un clasificador que utiliza el teorema de Bayes para calcular la probabilidad de pertenencia a las clases y MLP, una red neuronal. Sin embargo, como se describió anteriormente, el presente trabajo únicamente centrará su análisis sobre el algoritmo LDA dado que éste fue elegido como algoritmo de clasificación del sistema detector de emociones planteado. La teoría aquí expuesta está basada en [63].

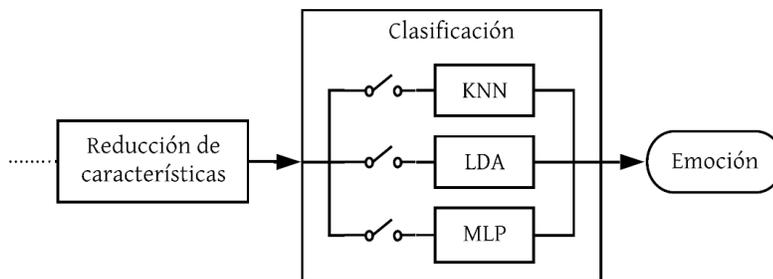


Figura 28. Etapa de clasificación

#### 3.5.1 Análisis discriminante lineal

Como se mencionó en la sección 3.4.3, el *Análisis discriminante lineal* (LDA, por sus siglas en inglés), es un algoritmo de aprendizaje supervisado utilizado para reducción dimensional y clasificación [61]. LDA como clasificador busca generar una función discriminante que permita alojar un

vector de características  $x$  dentro de una clase  $k$ . Uno de los métodos para crearla es utilizando la probabilidad de que el vector de características de entrada pertenezca a la clase  $k$  en cuestión. Utilizando la notación correcta, dicha probabilidad queda definida como  $P(Y = k|X = x)$ , también conocida como probabilidad *a posteriori*, que indica la probabilidad de que la variable aleatoria  $Y$  adquiera el valor de la clase  $k$  de las  $K$  existentes dada una variable aleatoria  $X$  (también llamada predictor) que adquiera el valor del vector de características  $\vec{x}$  (también llamado observación). Finalmente, el vector de características será asignado a la clase con la mayor probabilidad. En una implementación real, la mayoría de las veces se cuenta con vectores de características de dimensión  $D$ , por lo que es necesario generalizar el modelo a uno con múltiples predictores, es decir,  $X = X_0, X_1, \dots, X_d$  donde  $d = 0, 1, \dots, D - 1$ .

Para el entrenamiento del LDA es necesario hacer algunas consideraciones: se supone que las matrices de covarianza son iguales para todas las clases y que los vectores de características cuentan una *distribución Gaussiana Multivariante*, esto sólo se puede lograr si cada uno de los elementos individuales que lo conforman y toda combinación lineal entre ellos cuentan con una distribución Gaussiana.

El teorema de Bayes estipula que, dado un evento cualquiera  $X$  y un conjunto de sucesos mutuamente excluyentes y exhaustivos  $\{Y_0, Y_1, \dots, Y_n\}$ , la probabilidad de que un suceso  $Y_i$  suceda, dada la ocurrencia del evento  $X$ , es igual a la probabilidad de que ambos eventos ocurran al mismo tiempo dividido entre la probabilidad de que ocurra  $X$ , es decir:

$$P(Y_i|X) = \frac{P(X \cap Y_i)}{P(X)} = \frac{P(Y_i)P(X|Y_i)}{P(X)}$$

De dicha afirmación, y aplicando el Teorema de la probabilidad total sobre  $P(X)$ , es posible llegar a la ecuación (52), la cual es fundamental para el desarrollo de LDA.

$$P(Y_i|X) = \frac{P(Y_i)P(X|Y_i)}{\sum_{k=1}^n P(Y_k)P(X|Y_k)} \quad (52)$$

Donde  $P(Y_i)$  es la probabilidad *a priori*,  $P(X|Y_i)$  es la probabilidad de  $X$  dada la ocurrencia de  $Y_i$  y  $P(Y_i|X)$  es la probabilidad *a posteriori*.

El entrenamiento del algoritmo consta de determinar una serie de parámetros utilizando los vectores de características del conjunto de entrenamiento. La obtención de estos se describe a continuación.

1. Calcular la probabilidad *a priori*  $\hat{\pi}_k \equiv P(Y = k)$ , que indica la probabilidad de que un vector de características  $\vec{x}$  pertenezca a una clase  $k$ . Dicho parámetro suele ser sencillo de determinar, pues la probabilidad de que un vector  $\vec{x}$  cualquiera pertenezca a una clase  $k$  es igual al número de vectores pertenecientes a dicha clase dividido entre el total de vectores dentro del conjunto de entrenamiento, tal como se observa en la ecuación (53).

$$\hat{\pi}_k \equiv P(Y = k) = \frac{n_k}{N} \quad \text{donde } k = 0, 1, \dots, K - 1 \quad (53)$$

Donde  $n_k$  es el número de vectores en la clase  $k$ ,  $N$  el número total de vectores dentro del conjunto de entrenamiento y  $K$  el número total de clases.

2. Calcular el vector que corresponde a la media  $\vec{m}_k$  de los vectores de características de entrenamiento de cada clase  $k$ .

$$\vec{m}_k = \frac{1}{n_k} \sum_{i \in \Omega_k} \vec{x}_i \quad (54)$$

Donde  $\Omega_k$  es un conjunto que contiene los índices de los vectores de características pertenecientes a la clase  $k$ . Por lo que  $i \in \Omega_k$  indica que sólo se usarán aquellos vectores pertenecientes a dicha clase.

3. Calcular la matriz de covarianza  $S$  de dimensión de  $D \times D$  tal como se muestra en la ecuación (55). Cabe destacar que esta matriz es la media ponderada de las matrices de covarianza muestrales de las  $K$  clases, por lo que no es necesario que se cumpla la condición que dicta que todas las matrices de covarianza deben ser iguales para que funcione el algoritmo, sin embargo, se tendrá un mejor desempeño si lo son, es decir, que  $S_0 = S_1 = \dots = S_k = S$ .

$$S = \frac{1}{N - K} \sum_{k=0}^{K-1} \sum_{i \in \Omega_k} (\vec{x}_i - \vec{m}_k)(\vec{x}_i - \vec{m}_k)^T \quad (55)$$

Es necesario aclarar que los valores reales de  $\hat{\pi}_k$ ,  $\vec{m}_k$  y  $S$  se desconocen, por lo que las expresiones calculadas son únicamente estimaciones a partir de los datos del conjunto de entrenamiento. Mientras mayor sea el número de vectores de características dentro de éste, más aproximadas serán las estimaciones.

4. Definir una función de densidad  $f_k(\vec{x}) \equiv P(X = \vec{x}|Y = k)$  que indicará la probabilidad condicional de  $\vec{x}$  dado que  $Y$  adquiriera el valor de la clase  $k$ . Mientras mayor sea el valor de  $f_k(\vec{x})$ , mayor será la probabilidad de que el vector de la clase  $k$  adquiriera un valor de  $X \approx \vec{x}$ . Como se mencionó anteriormente, LDA asume que las características cuentan con una distribución Gaussiana Multivariante, por lo que es posible modelar la función de densidad tomando en cuenta dicha distribución, tal como se observa en la ecuación (56).

$$f_k(\vec{x}) \equiv P(X = \vec{x}|Y = k) = \frac{1}{(2\pi)^{\frac{D}{2}} |S|^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{x} - \vec{m}_k)^T S^{-1} (\vec{x} - \vec{m}_k)} \quad (56)$$

5. Calcular la probabilidad *a posteriori*  $P(Y = k|X = \vec{x})$  que indicará la probabilidad de que una observación  $x$  pertenezca a la clase  $k$ , tal como se muestra en la ecuación (57). Para ello se recurre a la Regla de Bayes expresada en la ecuación (52), donde se sustituyen las ecuaciones (53) y (56).

$$P(Y = k|X = \vec{x}) = \frac{P(Y = k)P(X = \vec{x}|Y = k)}{\sum_{j=0}^{K-1} P(Y = j)P(X = \vec{x}|Y = j)} = \frac{\hat{\pi}_k f_k(\vec{x})}{\sum_{j=0}^{K-1} \hat{\pi}_j f_j(\vec{x})} \quad (57)$$

$$P(Y = k|X = \vec{x}) = \frac{\hat{\pi}_k \frac{1}{(2\pi)^{\frac{D}{2}} |S|^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{x} - \vec{m}_k)^T S^{-1} (\vec{x} - \vec{m}_k)}}{\sum_{j=0}^{K-1} \hat{\pi}_j \frac{1}{(2\pi)^{\frac{D}{2}} |S|^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{x} - \vec{m}_j)^T S^{-1} (\vec{x} - \vec{m}_j)}} \quad (58)$$

Es posible simplificar la ecuación (58) por medio de una transformación logarítmica, pues se evitan los exponentes y se convierten las multiplicaciones en una serie de sumas. Además, gran parte de los términos como es el caso del denominador y algunos otros de la función de densidad pueden ser agrupados en una constante, tal como se visualiza en la ecuación (59). Ésta, al estar presente de manera uniforme en las probabilidades de las  $K$  clases, será posible ignorarla, pues incrementa su valor por igual y no aporta información para discriminarlas.

$$\hat{\delta}_k(\vec{x}) \equiv \ln(P(Y = k|X = \vec{x})) = \ln(P(X = \vec{x}|Y = k)) + \ln(P(Y = k)) + \text{cte}$$

$$\hat{\delta}_k(\vec{x}) = -\frac{1}{2} \ln |S| - \frac{1}{2} (\vec{x} - \vec{m}_k)^T S^{-1} (\vec{x} - \vec{m}_k) + \ln(\hat{\pi}_k) + \text{cte} \quad (59)$$

De igual forma, dado que  $-\frac{1}{2} \ln |S|$  cuenta con el mismo valor en todos los cálculos de probabilidad de las  $K$  clases, es posible agruparlo con el resto de las constantes, tal como se expresa en la ecuación (60).

$$\hat{\delta}_k(\vec{x}) = -\frac{1}{2} (\vec{x} - \vec{m}_k)^T S^{-1} (\vec{x} - \vec{m}_k) + \ln(\hat{\pi}_k) + \text{cte} \quad (60)$$

Es preciso señalar que el término  $(\vec{x} - \vec{m}_k)^T S^{-1} (\vec{x} - \vec{m}_k)$  corresponde a la distancia de Mahalanobis entre el vector de características  $\vec{x}$  y la media  $\vec{m}_k$ . Esta distancia nos dice qué tan cercanos son dichos vectores tomando en cuenta la varianza de cada uno de sus elementos. De esta forma, puede interpretarse que LDA realiza una asignación de  $\vec{x}$  a la clase cuya media está más cercana en términos de esta distancia, considerando, de igual manera, la probabilidad *a priori* para cada clase.

La ecuación (60) puede ser desarrollada como se muestra a continuación, llegando finalmente a la función discriminante expresada en la ecuación (61).

$$\begin{aligned} \hat{\delta}_k(\vec{x}) &= -\frac{1}{2} (\vec{x} - \vec{m}_k)^T S^{-1} (\vec{x} - \vec{m}_k) + \ln(\hat{\pi}_k) + \text{cte} \\ \hat{\delta}_k(\vec{x}) &= -\frac{1}{2} (\vec{x}^T S^{-1} - \vec{m}_k^T S^{-1}) (\vec{x} - \vec{m}_k) + \ln(\hat{\pi}_k) + \text{cte} \\ \hat{\delta}_k(\vec{x}) &= -\frac{1}{2} (\vec{x}^T S^{-1} \vec{x} - \vec{m}_k^T S^{-1} \vec{x} - \vec{x}^T S^{-1} \vec{m}_k + \vec{m}_k^T S^{-1} \vec{m}_k) + \ln(\hat{\pi}_k) + \text{cte} \\ \hat{\delta}_k(\vec{x}) &= \vec{x}^T S^{-1} \vec{m}_k - \frac{1}{2} \vec{m}_k^T S^{-1} \vec{m}_k + \ln(\hat{\pi}_k) + \text{cte} \end{aligned} \quad (61)$$

Cabe resaltar que el término  $-\frac{1}{2} \vec{x}^T S^{-1} \vec{x}$  es una constante que puede ser agrupada de la misma forma en que se ha realizado con otros elementos anteriormente.

Una vez que se cuente con la función  $\hat{\delta}_k(\vec{x})$  para las  $K$  clases se tendrá toda la información para realizar una clasificación, pues únicamente será necesario calcular, para todas las clases, la probabilidad de que un nuevo vector de características  $\vec{x}$  pertenezca a dicha clase, siendo asignado a aquella con la mayor probabilidad. Es preciso señalar que, como puede observarse en la ecuación (61), la función discriminante  $\hat{\delta}_k(\vec{x})$  es lineal respecto de  $\vec{x}$ , por lo que el algoritmo toma el nombre de Análisis Discriminante Lineal, sin embargo, el desarrollo únicamente se hizo para ejemplificar dicha cuestión, pues se considera que la ecuación (60) es más adecuada para la implementación debido a que cuenta con un menor número de operaciones.

Cuando no se cumple la distribución Gaussiana Multivariante antes descrita, es posible que el algoritmo pierda precisión, sin embargo, sus predicciones seguirán siendo relativamente buenas. Además, si la matriz de covarianza no es igual en todas las clases, es posible que se mejoren los resultados recurriendo al *Análisis discriminante cuadrático* (QDA, por sus siglas en inglés), pues éste considera matrices de covarianza distintas en cada clase, resultando en una expresión similar a la mostrada en (59), sustituyendo la matriz  $S$  por  $S_k$ . En el caso del LDA, los límites de decisión son lineales, por lo que, si la separación de las clases no lo es, puede que el modelo sufra un subajuste. En QDA se producen límites curvos, aportando una mayor flexibilidad y ajuste a los datos, sin embargo, dichas cualidades también aumentan la probabilidad de un sobreajuste a los datos.

### 3.6 Esquema EMDC

Con anterioridad se han descrito aquellos algoritmos de clasificación y reducción de características con mejor desempeño en la detección de emociones de acuerdo con el Estado del Arte y el criterio establecido en la sección 3.5. Sin embargo, en [8] se menciona un esquema enfocado a la clasificación de emociones denominado *Clasificación dicotómica multinivel específica a emociones* (EMDC, por sus siglas en inglés) con el cual se obtuvieron resultados bastante satisfactorios. Es por este motivo que, a pesar de no ser un algoritmo como tal, se considera importante abordarlo, pues es un método de ensamble de modelos de clasificación que puede ser beneficioso para el objetivo de la presente tesis.

Al utilizar un algoritmo para clasificar múltiples clases se eleva su complejidad pues, a diferencia de una clasificación binaria, los límites de decisión serán más difíciles de determinar. Otra forma de resolver un problema de clasificación multiclase es utilizando una serie clasificadores binarios, pues es posible descomponer múltiples clases en pares de clases complementarias, como es el caso del método *uno contra todos*, en el cual se realiza una serie de clasificaciones binarias de cada clase contra el resto, una vez que se ingrese un dato, pasará a través de todos los clasificadores y se elegirá aquella con una mayor pertenencia. Un ejemplo de este método puede ser una clasificación de tres clases ( $C_1, C_2, C_3$ ), donde al ingresar el vector de características se realizarán tres clasificaciones binarias:  $C_1$  vs  $C_2 \cup C_3$ ,  $C_2$  vs  $C_1 \cup C_3$  y  $C_3$  vs  $C_1 \cup C_2$ .

EMDC explota la propiedad de categorización dicótoma utilizando el modelo dimensional de Valencia – Excitación, el cual es mayormente explorado en la sección 3.7.2. Este esquema toma en consideración que, de acuerdo con las investigaciones del estudio, la clasificación por medio de la excitación arroja mejores resultados que a través de la valencia o por un modelo discreto. EMDC consiste en dividir los datos de entrenamiento dentro de dos “superclases” opuestas usando uno de los dos ejes del modelo dimensional (valencia o excitación). Posteriormente, a través de múltiples clasificadores binarios ordenados en distintos niveles, se dividirán los datos en dos subconjuntos y se entrenará al algoritmo de clasificación en cada nivel hasta llegar al número de clases deseadas, dicho de otra forma, se iniciará con un grupo completo que será dividido en dos clases, donde dichos subconjuntos volverán a ser divididos en dos, una y otra vez, formando una especie de árbol de clasificación binario. Finalmente, una vez entrenados los clasificadores, los nuevos vectores de características pasarán a través de dicho árbol, con el cual se realizarán una serie de clasificaciones binarias hasta determinar su clase. Cabe destacar que es necesario que el número de clases sea potencia de dos. De igual manera, dicho valor definirá el número de niveles del esquema.

En la Figura 29 puede observarse una clasificación de tres niveles en ocho clases. Inicialmente se divide el conjunto de entrenamiento en dos superclases a través de una clasificación binaria, tomando como referencia el eje de excitación, donde  $h$  indicará aquellos datos con una alta excitación y  $l$  aquellos con baja excitación (nivel 1). Posteriormente, cada grupo será clasificado de acuerdo con un

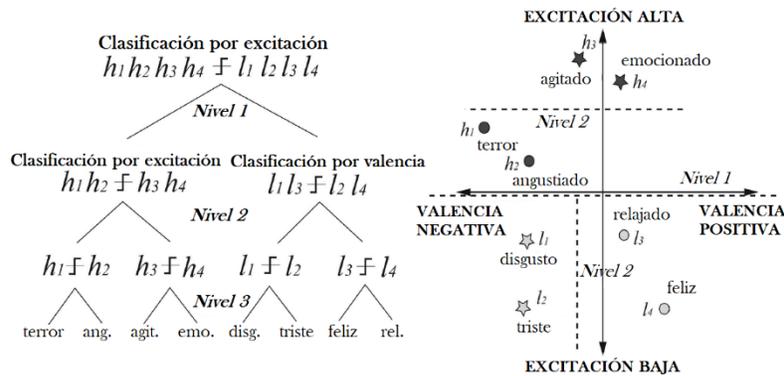
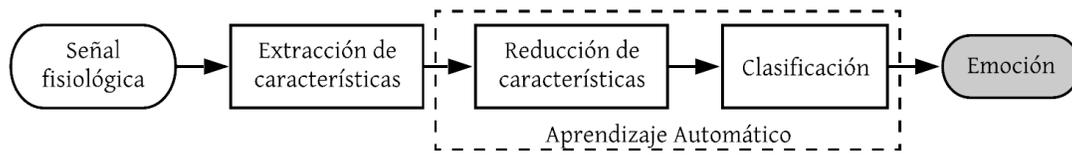


Figura 29. Esquema EMDC. Imagen adaptada de [8].

nuevo nivel de valencia o excitación (nivel 2), en el caso de los datos con etiqueta  $h$ , serán clasificados binariamente dependiendo de qué tan alta es su excitación  $y$ , por otro lado, los datos de etiqueta  $l$  serán clasificados de acuerdo con que tan negativa o positiva resulta su valencia. En el último nivel, y con cuatro subconjuntos de datos, únicamente se tendrá que realizar una simple distinción binaria entre dos clases, por lo que los clasificadores binarios podrán ser entrenados con los datos agrupados en ocho subconjuntos que corresponderán a las ocho clases que se buscaban (nivel 3).

Como antes se mencionó, este método no es un algoritmo de clasificación, sino un esquema que aprovecha la capacidad de realizar clasificaciones binarias y la estructura del modelo Valencia - Excitación para encontrar una forma óptima de predecir emociones. Este método necesitará de un algoritmo para realizar las clasificaciones binarias anteriormente descritas, por lo que podría utilizarse LDA, el cual ya fue abordado en la sección anterior.

### 3.7 Emoción



Finalmente, nos encontramos con el último bloque de un sistema detector de emociones cuyo fin es brindar el esquema bajo el cual se analizarán las emociones a clasificar. A continuación, se analizan las perspectivas con las que los investigadores han estudiado a las emociones, los modelos que se han generado a partir de ellas, sus implicaciones sobre los clasificadores vistos en la sección 3.5, los estimuladores con los cuales se evoca una emoción y el medio con el cual se registra.

#### 3.7.1 Perspectivas de la emoción

En primera instancia, resulta importante conocer las perspectivas bajo las cuales el ser humano da interpretación a las emociones considerando las diferentes circunstancias o elementos que influyen en su evocación. Estos resultan un parteaguas hacia ciertas limitantes o restricciones que deberán considerarse para el diseño del detector de emociones. A pesar de existir múltiples teorías respecto a las mismas, se mencionarán las perspectivas descritas en [64], dada la síntesis que realiza el autor con base en múltiples trabajos que ahondan en este tema:

- 1) **Emociones como expresiones:** Esta teoría parte de las exploraciones de Darwin [65] al observar que ciertas expresiones faciales y corporales humanas resultaban semejantes a las de los animales, por lo que concluye que las correlaciones conductuales de experiencias emocionales son el resultado de procesos evolutivos. Con esto Darwin declara que la evolución de las emociones se debe principalmente a su relación con otras acciones esenciales, por ejemplo, una cara de disgusto frente al rechazo de un objeto ofensivo para el consumo.

Mediante esta postura, el trabajo de varios investigadores gira en torno a la existencia de seis expresiones básicas reconocibles a lo largo de todas las culturas del mundo, siendo estas: enojo, miedo, sorpresa, disgusto, felicidad y tristeza.

- 2) **Emociones como personificaciones:** Basado en la teoría de James-Lange [2], que dicta que la evocación de una emoción prosigue a las respuestas fisiológicas que experimenta el Sistema Nervioso Simpático. De esta manera, identifica a la emoción a través de los cambios fisiológicos que se suscitan en el cuerpo, en vez de considerarlas expresiones de ésta. Dicha postura dicta la relación que posee la psicología de las emociones con las respuestas fisiológicas más allá de las expresiones faciales o corporales de la postura descrita en el inciso anterior.
- 3) **Acercamiento cognitivo a las emociones:** Considerando a Arnold [66] como su pionero, describe la influencia que tiene un determinado evento u objeto sobre la capacidad del

individuo para percibir una emoción, dada la relación que tiene éste con las experiencias de la persona, sus metas y oportunidades de reacción.

- 4) **Emociones como construcciones sociales:** El sociólogo J. R. Averill [67] declara la necesidad de realizar un análisis social para comprender enteramente la naturaleza de las emociones. Y por su parte, Salovey [68] describe procesos como la adaptación (ajustes en respuesta al entorno), la coordinación social (reacciones en respuesta a expresiones de otros) y la auto regulación (reacciones basadas en nuestro auto entendimiento emocional y nuestra relación con el entorno), que influyen directamente sobre los estados emocionales de las personas. Asimismo, en esta perspectiva se aborda la dificultad de expresar determinadas emociones en algunos idiomas dada la falta de vocabulario en el mismo.

A pesar de que algunos investigadores de la Computación Afectiva no terminan por adoptar todas las perspectivas mencionadas, la mayoría opta por utilizar a la perspectiva de las *emociones como personificaciones* dados los resultados satisfactorios que ha entregado en la práctica.

### 3.7.2 Modelos de la emoción

Una vez adquiridas las señales fisiológicas respectivas a una emoción y planteadas las perspectivas bajo las cuales se identificará, falta establecer el medio por el cual es posible hacer una distinción entre la gran cantidad de emociones que existen dentro de la psicología del ser humano, esto se logra mediante un modelo de la emoción. Estos modelos definen cuáles son las emociones que experimenta el ser humano y con base en ellos se realiza el entrenamiento del clasificador a implementar. Los modelos normalmente utilizados en el área de la computación afectiva son los siguientes:

- 1) **Modelo discreto:** define a las emociones mediante categorías discretas con las cuales es posible distinguirlas, así como conformar otras más complejas a través de la composición de algunas de ellas. De forma que este modelo termina por establecer un sistema de una sola etapa cuyo conjunto de características define a una emoción.

Dentro de los primeros modelos se encuentra el trabajo propuesto por el psicólogo Paul Ekman basado en seis emociones fundamentales: miedo, enojo, disgusto, tristeza, felicidad y sorpresa. Más adelante, surgieron otras propuestas como el diagrama de árbol de emociones de Parrott y la rueda de la emoción del psicólogo Robert Plutchik (Figura 30) en la cual se definen ocho emociones básicas: alegría, tristeza, anticipación, sorpresa, ira, miedo, desagrado y confianza,

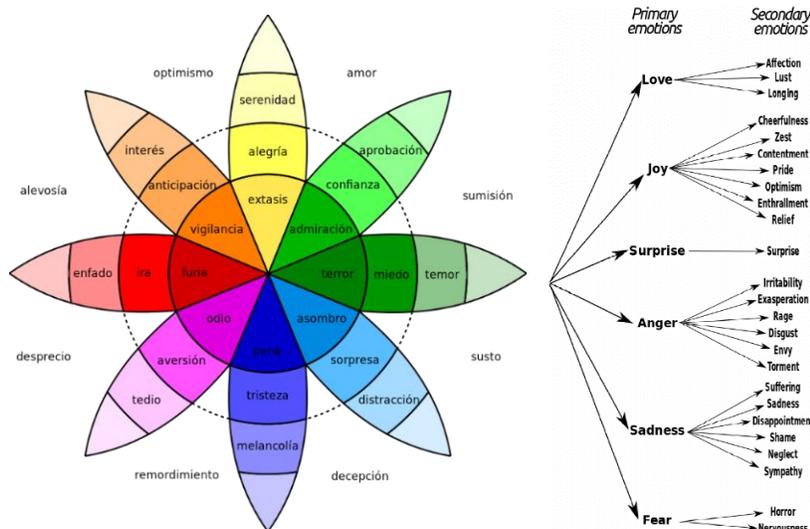


Figura 30. Modelos discretos: Rueda de la emoción de Plutchik (Izquierda) y primeras dos capas del diagrama de árbol de Parrott (Derecha). Imágenes adquiridas de [101] y [100].

de las cuales se presentan diversos grados de intensidad y con las que se conforman otras ocho emociones avanzadas, producto de la combinación de dos emociones básicas: optimismo, amor, sumisión, susto, decepción, remordimiento, desprecio y alevosía.

Estos modelos, a pesar de resultar los más cercanos a la clasificación natural que emplean las personas para identificar sus emociones, carecen de la capacidad de distinguir el intervalo de emociones que se experimentan durante un ambiente natural de comunicación. Así como carecer de las distintas maneras en que las personas denominan a una emoción, siendo en ocasiones imposible atribuirla a una palabra en concreto, o bien la cultura no cuenta con alguna para describirla.

- 2) **Modelo dimensional:** parte de una estructura de varias escalas (dimensiones), de naturaleza continua e independientes entre sí, de las cuales divergen todas las emociones. Este modelo puede entenderse como un sistema de dos etapas, en cuya primera etapa se adquiere el grado continuo de cada una de las escalas y en la segunda etapa se interpreta el grado anterior para la determinación de una emoción en concreto.

De estos modelos, el más utilizado dentro de la Computación Afectiva ha sido aquel del círculo circunflejo propuesto por el psicólogo James Russell cuya representación puede darse de acuerdo con un diagrama bidimensional conformado por un eje de *valencia*, asociado al grado de placer con la que se experimenta una determinada emoción, y otro eje de *excitación*, correspondiente a su nivel de intensidad (Figura 31). Asimismo, mediante la aportación de Merhabian se ha añadido un tercer eje denominado de *dominancia* que va desde un nivel débil hasta uno de empoderamiento para denotar el grado de control que tiene el individuo sobre una determinada emoción, permitiendo distinguir emociones como miedo y enojo.

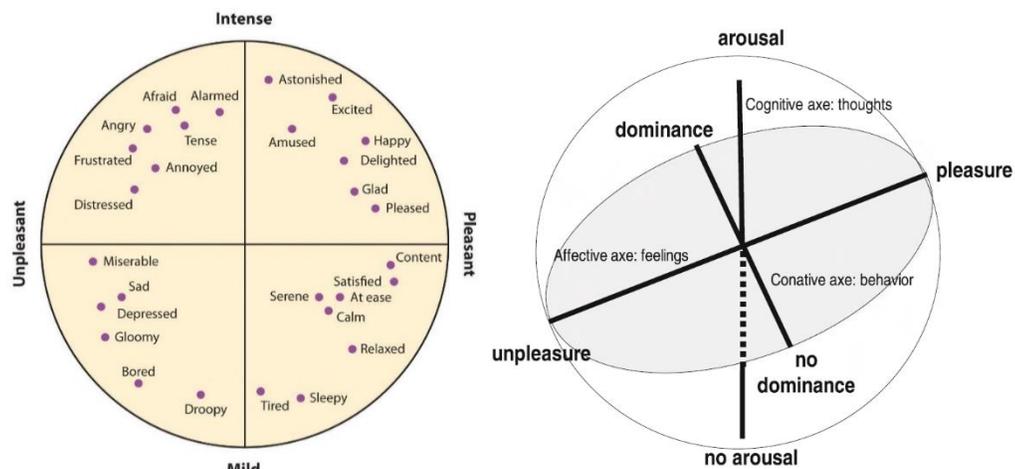


Figura 31. Modelos dimensionales: Círculo circunflejo de Russell (Izquierda) y modelo Russell-Merhabian (Derecha). Imágenes adquiridas de [22].

### 3.7.3 Implicaciones de la emoción sobre los clasificadores

Asimismo, es importante mencionar que dentro de este esquema emocional se verán reflejadas dos implicaciones cruciales para el entrenamiento y prueba de cualquiera de los clasificadores descritos en la sección 3.5, comenzando por las dos posturas que distinguen al número de individuos que entrenarán al clasificador mencionado, dividiéndose en:

- 1) **Clasificador dependiente:** describe la necesidad de emplear exclusivamente los resultados de un solo individuo, pues se considera que las experiencias que éste ha vivido influyen en la forma e intensidad de la emoción evocada ante determinadas situaciones.

Este esquema resulta ventajoso para aquellos individuos que sufren de algún trastorno fisiológico que afecte la intensidad con que se experimentan las emociones, ya que la etapa de entrenamiento del clasificador puede verse como un proceso adaptativo que evoluciona conforme a la condición del individuo. De forma que este acercamiento permite garantizar, en cierta medida, el *acercamiento cognitivo de las emociones* mencionado en los párrafos anteriores.

- 2) **Clasificador independiente:** parte de la necesidad de utilizar un conjunto de señales de múltiples individuos, pues se considera una universalidad de las emociones, es decir, parte de una postura en la cual todos los seres humanos son capaces de percibir las de la misma manera.

Por otra parte, mediante el trabajo de Villon y Lisetti en [69], se presenta una distinción respecto al tipo de evaluación que debe realizarse sobre el resultado que entregue un determinado estimulador, es decir, si la persona influirá en la decisión que se tome respecto a la emoción que éste evoca, dividiéndose en dos posturas:

- 1) **Evaluación subjetiva del estímulo:** Esta postura requiere de la evaluación independiente del sujeto de prueba sobre la emoción que le evoca un determinado estímulo. Por ello, se considera nuevamente al *acercamiento cognitivo de las emociones* de los párrafos anteriores.
- 2) **Acuerdo social del estímulo:** Esta postura emplea bases de datos de varios estímulos cuyo consenso respecto a la emoción que evoca fue recolectada de un gran número de individuos, por lo que pretenden poseer un alto grado de veracidad. De esta forma, se considera nuevamente una universalidad de las emociones.

### 3.7.4 Estimuladores y evaluaciones de la emoción

Los instrumentos que permiten evocar las emociones sobre un individuo reciben el nombre de *estimuladores*, de estos se observó mediante la Tabla 2 que los más utilizados por los estudios son los vídeos, sin embargo, aquellos más efectivos fueron los objetos o canciones cuya procedencia es por parte del individuo a evaluar, considerándose como objetos a elementos como fotografías, videos, etc. Asimismo, podría observarse que el uso de estos objetos se asocia al *acercamiento cognitivo de las emociones*, pues mediante ellos se pretenden que el individuo experimente la emoción dadas las experiencias personales que asocia a ellos.

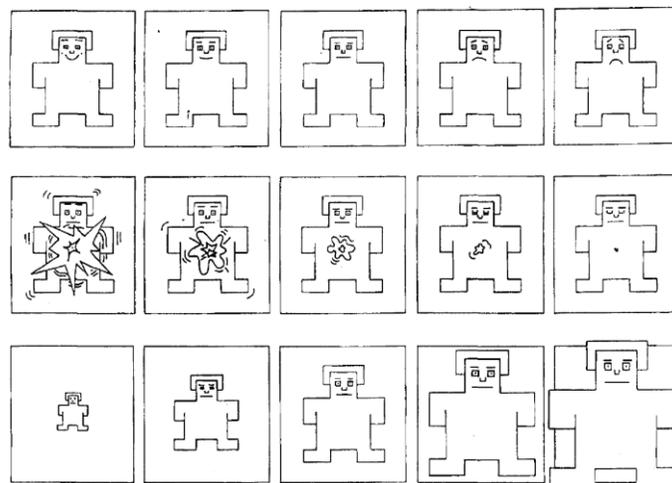


Figura 32. Maniquí de Auto Valoración para la medición de los niveles de valencia (panel superior), excitación (panel intermedio) y dominancia (panel inferior). Imagen adquirida de [70].

Por otro lado, para evitar el conflicto que presenta la perspectiva de las *emociones como construcciones sociales* respecto a la dificultad de expresar las emociones en un idioma donde no exista el vocabulario adecuado para ello, o incluso poblaciones que no sean lo suficientemente avanzadas en

el sentido lingüístico, por ejemplo niños, resulta conveniente el uso del Maniquí de Auto Valoración (SAM, por sus siglas en inglés) [70]. Este comprende un método fácil, no verbal y barato para el reporte de las experiencias emocionales de las personas, que además permite la descripción de los tres ejes *valencia-excitación-dominancia* que conforman al modelo dimensional Russel-Merhabian. Yace conformado por tres paneles que a la vez se subdividen en una serie de imágenes como se ilustra en la Figura 32, que bien parten desde un SAM con una cara sonriente a una cara triste para denotar el grado de valencia, desde unos ojos muy abiertos hasta unos cerrados para representar el grado de excitación, y desde un tamaño pequeño hasta uno grande para indicar el grado de dominancia. Este instrumento se emplea a modo que el entrevistado pueda seleccionar alguna de las cinco imágenes de cada uno de los paneles, o bien la línea intermedia entre ellas, para así conformar una escala de nueve puntos para cada uno de los ejes *valencia-excitación-dominancia*.

## 4 Diseño del proyecto

Analizados los trabajos en el área y estudiados los fundamentos clave de un detector de emociones, continuaremos por describir el proceso que sigue el diseño del proyecto de la presente tesis. Este proceso, inspirado en el libro [71], se conforma por las etapas ilustradas en la Figura 33 en las cuales se comienza por declarar la misión y visión del proyecto, seguido de un análisis donde se determina la idea a desarrollar, referido como “concepto” en los párrafos subsecuentes, con base en las necesidades del mercado y especificaciones que brinden al mismo de propiedades que lo diferencien del resto de trabajos en el área. Posteriormente, en el diseño a nivel sistema se concreta el concepto previo mediante el bosquejo de subsistemas que denoten sus funcionalidades. En el diseño de detalle se aclaran los medios y componentes a utilizar para llevar a cabo las funcionalidades anteriores. Y, finalmente, en la implementación se especifican los programas elaborados para el funcionamiento del sistema, así como la constitución final de todos sus elementos de hardware.

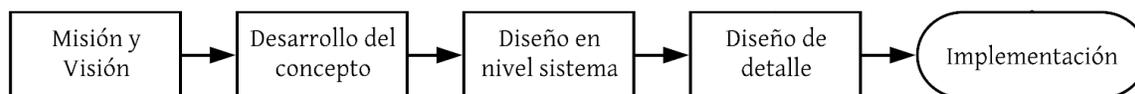


Figura 33. Etapas para el diseño del proyecto

### 4.1 Misión y visión

Durante el desarrollo de cualquier sistema, producto o prototipo, es imprescindible definir la misión y visión del proyecto, ya que reflejan el propósito del mismo, así como el compromiso de alcanzar metas específicas. La visión declara el objetivo ideal, sin la necesidad de definir tecnología o funciones particulares del dispositivo. Por otro lado, la misión establece el propósito y objetivo inmediato del proyecto, el cual se complementa con la Tabla 9, en donde se enlistan una serie de suposiciones bajo las cuales se operará, así como las diversas áreas de mercado potenciales.

**Misión:** Conformar los diferentes módulos que integren a un dispositivo detector de emociones económico, ergonómico y confiable, el cual tenga aplicación sobre múltiples áreas del conocimiento con el propósito de mejorar la calidad de vida del ser humano.

Tabla 9. Declaración de la misión

Declaración de la misión: detector de emociones	
<b>Descripción del producto</b>	<ul style="list-style-type: none"> <li>• Prototipo de un sistema detector de emociones</li> </ul>
<b>Propuesta de valor</b>	<ul style="list-style-type: none"> <li>• Clasificador preciso</li> <li>• Económico</li> <li>• Ergonómico</li> </ul>
<b>Mercados objetivo</b>	<ul style="list-style-type: none"> <li>• Sistema de salud</li> <li>• Bienestar en entornos laborales</li> <li>• Criminalística</li> <li>• Proceso de aprendizaje</li> </ul>
<b>Suposiciones y restricciones</b>	<ul style="list-style-type: none"> <li>• Emplear un dispositivo con capacidades matemáticas avanzadas</li> <li>• Implementación de elementos que propicien el desarrollo de un dispositivo ergonómico</li> <li>• Compatibilidad con otros entornos de desarrollo</li> </ul>

Es importante aclarar que algunos puntos abordados en la declaración de la misión, tal como la propuesta de valor, son tomados en cuenta como hipótesis y serán verificados durante el desarrollo del concepto.

**Visión:** Desarrollar el prototipo de un detector de emociones integrado en un sistema embebido, el cual cumpla con todas las especificaciones planteadas.

## 4.2 Desarrollo del concepto

En esta etapa se busca definir la forma, función y características que va a adquirir el proyecto, partiendo por identificar las necesidades del mercado objetivo, seguido de la descripción de éstas en un sentido técnico (especificaciones objetivo) para conformar diversos prototipos (conceptos). De ellos se analizan las diferentes opciones que presentan a fin de seleccionar uno solo que constituya el concepto final del proyecto. A continuación, se ahonda en cada uno de estos pasos a través de diversas secciones a fin de justificar las decisiones tomadas para su desarrollo.

### 4.2.1 Identificación de necesidades

El primer paso consiste en identificar las necesidades que busca satisfacer el proyecto, las cuales serán evaluadas a lo largo de la sección 4.2, por lo que es posible que no todas sean resueltas. A continuación, se enlistan las necesidades tanto generales como particulares y la relevancia de cada una de ellas. Esto último se logra asignando un número del 1 al 3, donde el 1 representa la mayor prioridad.

Tabla 10. Lista de necesidades involucradas con el diseño de un clasificador de emociones.

Necesidades generales	Necesidades particulares	Relevancia
Confiabilidad en la clasificación	<ul style="list-style-type: none"> <li>Mínimo error en el algoritmo utilizado para clasificar</li> <li>Óptima adquisición de datos del usuario</li> </ul>	1 1
Bajo costo	<ul style="list-style-type: none"> <li>Uso de componentes de bajo costo</li> </ul>	2
Intuitivo	<ul style="list-style-type: none"> <li>Fácil interpretación de los resultados</li> <li>Interfaz sencilla para el usuario</li> </ul>	1 2
Diseño multiplataforma	<ul style="list-style-type: none"> <li>Compatibilidad con otros entornos de desarrollo</li> </ul>	2
Bajo consumo energético	<ul style="list-style-type: none"> <li>Larga autonomía energética</li> </ul>	2
Ergonómico	<ul style="list-style-type: none"> <li>Utilizar elementos que permitan ergonomía</li> </ul>	1
Uso en conjunto con otros dispositivos	<ul style="list-style-type: none"> <li>Facilitar la portación de otros dispositivos</li> <li>Funcionalidad complementaria con otros dispositivos</li> </ul>	3 3

### 4.2.2 Especificaciones objetivo

Una vez determinadas las necesidades que busca satisfacer el detector de emociones, es necesario dar una descripción precisa en términos técnicos de las cualidades a las que se aspiran, lo cual ayudará a establecer las expectativas del proyecto. Dichas cualidades son conocidas como *especificaciones objetivo*, las cuales hacen uso de *métricas* que permitirán describirlas en términos medibles, así como establecer una serie de valores marginales e ideales para ellas. Se entiende por *valor ideal* al mejor resultado esperado en el dispositivo, y por otro lado, el *valor marginal* es un valor apenas aceptable.

Cabe resaltar que estas especificaciones se establecen antes de conocer las restricciones que impondrá la tecnología en la construcción del prototipo, por lo que es posible que algunas de ellas no se puedan satisfacer o inclusive se rebasen.

Tabla 11. Métricas de las necesidades

Necesidad	Métrica	Valor marginal	Valor ideal
Mínimo error en el algoritmo utilizado para clasificar	Exactitud	70%	90%
Óptima adquisición de datos del usuario	Razón señal a ruido (SNR) Frecuencia de muestreo	25 dB 500 Hz	50 dB 250 Hz
Uso de componentes de bajo costo	Costo total de los componentes de un sistema	Capacidad propia de financiamiento	
Fácil interpretación de los resultados	Fácil interpretación de los resultados	(Subj.)	(Subj.)
Interfaz sencilla para el usuario	Interfaz sencilla para el usuario	(Subj.)	(Subj.)
Compatibilidad con otros entornos de desarrollo	Número de familias de dispositivos en los que puede replicarse	1	>1
Larga autonomía energética	Número de técnicas para el ahorro de energía Duración del dispositivo activo	1 6 hrs	3 >24 hrs
Utilizar elementos que permitan ergonomía	Tamaño del dispositivo Tamaño de la batería	50 x 50 mm <sup>2</sup> 40 x 50 mm <sup>2</sup>	20 x 20 mm <sup>2</sup> <20 x 20 mm <sup>2</sup>
Facilitar la portación de otros dispositivos	Número de zonas del cuerpo para uso del dispositivo	1	5
Funcionalidad complementaria con otros dispositivos	Número de medios para fácil comunicación	1	4

Los valores marginales e ideales establecidos para las métricas de la Tabla 11 parten de la comparación de artículos o dispositivos semejantes que se relacionan a ellas. En el caso del factor SNR se designan los valores típicos que adquieren las señales fisiológicas en artículos que hacen uso de ellos, como son [72] y [45]. Las frecuencias de muestreo parten de la señal fisiológica que requiera la mayor tasa de muestreo posible, la cual corresponde a la señal obtenida a partir de la ECG, por lo que se utilizan sus valores mencionados en la sección 3.1.2. Los valores de las métricas relativos a la fácil interpretación de los resultados y la interfaz se denotan como subjetivo debido a que su cumplimiento depende de la percepción particular de cada individuo que lo utilice. A su vez, se representa el costo total de los componentes de un sistema mediante la frase “Capacidad propia de financiamiento” debido a que en principio se desconocen los componentes más adecuados para implementar el detector de emociones, ya que existe una amplia gama de dispositivos que cumplen con las especificaciones planteadas y cuyo precio es muy variado sin importar el tipo de tecnología, y por ende el límite de este valor se refiere a la capacidad monetaria disponible por los autores de esta tesis. En cuanto a la duración y las dimensiones del dispositivo se comparan estas cualidades con diversos dispositivos vestibles de uso comercial.

#### 4.2.3 Generación de conceptos

La generación de conceptos consiste en conformar diversos prototipos del proyecto que cumplan satisfactoriamente con las funcionalidades que contempla en cada etapa de su operación, esto con el objetivo de resolver el problema principal que se plantea. Sin embargo, dada la variedad de configuraciones existentes, en lugar de especificar uno a uno los conceptos propuestos, se opta por

presentarlos de forma comprimida, mostrando un compilado de todas las posibles soluciones para cada funcionalidad del dispositivo, lo cual da lugar al esquema de la Figura 34. Éste último será utilizado para especificar el concepto final detallado a lo largo de la sección 4.2.4.

El esquema se encuentra dividido en seis secciones principales: la configuración del dispositivo, adquisición de señales fisiológicas, acondicionar la señal fisiológica, extraer características de la señal fisiológica, clasificar emociones y comunicar la emoción clasificada.

En el caso de la *configuración del dispositivo* se enlistan dos vertientes, siendo éstas las variables de configuración y la interfaz. El primer caso hace referencia a aquellas variables que permiten el adecuado funcionamiento del detector, así como parámetros que se consideran relevantes para la ejecución de diversas funcionalidades y algoritmos. En el segundo caso se hace referencia a la manera en que el usuario se comunica con el dispositivo para realizar la configuración de éste.

La *adquisición de señales fisiológicas* es un compilado de todas las señales útiles para la identificación de estados emocionales de acuerdo con lo visto en el Estado del Arte. De igual forma, en esta rama se realiza un resumen de las técnicas de obtención estipuladas en la sección 3.1 para las señales utilizadas en la presente tesis, así como de aquellas no contempladas para el diseño final.

*Acondicionar la señal fisiológica* muestra las diversas maneras en que las señales adquiridas pueden ser procesadas para obtener una fiel representación de la variable medida, tomando en consideración la etapa de digitalización, amplificación y filtrado.

En *extraer características de la señal fisiológica* se hace un listado de las características más relevantes que pueden ser obtenidas a partir de las señales fisiológicas. Entre estas características se encuentran aquellas en el dominio del tiempo, la frecuencia y las obtenidas mediante métodos no lineales. Éstas se detallan de mejor manera en la sección 3.2.

*Clasificar emociones* muestra un resumen de las consideraciones para la estimación de emociones, como lo puede ser la perspectiva y el modelo, lo cual se puede ver con mayor detalle en la sección 3.7. Asimismo, se realiza un compilado de los algoritmos que tienen mejor desempeño en la identificación de emociones a partir de señales fisiológicas, siendo aquellos indicados en la sección 3.3.

En *comunicar la emoción clasificada* se enlistan las formas en que el usuario se puede informar sobre su estado emocional, ya sea en el propio dispositivo o por un medio externo.

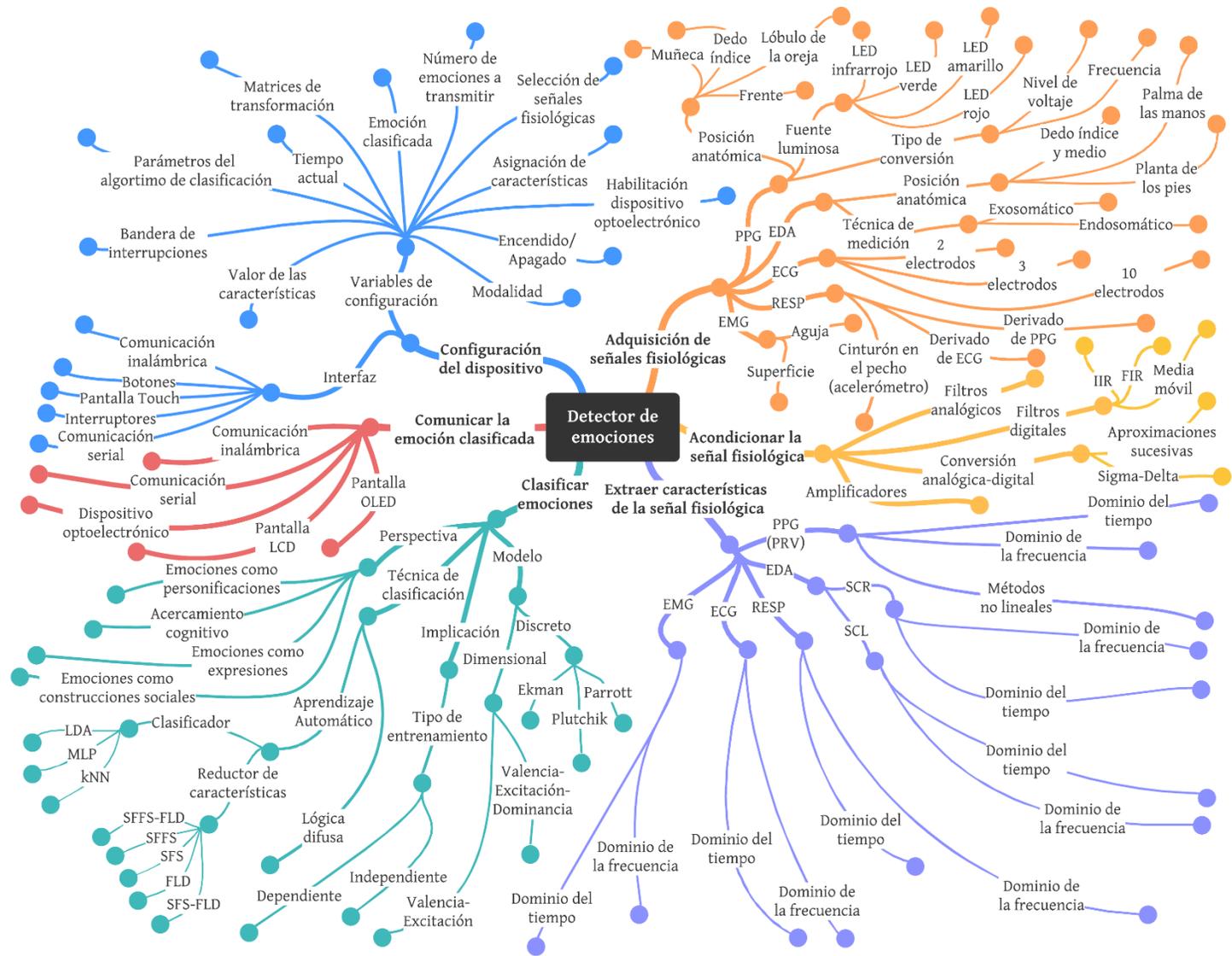


Figura 34. Descomposición funcional

#### 4.2.4 Selección de concepto

Una vez desglosadas las posibles soluciones a cada funcionalidad del dispositivo, es preciso realizar un análisis de las características de cada una de ellas con el objetivo de mantener únicamente las más factibles para el prototipo final. Como se mencionó en la sección 4.2.3, a continuación, se presentará únicamente el resultado de aquella selección, haciendo un desarrollo de cada atributo elegido en conjunto con su justificación.

Inicialmente, en el Estado del Arte los diferentes autores emplearon diversas señales fisiológicas que demostraron ser de gran utilidad para la detección de emociones, en cuyas conclusiones se enlistaron aquellas que brindaron los mejores resultados. Por otro lado, en el marco teórico se profundizó en sus características eléctricas, dispositivos médicos y puntos anatómicos adecuados para su adquisición. Analizando la información recabada es posible seleccionar las señales que se ajustan mejor a las especificaciones planteadas para el prototipo.

La frecuencia cardíaca ha demostrado ser indispensable en la detección de emociones debido a su conexión tanto con la rama simpática como con la rama parasimpática del SNA. Anteriormente se estudió que la actividad cardíaca es comúnmente analizada mediante técnicas como la PPG o el ECG. Por las razones expuestas en la sección 3.1.3, en la cual se realiza una comparación de ambas técnicas, se concluye que la implementación del ECG representa un obstáculo para cumplir con el concepto del dispositivo portátil planteado, mientras que la PPG resuelve todos aquellos inconvenientes y, como se demostró, brinda excelentes resultados. Dicho lo anterior, se decide optar por la PPG como técnica para analizar la actividad cardíaca y extraer las características correspondientes. Por otro lado, la sudoración de la piel manifestó ser un excelente indicador durante situaciones de estrés. A pesar de ser una respuesta fisiológica relativamente lenta, brinda información de gran valor gracias a las componentes fásicas (SCR) de la señal. La EDA es posiblemente la señal fisiológica más útil en cuanto a cambios de excitación, ya que no se encuentra contaminada por la actividad parasimpática, debido a que se rige únicamente por la rama simpática del SNA. Por lo tanto, la PPG y la EDA son seleccionadas debido a la ergonomía y puntos anatómicos sugeridos para sus sensores, información valiosa proporcionada y buenos resultados en el Estado del Arte. Además, agregar más señales fisiológicas al sistema resultaría inconveniente para el usuario y dificultaría el desarrollo de un dispositivo portátil.

Una vez definidas las señales fisiológicas y técnicas empleadas para su adquisición, es necesario seleccionar las propiedades para la medición y posiciones anatómicas que permitan obtener los resultados más favorables. Respecto a la PPG es preciso seleccionar correctamente la longitud de onda de la fuente luminosa, ya que ésta influye notablemente en la fidelidad de los datos. Por lo tanto, en la sección 3.1.3 se realizó el análisis de diversos artículos y con base en sus resultados, se decide implementar luz verde debido al buen rendimiento que presenta en comparación con las demás longitudes de onda. Otro aspecto por considerar es la posición anatómica óptima del sensor, la cuál es considerada en el dedo índice (específicamente la falange distal), debido a que es ampliamente recomendado tanto en estudios como en libros especializados, ya que las características de dicha zona favorecen la adquisición de datos. El tipo de conversión a utilizar para la PPG será por nivel de voltaje dada su simplicidad. En cuanto a la EDA, en la sección 3.2 se presentaron las técnicas de medición para calcular la conductancia de la piel, de las cuales la más empleada en la literatura es la exosomática, por lo que se decide adoptar dicho método. La posición anatómica se selecciona de acuerdo con las recomendaciones de diversos textos citados en la sección 3.1.4 del Marco Teórico, donde se argumenta que los puntos anatómicos con mayor densidad de glándulas sudoríparas presentan las mejores condiciones para la medición, estas zonas son las palmas de las manos y los dedos índice y medio.

Con el objetivo de adquirir las señales fisiológicas con la mayor fidelidad posible es necesario darles un tratamiento previo a su análisis, también llamado  *acondicionamiento* , que permita atenuar ruido no deseado, amplificar y, finalmente, digitalizar la señal para emplearla en un medio electrónico.

Para eliminar las perturbaciones en todas las señales se optó por implementar filtros digitales, pues éstos implican algunas ventajas respecto a los filtros analógicos. En primer lugar, se ofrece una estabilidad en las frecuencias de corte a diferencia de los analógicos que pueden variar por factores físicos como la temperatura; de igual manera, implican una implementación sencilla, dado que no necesitan de hardware extra y, por último, su funcionamiento es fácilmente modificable, pues a diferencia de los filtros analógicos, el tipo de filtro y las frecuencias de corte pueden ser modificados simplemente cambiando el valor de sus coeficientes.

El filtro digital seleccionado es el filtro FIR, pues trae consigo ventajas en cuanto a estabilidad y fase lineal, a diferencia de los filtros IIR que, si bien son superiores en cuanto a la pendiente generada en la banda de transición y el bajo número de coeficientes, resultan más inestables dada la precisión finita de los cálculos matemáticos, lo que puede ocasionar una modificación de la posición de sus polos y llevarlos fuera del círculo unitario, poniendo en peligro el buen funcionamiento del sistema.

Debido a que se utilizará un filtro digital, será necesaria su implementación una vez sea amplificada y digitalizada la señal. Dado que ambos factores dependen de las propiedades de los sensores biométricos seleccionados, se buscará en ellos cualidades que garanticen una buena amplificación y digitalización. Para este último caso, será conveniente buscar dispositivos que utilicen el método de aproximaciones sucesivas dado su bajo coste y eficacia.

Dentro de la *extracción de características*, se escogieron aquellos parámetros representativos de las señales fisiológicas que dieran noción de los cambios que ocurren en éstas al evocarse una emoción. Para ello, a partir de la señal obtenida por PPG se conformará y analizará un tacograma, considerando algunos de sus parámetros, expuestos previamente en la sección 3.2.3.1, correspondientes a NN50, pNN50, PPImedia, SDNN y RMSSD como estimaciones estadísticas en el dominio del tiempo, así como los valores SD1, SD2 y su razón SD1/SD2 mediante el método no lineal de la gráfica de Poincaré dados sus amplios usos en los trabajos del Estado del Arte. En el dominio de la frecuencia, se estudiarán las bandas VLF, LF y HF, así como la razón LF/HF, dada la relación que mantienen las dos primeras bandas con respuestas del SNS y la última con la actividad del SNP. Para esta estimación se emplea el método paramétrico de Burg debido a la buena resolución que otorga para señales con SNR bajos, alta fidelidad que brinda para señales con pocos datos y la nula existencia de derrames espectrales.

La elección de realizar estimaciones en el dominio de la frecuencia conlleva cierta restricción en el rango de aplicaciones sobre las cuales resultará útil el detector. Con estas estimaciones surge la necesidad de emplear ventanas de datos prolongadas que permitan obtener resoluciones óptimas para medir adecuadamente el promedio de las bandas VLF, LF y HF. Por ende, el dispositivo desarrollado plantea como un detector de emociones prolongadas de propósito general, dado que las ventanas mencionadas anulan la posibilidad de implementar aplicaciones de respuesta menor a los 64 s.

Por otro lado, para la EDA se elige analizar la señal SCR en lugar de SCL (señales explicadas en la sección 3.1.4) dado que, a pesar de que ambos dan información respecto a cambios en el SNS, éste último resulta complicado de diferenciar con respecto a la línea base de la señal ya que su naturaleza yace comprendida por debajo de las frecuencias de 0.16 Hz, lo cual puede complicar la distinción de la señal con factores externos como pueden ser los movimientos corporales del usuario. Por ende, las características por extraer de la SCR serán aquellas abordadas en la sección 3.2.3.2, que en el dominio del tiempo corresponden al número de eventos ER-SCR, dado que de estos se identifican las reacciones a estímulos específicos, además de la media de sus tiempos de levantamiento, la desviación estándar y media de sus amplitudes, así como la estimación de cuantiles al 25%, 50%, 75% y 90%. En el dominio de la frecuencia se analizará del intervalo de frecuencias correspondiente a EDA de 0.045-0.15 Hz, una banda denotada como HF (0.15-0.5 Hz), cuyo análisis se realizará mediante su PSD empleando el método de Burg. Cabe mencionar que el resto de los parámetros para PPG y EDA expuestos en la sección 3.2.3 cuya omisión se realiza en la selección anterior es a causa de su poca efectividad y redundancia de información dentro de los trabajos expuestos en el Estado del Arte.

Asimismo, se implementarán técnicas de Aprendizaje Automático para clasificar y reducir las características extraídas dentro del sistema detector de emociones.

En primer lugar, se utilizará una mezcla entre SFFS y FLD (dos algoritmos de reducción de características vistos en la sección 3.4) con el objetivo de eliminar aquellas características que resulten menos útiles para la clasificación y, posteriormente, comprimir su información de modo que se reduzca el número de operaciones necesarias en las etapas subsecuentes. Dichos algoritmos fueron seleccionados dada la efectividad en conjunto demostrada en el Estado del Arte. Cabe resaltar que hasta el momento se ha referido al algoritmo FLD como FP dada la notación utilizada en los trabajos revisados, sin embargo, se considera que FLD es una notación más adecuada ya que alude al verdadero nombre del algoritmo y será utilizada a partir de este punto.

Como algoritmo de clasificación se planea utilizar LDA (abordado en la sección 3.5.1) dada su efectividad demostrada en el Estado del Arte, lo que permitió prescindir del proceso de validación de modelos especificado en la sección 3.3.5. Asimismo, tras la investigación realizada a lo largo del Marco Teórico, se evidenció la ligereza computacional de este algoritmo, lo cual beneficia en gran medida al cumplimiento de las especificaciones objetivo de la sección 4.2.2. Por último, se planea utilizar el esquema EMDC para garantizar un mejor rendimiento en la clasificación de emociones como se demuestra en [8].

Con la intención de reducir la carga computacional del detector de emociones en cuanto a la etapa de Aprendizaje Automático, se opta por implementar el entrenamiento de sus algoritmos fuera del sistema embebido, siendo este último solo encargado de recibir los parámetros de los mismos e implementarlos para la clasificación de emociones. La etapa de entrenamiento se realizará por medio de una computadora externa. Es necesario recordar que el Aprendizaje Automático consta de dos etapas, siendo el entrenamiento de los algoritmos y su puesta en funcionamiento. Es por este motivo que se considera necesario dividir en dos modalidades distintas tanto el detector de emociones como el sistema generado en la computadora, las cuales serán descritas con mayor detalle en la sección 4.3.

Respecto a los *modelos* de la emoción, abordados previamente en la sección 3.7.2, se decide utilizar el *modelo dimensional* del círculo circunflejo de Russel dado que se considera el más conveniente para llevar a cabo la implementación del esquema EMDC gracias a la repartición que ofrece de las emociones en los ejes de valencia y excitación, además de resultar el más utilizado en el Estado del Arte. De esta forma, el enfoque del trabajo resultará más bien en la detección de grupos de emociones alojadas en las diversas regiones que ofrece el modelo mencionado, siendo las cinco regiones por clasificar en este trabajo: excitación alta-valencia positiva (EA-VP), excitación baja-valencia positiva (EB-VP), excitación alta-valencia negativa (EA-VN), excitación baja-valencia negativa (EB-VN) y una región Neutra. Asimismo, se utilizará el esquema SAM, ilustrado en la Figura 32, para evaluar las emociones evocadas por los usuarios dada su sencillez y facilidad que ofrece para evaluar los grados de valencia y excitación de forma gráfica, lo cual permite su comprensión en un ámbito multicultural. Se contempla el manejo de nueve niveles para cada uno de los ejes de valencia y excitación del modelo SAM, esto facilita la evaluación de las emociones por parte del usuario.

En cuanto al ámbito de las *perspectivas de la emoción* se decidió adoptar aquella que contempla a las *emociones como personificaciones* dado que, como se explicó previamente en la sección 3.7.1, el uso de señales fisiológicas denota inherentemente una expresión del SNS del cual se basa esta perspectiva.

La selección de alguna de las *implicaciones* abordadas en la sección 3.7.3 posee una estrecha relación con el tipo de aplicaciones en las cuales es posible utilizar el detector de emociones, ya que de ellas es necesario identificar si es posible evocar las emociones necesarias para el estudio en el individuo. Por ejemplo, existen casos como el apoyo a la detección de trastornos psicológicos (tales como la depresión y/o ansiedad) que requieren tanto de individuos que genuinamente presenten dichas patologías, siendo casos clínicamente comprobados, como de otro grupo de personas que no las

presenten, garantizando un contraste en la clasificación y haciendo necesario que el sistema se configure como un clasificador independiente. En cambio, para el apoyo a la detección de empleados que sufran un alto nivel de estrés debido a su entorno laboral, los sujetos de prueba resultan ser personas ordinarias cuya afección del estrés puede evitarse o evocarse de forma individual, abriendo las puertas para implementar un clasificador dependiente.

Para comunicar la emoción que se encuentre clasificando el sistema, se decide utilizar un dispositivo optoelectrónico cuyos colores se asocien a las emociones suscitadas, esto con el propósito de brindar una fácil visualización del funcionamiento del sistema, o bien hacer uso de una comunicación inalámbrica que permita emplear los resultados del detector en otras aplicaciones.

En el caso de la interfaz que permite realizar la configuración del dispositivo y comunicar la emoción clasificada por medio de un dispositivo externo, se eligió una transmisión *inalámbrica* que brinda portabilidad y un sentido ergonómico al detector de emociones. Asimismo, utilizando dicha interfaz es posible elegir diversas opciones de configuración que permiten el accionamiento del sistema, uso del dispositivo optoelectrónico, modalidad bajo la cual operará, seleccionar las señales fisiológicas y características a extraer, depositar los parámetros requeridos por el clasificador, así como recoger la emoción que clasifique el sistema y las características que extraiga para una determinada emoción.

Mediante las elecciones realizadas en los párrafos anteriores, resulta posible ilustrar mediante la Figura 35 la estructura que sigue el concepto final del proyecto considerando sus diversas funcionalidades antes descritas.

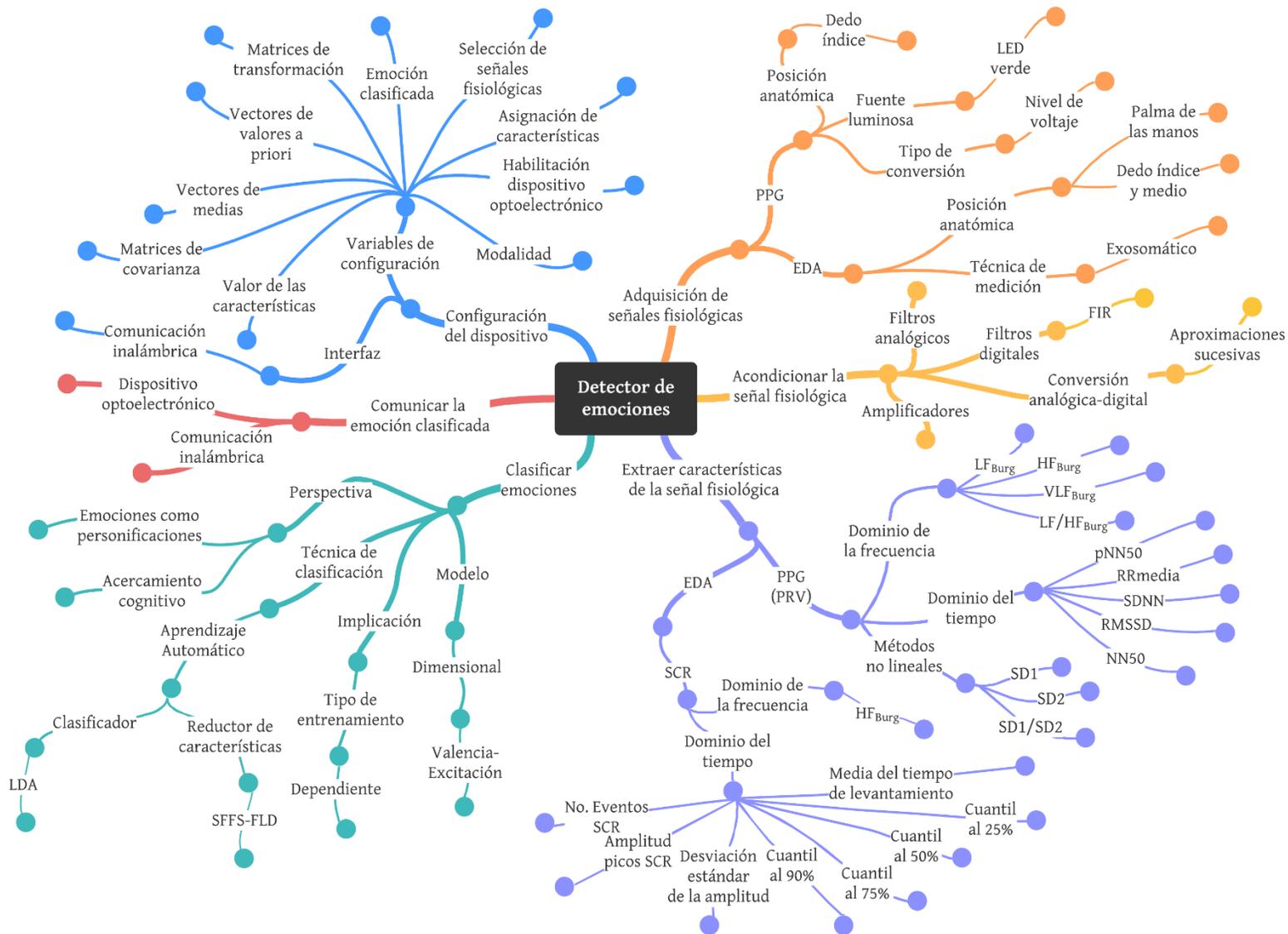


Figura 35. Concepto final para el detector de emociones

### 4.3 Diseño en el nivel sistema

Una vez seleccionado el concepto final del proyecto, es posible definir su arquitectura, la cual se divide en subsistemas que, a su vez, se encuentran conformados por diversos componentes. En esta etapa se parte el esquema general en dos bloques, los cuales representan los procesos que se realizan tanto en el sistema embebido como en la computadora. El sistema embebido se encuentra dividido en cinco subsistemas principales: comunicaciones, acondicionamiento de la señal, extracción de características, clasificación y configuración. Por otro lado, la computadora yace dividida en cinco subsistemas: Comunicación RF, la interfaz gráfica de usuario (GUI, por sus siglas en inglés), SFFS, FLD y LDA. A continuación, se presenta el diseño geométrico y especificaciones funcionales de cada uno de estos subsistemas, además del diagrama de flujo preliminar que ilustra el proceso que llevan a cabo.

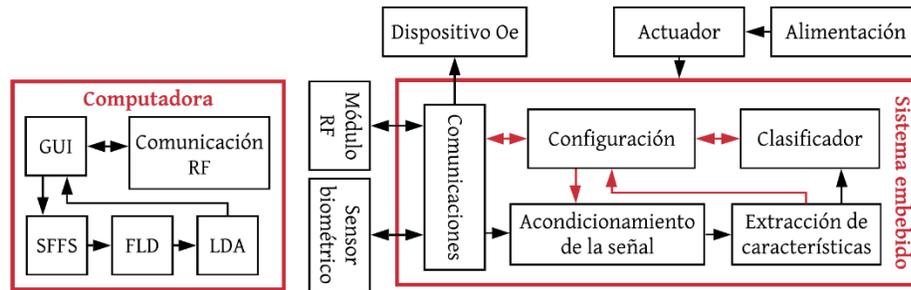


Figura 36. Diagrama general del sistema

No obstante, previo a comenzar con la descripción de cada uno de los subsistemas, se opta por asignar a ambos bloques (Sistema embebido y Computadora) ciertas modalidades que describen las diferentes etapas bajo las cuales puede operar el sistema en conjunto. Por ende, ambos bloques deben coincidir en la operación bajo una misma modalidad para el correcto funcionamiento del sistema. Dentro de la Tabla 12 se ilustran las distinciones que presentan cada una de ellas de acuerdo con las funcionalidades que ejecuta sobre el detector de emociones o la computadora externa.

El modo de *entrenamiento* está enfocado a la adquisición y envío de las características extraídas de las señales fisiológicas a una computadora externa al accionarse un actuador. Ésta realizará el entrenamiento del clasificador con los datos recibidos y las emociones evocadas al momento de su extracción. Por otro lado, las emociones serán asignadas por el usuario mediante un diagrama con el esquema SAM desplegado en una ventana del subsistema GUI, cuyos resultados serán grabados en la computadora. Posteriormente, considerando las elecciones realizadas por los diversos selectores SFFS y los entrenamientos de los diferentes reductores FLD y clasificadores LDA respectivos a cada etapa del esquema EMDC. El sistema embebido deberá recibir los valores de las matrices de reducción, denotada como  $W_n$ , información sobre las características seleccionadas de las señales fisiológicas dada las reducciones SFFS y los parámetros  $\vec{m}_k$ ,  $S^{-1}$  y  $\ln(\pi_k)_n$  de LDA requeridos en cada etapa del esquema EMDC para así avanzar al modo de *Prueba*. Dentro de la nomenclatura anterior  $k$  representa el número de clases y  $n$  el número de etapas del esquema EMDC.

El modo de *prueba* se encarga de realizar la clasificación de las emociones del usuario con base en las características extraídas de sus señales fisiológicas y los valores recibidos en el modo de *Entrenamiento*. En este modo sólo se realiza la transmisión de las emociones clasificadas a la computadora externa a fin de contar con una gráfica de barras que permita visualizar el desempeño del detector a lo largo de extensos periodos de tiempo. La GUI dispondrá de un indicador que despliegue el tiempo total del historial de datos adquirido que permita al usuario realizar sesiones de extensión variable que vaya acorde al interés de sus estudios a desarrollar sobre los sujetos de prueba. Además, se dispone de un botón que permite reiniciar el historial al eliminar todo su contenido.

Tabla 12. Modos de operación para el detector de emociones y la computadora externa

Detector de emociones		Modalidad	
		Entrenamiento	Prueba
Comunicación a computadora	Síncrono (64 seg)	<b>Tx</b> - Características	<b>Tx</b> - Emoción
	Asíncrono	<b>Rx</b> - Parámetros $\vec{m}_k, S, \hat{\pi}_k, W$ y datos de configuración	<b>Rx</b> - Datos de configuración
Computadora externa		Modalidad	
		Entrenamiento	Prueba
Interfaz		Botones de configuración	Botones de configuración Gráfica de emoción vs tiempo
Memoria		Añade contenido <ul style="list-style-type: none"> <li>• Características</li> <li>• Emoción</li> </ul> Borra contenido <ul style="list-style-type: none"> <li>• Características</li> <li>• Emoción</li> </ul>	
Gráfica de emoción		Se omite	Se realiza
Comunicación a detector	Síncrono (64 seg)	<b>Rx</b> - Características	<b>Rx</b> - Emoción
	Asíncrono	<b>Tx</b> - Parámetros $\vec{m}_k, S, \hat{\pi}_k, W$ y datos de configuración	<b>Tx</b> - Datos de configuración

### 4.3.1 Configuración

El subsistema de configuración, representado por la Figura 37, yace conformado por variables que cumplen diversas funcionalidades sobre los diferentes subsistemas del detector de emociones. Son accesibles de forma paralela mediante los subsistemas de comunicaciones, de Clasificación y de extracción de características. A continuación, se describen las funcionalidades de cada una de estas variables, así como sus requerimientos en longitud de bits.

- 1) **Medición:** (1 bit) Variable que habilita o desactiva el funcionamiento del sensor biométrico para realizar las mediciones de las señales PPG y EDA.
- 2) **Modalidad:** (1 bit) Variable que indica el modo de operación del detector de emociones, variando entre *entrenamiento* o *prueba*.
- 3) **Habilitación del dispositivo optoelectrónico:** (1 bit) Variable que habilita el uso del dispositivo optoelectrónico para comunicar visualmente la emoción clasificada por el sistema.
- 4) **Asignación de características:** (4 elementos de 22 bits) Variable que designa el uso u omisión de cada una de las 22 características extraíbles de las señales PPG y EDA. El manejo de 4 valores radica en el uso de una variable por cada etapa del esquema EMDC.
- 5) **Emoción:** (3 bits) Variable codificada que denota la emoción adquirida del clasificador de emociones.
- 6) **Valor de características:** (32 bits) Conjunto de variables que contienen los valores de cada una de las 22 características extraíbles de las señales PPG y EDA.
- 7) **Matrices de transformación  $W_n$ :** (4 vectores con 22 elementos de 32 bits en su interior) Conjunto de variables que contienen los valores de cada uno de los elementos de las matrices

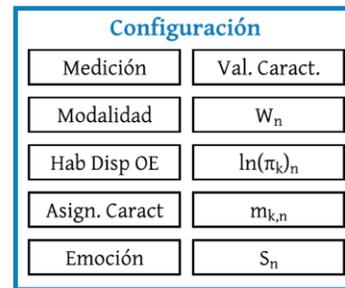


Figura 37. Desglose del subsistema de configuración

de transformación empleadas por los reductores de características FLDn dentro del subsistema de clasificación. La razón del manejo de 4 vectores radica en el uso de un vector por cada etapa del esquema EMDC, mientras que la asignación de 22 valores responde al peor caso posible donde el selector SFFS seleccione todas las características para el clasificador.

- 8) **Vectores de valores a priori  $\ln(\pi_k)_n$ :** (4 vectores con 2 elementos de 32 bits en su interior) Conjunto de variables que contienen el logaritmo natural de los valores a priori utilizados por los diferentes clasificadores LDAn dentro del subsistema de clasificación. El uso de 4 vectores radica en el uso de un vector por cada etapa del esquema EMDC, mientras que la asignación de 2 valores responde a la clasificación binaria que se realiza en cada uno de sus clasificadores LDA.
- 9) **Vectores de medias  $m_{k,n}$ :** (4 vectores con 2 elementos de 32 bits en su interior) Conjunto de variables que contienen los valores de las medias de cada una de las características empleadas por los diferentes clasificadores LDAn dentro del subsistema de clasificación. El uso de 4 vectores radica en el uso de un vector por cada etapa del esquema EMDC, mientras que la asignación de 2 valores responde a la clasificación binaria que se realiza en cada uno de sus clasificadores LDA.
- 10) **Matrices de covarianza  $S_n$ :** (4 elementos de 32 bits) Conjunto de variables que contienen los valores de cada uno de los elementos de las matrices de covarianza empleadas por los clasificadores LDAn dentro del subsistema de clasificación. El uso de 4 valores radica en el uso de una variable por cada etapa del esquema EMDC.

Las variables que requieren de 32 bits se designan de dicha forma dado que se busca emplear una aritmética de punto flotante, lo cual implica el requerimiento de una unidad de punto flotante (FPU, por sus siglas en inglés). Ésta se utilizará debido a que ofrece un amplio intervalo dinámico que resolverá la incertidumbre que se tiene de la magnitud de los valores a obtener en cada una de estas variables. Mientras tanto, la velocidad de recepción de las variables anteriores será determinada primordialmente por los demás subsistemas que las utilicen, por lo que sus requerimientos se explicarán en conjunto con éstos.

Todas las variables anteriores se estructuran dentro de un mapa de registros mediante el cual sea posible configurar y leer los datos que genere el detector. Esta estructura estará basada en direcciones a fin de brindar compatibilidad con otros dispositivos externos que requieran utilizar al detector como complemento en otras aplicaciones.

### 4.3.2 Comunicaciones

Este subsistema, representado en la Figura 38, tiene el objetivo de realizar la comunicación de las variables internas del sistema al exterior a través de un bloque GPIO y un bloque de Comunicación serial. Su funcionamiento resulta de carácter asíncrono dado que usualmente el manejo de protocolos seriales solicita el uso de interrupciones.

Por medio del bloque GPIO se realiza la comunicación con un dispositivo optoelectrónico (Oe) con el fin de recibir el dato de la variable “Emoción” y comunicar visualmente alguno de los siguientes tres casos: la emoción que se encuentre clasificando el sistema, la suscitación de un error por desconexión del módulo RF o la calibración del sensor biométrico. Asimismo, se emplea otro puerto GPIO para recibir la alerta emitida por el sensor biométrico externo y notificar al bloque de Comunicación serial de la disponibilidad de un nuevo dato a leer del sensor.

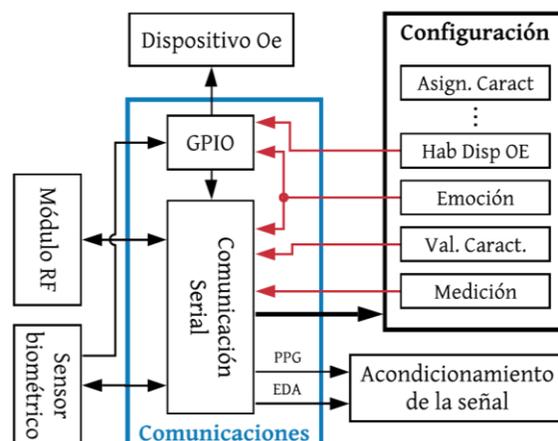


Figura 38. Desglose del subsistema de comunicaciones

A través del bloque de Comunicación serial se realiza la transferencia de datos con el sensor biométrico para configurarlo, extraer de éste los valores de las señales fisiológicas PPG y EDA y, posteriormente, transmitirlos al subsistema de acondicionamiento de la señal. Por ende, para éste se requiere de un sensor de bajo costo, ergonómico y capaz de manejar ambas señales fisiológicas, además de disponer de módulos de amplificación y digitalización de éstas, tal como se mencionó en la sección 4.2.4. Igualmente, mediante el bloque de Comunicación serial se entabla el enlace con un módulo RF que permite la conexión de un dispositivo externo para realizar la escritura y lectura de todas las variables del subsistema de configuración. Para este último se requiere de un módulo de bajo costo que maneje un protocolo RF de bajo consumo y posea una longitud de enlace media de alrededor de unas cuantas decenas de metros.

A continuación, se desglosan los diversos requerimientos en longitud de bits y espacio de memoria necesarios para entablar la comunicación con los bloques internos del subsistema de Comunicaciones:

- 1) **GPIO:** (4 valores de longitud variable) Bloque con comunicación paralela a los siguientes elementos:
  - a. *Dispositivo optoelectrónico:* (7 bits) uso de siete puertos para representar los estados de emoción, error por desconexión y calibración, con una velocidad de comunicación equivalente a la velocidad de clasificación que ofrezca el subsistema de clasificación al entregar la variable “Emoción”
  - b. *Sensor biométrico:* (1 bit) uso de un puerto que notifica cuando hay un nuevo dato del sensor biométrico. Su velocidad deberá corresponder a la tasa de muestreo de las señales PPG y EDA.
  - c. *Subsistema de configuración:* 2 variables respectivas a “Habilitación del dispositivo optoelectrónico” y “Emoción” cuyas longitudes corresponden a las descritas en la sección 4.3.1.
- 2) **Comunicación Serial:** (9 valores de longitud variable) contempla la comunicación serial con los bloques de Módulo RF y Sensor biométrico, mientras que con el subsistema de configuración y acondicionamiento de la señal entabla una comunicación paralela. Los requerimientos de cada una de sus entradas y salidas son las siguientes:
  - a. *Módulo RF:* la longitud de bits requerida se encuentra relacionada con el protocolo a utilizar, sin embargo, éste debe ser compatible con el módulo a emplear, además de contemplar la necesidad de recibir y transmitir todas las variables del subsistema de configuración descritas en la sección 4.3.1.
  - b. *Sensor biométrico:* (8 bits) Tras una exploración a través de los diferentes sensores biométricos disponibles en el mercado, se observó que uno de los medios más ergonómicos para la lectura de una señal biométrica es mediante un sensor externo que en su mayoría se ofrece con un convertidor analógico-digital (ADC, por sus siglas en inglés) integrado y una comunicación serial mediante el protocolo de comunicación serial I2C. Por ende, éste último resultará el requerimiento para su comunicación con el bloque de Comunicación serial, además de solicitar una tasa de transmisión suficiente que permita la adquisición de los datos de ambas señales PPG y EDA conforme a sus frecuencias de muestreo que se les designen en la siguiente etapa de diseño.
  - c. *GPIO:* (1 bit) Traspasa el estado de notificación recibido por el sensor biométrico para que el bloque realice la lectura del nuevo dato.
  - d. *Subsistema de configuración:* 4 variables respectivas a “Emoción”, “Valor de características”, “Medición” y “Habilitación del dispositivo optoelectrónico” cuyas longitudes corresponden a las descritas en la sección 4.3.1.
  - e. *Subsistema de acondicionamiento de la señal:* 2 variables respectivas a PPG y EDA cuya longitud de bits requerida se encuentra en función de la resolución que ofrezca el ADC del sensor biométrico y su velocidad está asociada a la tasa de muestreo requerida para cada una de las señales.

### 4.3.3 Acondicionamiento de la señal

En este subsistema se lleva a cabo el preprocesamiento de las señales fisiológicas, su propósito es atenuar el ruido no deseado contenido en la señal para su posterior análisis en el subsistema de extracción de características. Los elementos que conforman el acondicionamiento de la señal son dos filtros digitales FIR, correspondientes a cada una de las señales fisiológicas (PPG y EDA).

A continuación, se desglosan las características de las variables que son recibidas a través del subsistema de Comunicaciones, así como aquellas transmitidas al subsistema de extracción de características:

- 1) **Comunicaciones:** (2 valores) Se reciben las variables de las señales PPG y EDA cuya longitud dependerá de la resolución que ofrezca el ADC del sensor biométrico. Asimismo, su velocidad está asociada a la tasa de muestreo requerida para cada una de las señales.
- 2) **Extracción de características:** (2 valores de 16 bits) La salida del subsistema será un número en formato de punto fijo de 16 bits con el objetivo de cubrir los requisitos del subsistema de extracción de características.

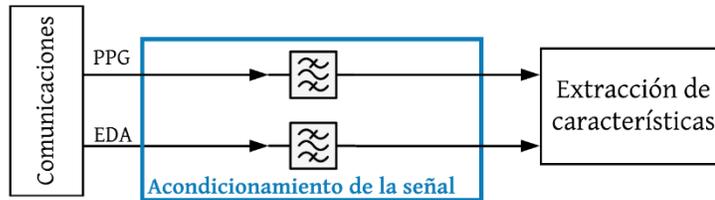


Figura 39. Desglose del subsistema de acondicionamiento de la señal

### 4.3.4 Extracción de características

Una vez realizado el acondicionamiento de las señales fisiológicas, este subsistema se encarga de analizarlas con el objetivo de extraer sus características más representativas, las cuales fueron definidas en la Selección de concepto. Respecto a la señal obtenida a partir de la PPG, primeramente se obtiene el tacograma (Figura 16) a través del bloque PRV, en el cual se realiza la detección de los picos sistólicos y el cálculo de los intervalos pico a pico. Posteriormente, la señal es dirigida a los bloques subsecuentes donde es procesada por los métodos correspondientes al análisis no lineal de la gráfica de Poincaré, análisis en el dominio del tiempo y de la frecuencia. Por otro lado, la señal EDA es tratada por el bloque SCR con el objetivo de extraer dicha componente, la cual es dirigida a los bloques de tiempo y frecuencia. Con la variable “Modalidad” se define si el sistema se encuentra en modo *entrenamiento* o *prueba*, por lo que es posible gestionar la transferencia de características a los bloques correspondientes según la modalidad.

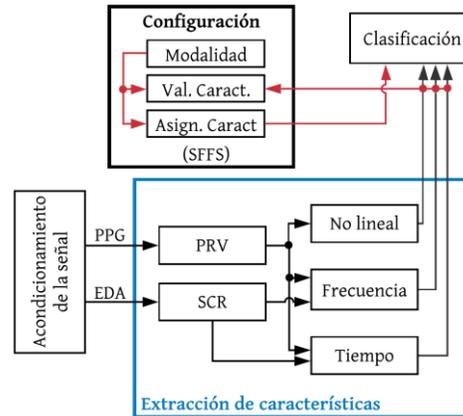


Figura 40. Desglose del subsistema de extracción de características

**Entrenamiento:** cuando se establece el modo de *entrenamiento*, todas las características son calculadas y almacenadas como variables en “Valor de características” para ser enviadas por el subsistema de Comunicación a una computadora externa, donde se realiza el entrenamiento del clasificador.

**Prueba:** cuando el sistema se encuentra en modo de *prueba*, únicamente son calculadas las características seleccionadas por el reductor SFFS, las cuales son indicadas mediante la variable “Asignación

de características”, además, las características son enviadas directamente al subsistema de clasificación.

A continuación, se desglosan las propiedades de las variables que son recibidas y transmitidas a los bloques correspondientes del sistema:

- 1) **Acondicionamiento de la señal:** (2 valores de 16 bits) La entrada del subsistema será un número en formato de punto fijo de 16 bits con el objetivo de cubrir los requisitos del subsistema de extracción de características.
- 2) **Clasificación:** (22 valores de 32 bits) La salida comprenderá el valor de las 22 características extraídas, las cuales deberán estar en un formato de punto flotante para ser recibidas por los bloques FLDn del subsistema de clasificación.

#### 4.3.5 Clasificación

El subsistema de clasificación, representado por la Figura 41, está activo durante la modalidad de *prueba* del prototipo, la cual está indicada por medio de la variable “Modalidad”. Su función inicial es realizar una compresión de los datos (anteriormente seleccionados por el algoritmo SFFS) provenientes del subsistema de extracción de características a través del algoritmo FLD para que, posteriormente, sea posible identificar la región del modelo dimensional en el que se encuentra la persona al momento de la extracción. En total se distinguen cinco regiones distintas correspondientes al estado neutro y cuatro emociones representativas de los cuatro cuadrantes del modelo dimensional, tal como se expresó en la sección 4.2.4. Para ello se realiza una clasificación de las características comprimidas a través del esquema EMDC, el cual comprende un árbol de clasificaciones binarias que inicialmente

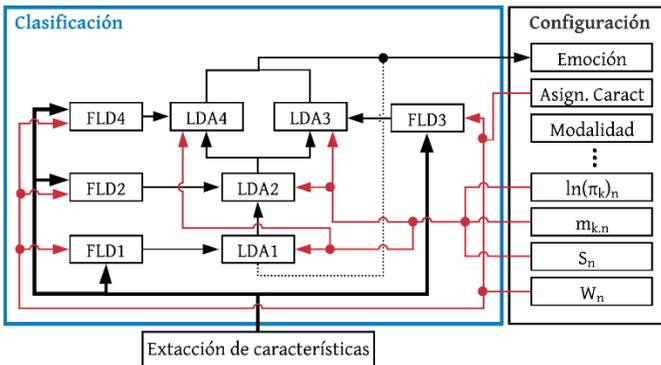


Figura 41. Desglose del subsistema de clasificación

distingue entre el estado neutro (correspondiente a la región central del plano) y no neutro (algún cuadrante del modelo dimensional) por medio de un primer clasificador LDA1. En caso de resultar un estado no neutro, se realiza una clasificación binaria del grado de excitación (baja o alta) a través de un clasificador LDA2 para que, finalmente, se realice una clasificación del nivel de valencia (negativa o positiva) con un clasificador LDA3/LDA4 (la elección depende del resultado de LDA2).

Para realizar estas clasificaciones es necesario que cada clasificador LDAn cuente con los valores  $m_{k,n}$ ,  $\ln(\pi_k)_n$  y  $S_n$  respectivos a cada clase y etapa del esquema EMDC, de igual forma es necesario aplicar una reducción diferente por parte de FLD para cada clasificación binaria realizada en el esquema, lo que comprende generar una matriz de transformación  $W_n$  por cada LDAn. Esto tiene explicación dada la naturaleza del reductor, pues si bien su principal función es proveer de un vector de características reducido, éste necesita la información de las clases comprendidas en cada clasificación para separarlas debidamente, por lo que no es posible aplicarlo para el esquema por completo pues se comprenden distintas clases en cada nivel. De igual forma cada entrada de los algoritmos FLDn corresponde a una selección diferente del algoritmo SFFS, pues se busca encontrar aquellas características que maximicen la clasificación dependiendo del rasgo que se quiere identificar en cada etapa del EMDC (neutralidad, excitación o valencia).

Por último, una vez obtenida la región del modelo dimensional, se registra la emoción detectada en la variable “Emoción”, la cual es enviada por medio del subsistema de comunicaciones a una computadora externa en la modalidad de *prueba*.

En cuanto a los requerimientos del subsistema, el subsistema de clasificación lee y escribe los datos de forma paralela en todo momento, por lo que no es necesario considerar algún tipo de comunicación serial. Respecto al tiempo de ejecución, no existe alguna limitación más allá de mantener una duración tal que permita una respuesta del sistema acorde a los objetivos de la propuesta. A continuación, se muestra un resumen de los requerimientos en relación con el tipo de dato y memoria necesarios para la comunicación de los bloques internos del subsistema, así como sus entradas y salidas.

- 1) **Extracción de características:** (22 valores de 32 bits) Entrada de los bloques FLDn. Comprende el valor de las 22 características extraídas por medio del subsistema de extracción de características. Éstas deben estar en un formato de punto flotante.
- 2) **FLDn:** (3 variables de longitud mixta) Bloque encargado de la reducción dimensional de los vectores de características en cada etapa del esquema EMDC. A continuación, se describen los requerimientos para cada una de sus entradas y salidas:
  - a. **LDA<sub>n</sub>** (32 bits): La salida de cada bloque FLDn es la reducción del vector de características seleccionadas por SFFSn. Este valor debe estar en un formato de punto flotante.
  - b. *Subsistema de configuración:* 2 variables de entrada respectivas a “Asignación de características” y “Matrices de transformación W” cuyas longitudes corresponden a aquellas descritas en la sección 4.3.1.
- 3) **LDA<sub>n</sub>:** (4 variables de longitud mixta) Bloque encargado de la clasificación de las características. A continuación, se describen los requerimientos para cada una de sus entradas y salidas:
  - a. **LDA<sub>k</sub>**: (1 bit) La salida de cada bloque LDA<sub>n</sub> indica la pertenencia a una u otra clase dentro de cada clasificación del esquema EMDC y determina el uso del bloque LDA<sub>k</sub> siguiente.
  - b. *Subsistema de configuración:* 3 variables respectivas a “Vectores de valores a priori  $\ln(\pi_k)_n$ ”, “Vector de medias  $m_{k,n}$ ” y “Matrices de covarianza  $S_n$ ” cuyas longitudes corresponden a aquellas descritas en la sección 4.3.1.
- 4) **Emoción:** (3 bits) Valor codificado que indica una de las cinco regiones contempladas del modelo dimensional.

La razón por la que se necesitan algunos valores en formato de punto flotante es que se desconoce qué tan grandes o pequeños podrían resultar, lo que complicaría su representación y aritmética con un formato de punto fijo a un determinado  $Q_i$ . Por el contrario, el punto flotante otorga un intervalo dinámico mucho más amplio para representar los valores y una mayor flexibilidad a la hora de operarlos. Para ello es necesario que éstos sean almacenados en al menos 32 bits para garantizar un cierto grado de precisión y, además, es conveniente contar con una unidad de punto flotante que permita realizar su aritmética con soltura.

#### 4.3.6 Sistema en la computadora

Una parte fundamental del detector de emociones es el sistema desarrollado en la computadora, pues éste permite almacenar y plasmar información, configurar el dispositivo y encargarse de tareas que ameritan una alta carga computacional.

En primer lugar, es necesario explicar las conexiones entre los subsistemas del sistema, ilustradas en la Figura 42, las cuales se encargan de ejecutar las funciones antes descritas. La forma en que se realiza la comunicación del prototipo con la aplicación es por medio de radiofrecuencia (RF), el cual proporciona una serie de datos a la computadora cuya naturaleza depende de la modalidad en que se encuentre el dispositivo. La interfaz gráfica de usuario (GUI, por sus siglas en inglés) trabaja en conjunto con los subsistemas de aprendizaje automático (SFFS, FLD y LDA) únicamente en la modalidad de *entrenamiento*. Dichos subsistemas están conectados en serie y pueden describirse de la siguiente manera: SFFS, encargado de la selección de características y generación de nuevos espacios; FLD, cuyo propósito es la creación de matrices de transformación para la comprensión de datos y, finalmente, LDA, encargado de generar los parámetros necesarios para cada clasificador LDA utilizado en el detector de emociones. Finalmente, los datos generados en la aplicación son entregados al detector utilizando el subsistema de Comunicación RF, esto se realiza al presionar el botón correspondiente en la GUI.

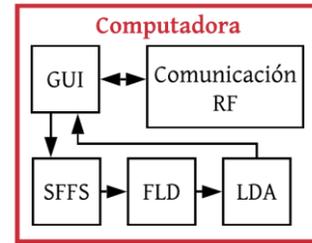


Figura 42. Sistema en la computadora

Dichos subsistemas están conectados en serie y pueden describirse de la siguiente manera: SFFS, encargado de la selección de características y generación de nuevos espacios; FLD, cuyo propósito es la creación de matrices de transformación para la comprensión de datos y, finalmente, LDA, encargado de generar los parámetros necesarios para cada clasificador LDA utilizado en el detector de emociones. Finalmente, los datos generados en la aplicación son entregados al detector utilizando el subsistema de Comunicación RF, esto se realiza al presionar el botón correspondiente en la GUI.

Ya especificadas las conexiones de los subsistemas, es necesario ahondar en el funcionamiento general de este sistema dependiendo de la modalidad, pues ésta definirá las entradas y salidas requeridas en el mismo.

- En la modalidad de *entrenamiento* se reciben los vectores con 22 características guardadas en la variable “Valor de características” especificada en la sección 4.3.1 y, posteriormente, se hace un registro de las regiones del modelo dimensional a las que están ligados dichos vectores con ayuda de los resultados del esquema SAM, mismas que serán almacenados en dos archivos, respectivamente. Una vez almacenada la información, se puede realizar la selección de características por medio del algoritmo SFFS y, finalmente, generar los parámetros de los reductores FLDn y los clasificadores LDA. Esto se realiza por medio de un entrenamiento que se ejecuta en la misma aplicación. Los parámetros generados son enviados de vuelta por medio del subsistema de Comunicación RF.
- En la modalidad de *prueba* se reciben, de manera constante, las emociones detectadas en el prototipo por medio de su módulo RF. Una vez detectada la emoción, ésta es plasmada en una gráfica de barras con una duración máxima de 24h.

A continuación, se hace una descripción más detallada de cada subsistema plasmado en la Figura 42 con el objetivo de conocer su funcionamiento interno y la forma en que da resolución a la tarea encomendada.

#### 4.3.6.1 Interfaz gráfica de usuario

La finalidad de este subsistema es brindar una visualización estructurada de los campos de selección, los cuales permiten el registro y envío de datos que definen los parámetros y funcionalidad del sistema. A continuación, se enlistan los elementos que, con el objetivo de cumplir con los requerimientos planteados, son integrados en la interfaz gráfica.

- **Entablar conexión:** elemento que comienza la transmisión de datos entre la computadora y el detector de emociones mediante el subsistema de comunicación RF.
- **Inicio de medición:** indica al detector de emociones el comienzo de la adquisición de las señales fisiológicas a partir del sensor biométrico.
- **Sección de configuración:** bloque dirigido a la selección de la modalidad del sistema, además de activar/desactivar el dispositivo optoelectrónico.

- **Sección de entrenamiento:** conjunto de elementos enfocados a la reducción de características y entrenamiento del sistema, así como del envío de los parámetros al detector de emociones.
- **Autoevaluación:** interacción del usuario con el esquema SAM, en el cual se registran los niveles de excitación y valencia que presentó durante el estímulo.
- **Visualización de resultados:** elemento que permite interpretar con facilidad la ocurrencia de las emociones suscitadas en un intervalo de tiempo definido.
- **Cuadros de texto:** notas de texto presentes en la GUI que auxilian al usuario en el uso de la misma.

#### 4.3.6.2 Comunicación RF

Mediante este subsistema se realiza la comunicación por RF con el detector de emociones a fin de permitir la configuración de éste a través de las variables descritas en el subsistema de Configuración. En éste se reciben tanto las características necesarias para entrenar al clasificador LDA como las emociones que se encuentre clasificando el sistema. Su operación es asíncrona dado que la transmisión de datos requiere de oprimir los botones desplegados en la GUI, y, por otro lado, la recepción requiere esperar que el detector mande la información deseada, lo cual ocurre de forma no periódica.

Cada uno de los botones de las ventanas de configuración y entrenamiento dentro de la GUI se encuentra relacionado a una variable que a la vez se asocia a una dirección preestablecida dentro del mapa de registros del detector de emociones. Por esta razón, la comunicación de datos con el detector sigue la estructura ilustrada en la Figura 43 inciso A, que parte del envío del *nombre* del sensor, seguido de la *dirección* del registro, al cual se desea escribir o leer un dato, y finalmente se escribe o lee el *dato* deseado. No obstante, se busca utilizar un bit del campo de *dirección* para indicar el tipo de mensaje, ya sea de escritura o lectura de datos, así como otro bit en el mismo campo para indicar la continuidad de mensajes sin necesidad de enviar nuevamente el *nombre* del sensor, como se ilustra en la Figura 43 inciso B.

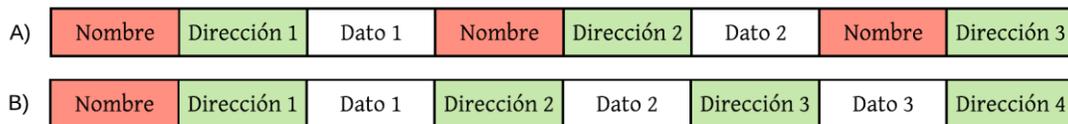


Figura 43. Estructura de trama de comunicación:  
A) Sin bit de continuidad B) Con bit de continuidad

Asimismo, las entradas y salidas a este sistema requieren de las mismas longitudes de palabra descritas en la sección 4.3.1 para el subsistema de configuración en el sistema embebido.

#### 4.3.6.3 Selección secuencial hacia delante flotante

El subsistema SFFS únicamente funciona durante la etapa de *entrenamiento* del sistema. Su operación resulta asíncrona, pues éste es activado tras seleccionar el tipo de entrenamiento y posteriormente presionar el botón “Entrenar” de la GUI. Cabe resaltar que la clase de entrenamiento (*rápido* o *completo*) depende del tipo de validación utilizada dentro del algoritmo SFFS (véase la sección 3.3.5). Este último aspecto se verá con mayor énfasis en el Diseño de detalle.

El seleccionador hace uso de la información enviada por parte del detector de emociones para el entrenamiento. Para ello será necesario almacenar la información en dos archivos, los cuales corresponderán a los vectores de características y sus respectivas clases para cada etapa del EMDC, tal como se describió en la sección 4.3.6.1. SFFS se utiliza en su modalidad de envoltorio, por lo que el algoritmo crea momentáneamente subespacios de características que contengan elementos del espacio original y, posteriormente, se evalúan a través de un clasificador, tal como se explicó en la sección 3.4.2. Para ello se utiliza el clasificador LDA con el objetivo de identificar aquellas características

que maximicen la exactitud de la clasificación en cada etapa del esquema EMDC, siendo posible identificar y desechar directamente aquellas que no aportan información decisiva para identificar el rasgo deseado en la persona durante cada nivel (neutralidad, excitación o valencia).

Tal como se muestra en la Figura 44, inicialmente se realiza una selección por medio de una SFFS1 considerando únicamente las clases neutro y no neutro (A), posteriormente, se crea un subespacio desechando aquellos vectores pertenecientes a la clase neutro del espacio original y se realiza con él una selección con SFFS2 enfocándose solo en la clase excitación (B), finalmente, se hacen dos últimos subespacios para la selección con SFFS3 y SFFS4 en el nivel de valencia que contempla desechar aquellos vectores pertenecientes a la clase de excitación baja o excitación alta dependiendo del espacio reducido que se quiera obtener (C y D, respectivamente).

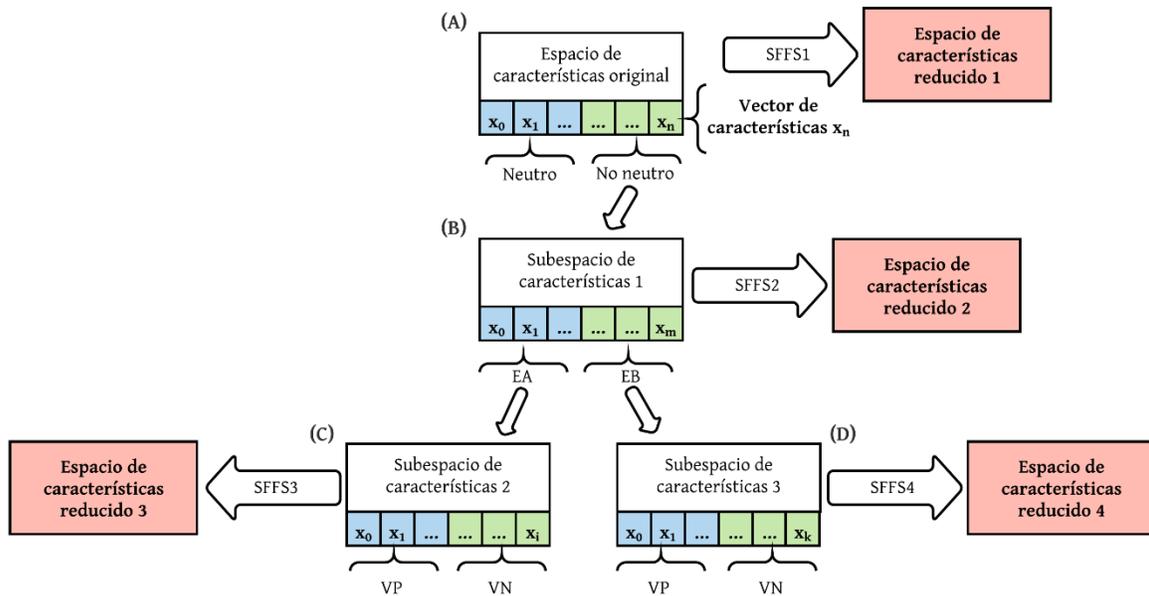


Figura 44. Metodología de la aplicación de SFFS para el esquema EMDC.

Al terminar el proceso se obtienen cuatro espacios de características reducidos que serán utilizados para el entrenamiento de cada FLDn y LDAn del EMDC. De igual forma, este algoritmo proporciona una codificación de las características seleccionadas durante cada etapa del esquema en 4 variables de 22 bits para enviarlas por medio del subsistema de Comunicación RF y almacenarlas en la variable “Asignación de características” del prototipo.

#### 4.3.6.4 Discriminante lineal de Fisher

Una vez obtenidos los nuevos conjuntos de entrenamiento, se ejecuta el algoritmo FLD considerando cada etapa del esquema EMDC, de modo que se utilizan los espacios generados anteriormente mediante el subsistema SFFS. De esta forma se obtienen un total de cuatro matrices de transformación  $W$  que corresponden a cada clasificación binaria realizada en el esquema, tal como se ilustra en la Figura 41, lo cual permitirá comprimir los vectores de características y separar adecuadamente las clases contempladas en cada etapa. Cada elemento de las matrices  $W_n$  debe ser codificado para hacer posible su envío por medio del subsistema de Comunicación RF para la reducción de los vectores de características (previamente seleccionadas por cada SFFSn) durante la modalidad de *prueba* del detector, esta información quedará almacenada en la variable “Matriz de transformación”. De igual forma serán utilizadas para comprimir los espacios de características que se necesitan para el entrenamiento del LDA.

#### 4.3.6.5 Análisis discriminante lineal

Tras comprimir los espacios de características seleccionados por el subsistema SFFS mediante las matrices de transformación  $W_n$ , se realiza el entrenamiento de cada algoritmo de clasificación LDAn, resultando en un total de cuatro entrenamientos. Como consecuencia se obtienen una serie de parámetros  $m_{k,n}$ ,  $\ln(\pi_k)_n$  y  $S_n$  correspondientes a cada clase y clasificación realizada en el esquema. Dichos parámetros deben ser codificados para su envío al dispositivo por medio del subsistema de Comunicación RF, mismos que serán asignados a las variables de “Vector de valores a priori”, “Vector de medias” y “Matriz de covarianza”, respectivamente.

### 4.4 Diseño de detalle

Una vez establecida la arquitectura, diseño geométrico, los requerimientos y las especificaciones funcionales de cada subsistema, es necesario puntualizar los detalles de cada uno de éstos, como lo puede ser el hardware seleccionado, las particularidades de cada algoritmo y protocolo, etc. Esto tiene como objetivo establecer una visión clara del prototipo final y generar las bases necesarias para su implementación. A continuación, se describe el diseño de detalle de cada subsistema presentado en la sección 4.3.

#### 4.4.1 Configuración

El mapa de registros mencionado en la sección 4.3.1 para alojar todas las variables del subsistema de configuración se estructura en direcciones y contenido de 8 bits con el objetivo de dar versatilidad al detector para su uso con algunos de los protocolos de comunicación seriales más utilizados en sensores, como son el transmisor-receptor universal asíncrono (UART, por sus siglas en inglés), la interfaz de periféricos serie (SPI, por sus siglas en inglés) y el circuito inter-integrado (I2C), sin embargo, este trabajo hará uso exclusivo del protocolo UART para este fin. Queda a disposición del lector la adaptación del código proporcionado en el repositorio de la sección 5 para soportar el resto de los protocolos mencionados.

El mapa de registros de la Tabla 13 describe el nombre y dirección respectivo a cada uno de los registros, así como la asignación que se da a cada uno de sus bits. Cabe mencionar que al final de la Tabla 13 se hace la descripción de los dos bits más significativos del campo de dirección reservados para cumplir las funcionalidades especiales del tipo y continuidad del mensaje mencionadas en la sección 4.3.6.2. En la Tabla 14 se realiza un desglose de los registros de la Tabla 13 con la descripción a detalle de los bits que alberga, describiendo las funcionalidades, opciones y limitaciones que pueden presentar, así como el tipo de acceso que poseen, ya sea de carácter lectura/escritura o puramente de lectura, y el valor que poseen de forma predeterminada al arrancar por primera vez el sistema.

Contemplando los requerimientos mencionados para los subsistemas de configuración y clasificación, se observó la necesidad de utilizar múltiples elementos para las variables de asignación de características, las matrices de transformación y de covarianza inversa, y los vectores de medias y de valores a priori, por lo que se opta por reservar algunos espacios en el mapa para fungir como registros de configuración. La funcionalidad de éstos será multiplexar el contenido a escribir o leer de los diversos elementos de las variables mencionadas. Asimismo, se anexa a cada uno de estos registros un bit denominado *Auto* que permite realizar una escritura continua de las variables que involucran a un determinado parámetro, no obstante, esta funcionalidad resulta exclusiva para la comunicación con el protocolo UART. De esta forma se logra reducir el tamaño del mapa en la memoria del sistema, además de simplificar la configuración del dispositivo.

A su vez, se incluye en el primer registro dos variables que permitan habilitar el proceso de calibración del sensor (Calib) y la transmisión automática de las características estimadas por el detector (AutoTx) exclusivamente por medio de UART con el módulo BLE.

Tabla 13. Mapa de registros del detector de emociones

DIR	Nombre	B7	B6	B5	B4	B3	B2	B1	B0
0x00	Config	Reserv	Reserv	Reserv	AutoTx*	Medic	Calib	Modo	RGB
0x01	CC Config	Reserv	Reserv	Reserv	Reserv	Reserv	Auto*	Sel CC	
0x02	Caract PRV 1	VLF	LF/HF	SD1/SD2	RMSSD	SDNN	PPImed	NN50	pNN50
0x03	Caract PRV 2	Reserv	Reserv	Reserv	Reserv	SD2	SD1	HF	LF
0x04	Caract EDA 1	ctl90	ctl75	ctl50	ctl25	sdAmpl	mTL	AmpSCR	NumSCR
0x05	Caract EDA 2	Reserv	Reserv	Reserv	Reserv	Reserv	Reserv	Reserv	HF
0x06	$W_n$ Config	Auto*	$W_n$		Elemento Mtz				
0x07	Matriz FLDn $W_n$	Parte fracc D [7:0]							
0x08		Parte fracc C [15:8]							
0x09		Parte fracc B [24:16]							
0x0A		Parte fracc A [27:25]/ Parte entera [4:0]							
0x0B	Vect Config	Reserv	Auto*	Sel Vect		Elemento Vect			
0x0C	Vector $m_{k,n}/$ $\ln(\pi_k)_n/S_n$	Parte fracc D [7:0]							
0x0D		Parte fracc C [15:8]							
0x0E		Parte fracc B [24:16]							
0x0F		Parte fracc A [27:25]/ Parte entera [4:0]							
0x10	Emoción	Reserv	Reserv	Reserv	Reserv	Reserv	Emoción		
0x11	Caract Config	Reserv	Reserv	Auto*	Sel Caract				
0x12	Valor Caract	Parte fracc B [7:0]							
0x13		Parte fracc A [15:8]							
0x14		Parte entera B [7:0]							
0x15		Parte entera A [15:8]							
1XXX XXXXb*		Este bit indica el <b>tipo</b> de mensaje (0) Escritura (1) Lectura							
X1XX XXXXb*		Este bit en 1 habilita la <b>continuidad</b> de mensajes sin el reenvío del campo de nombre							

\*Sólo aplica para comunicaciones con protocolo UART

Tabla 14. Desglose de registros ilustrados en el mapa de la Tabla 13.

DIR: 0x00		Config							
Bit	Nombre	Predet	Acceso	Descripción					
4	AutoTx*	0	RW	0 ... Deshabilita la transmisión automática de las variables de Emoción, Caract Config y Valor Caract. 1 ... Habilita la transmisión automática de las variables de Emoción, Caract Config y Valor Caract, considerando ambos campos de <i>dirección</i> y <i>dato</i> , de acuerdo con el respectivo modo de operación del detector. <i>Entrenamiento</i> Caract Config y Valor Caract					

<i>Prueba Emoción</i>				
3	Medic	0	RW	0 ... Deshabilita el funcionamiento del sensor biométrico 1 ... Habilita el funcionamiento del sensor biométrico
2	Calib	0	RW	0 ... Deshabilita la calibración de la señal PPG en el sensor biométrico 1 ... Habilita la calibración de la señal PPG en el sensor biométrico. Este bit debe ir acompañado del bit Medic en 1 y se limpia a 0 en cuanto se termina la calibración de la señal.
1	Modalidad	0	RW	0 ... Modo de entrenamiento 1 ... Modo de prueba
0	RGB	1	RW	0 ... Deshabilita el funcionamiento del LED RGB 1 ... Habilita el funcionamiento del LED RGB <u>NOTA:</u> No aplica durante la calibración del sensor biométrico

<b>DIR: 0x01</b>		<b>CC Config</b>		
Bit	Nombre	Predet	Acceso	Descripción
2	Auto*	0	RW	0 ... La escritura de cada byte de las variables Caract PRVn y Caract EDAn debe seguirse del envío de su respectivo campo de <i>dirección</i> (0x02h-0x05h). 1 ... Permite la escritura consecutiva de los cuatro bytes de las variables Caract PRVn y Caract EDAn sin el envío de sus campos de <i>dirección</i> (0x02h-0x05h). Los datos se leen/escriben posterior a la escritura de esta variable CC Config y parten de 0x02 hasta 0x05, tras la acción sobre el último registro 0x05 se limpia este bit a 0.
1:0	Sel CC	0	RW	Selecciona el conjunto de características al que se escribirá o leerá en los 4 bytes de Caract PRVn y Caract EDAn: 0 ... Conjunto de características para LDA1 1 ... Conjunto de características para LDA2 2 ... Conjunto de características para LDA3 3 ... Conjunto de características para LDA4

<b>DIR: 0x02</b>		<b>Caract PRV 1</b>		
Bit	Nombre	Predet	Acceso	Descripción
7	VLF	1	RW	0 ... Ignora la estimación del VLF en la PRV 1 ... Realiza la estimación del VLF en la PRV
6	LF/HF	1	RW	0 ... Ignora la estimación del LF/HF en la PRV 1 ... Realiza la estimación del LF/HF en la PRV
5	SD1/SD2	1	RW	0 ... Ignora la estimación del SD1/SD2 en la PRV 1 ... Realiza la estimación del SD1/SD2 en la PRV
4	RMSSD	1	RW	0 ... Ignora la estimación del RMSSD en la PRV 1 ... Realiza la estimación del RMSSD en la PRV

3	SDNN	1	RW	0 ... Ignora la estimación del SDNN en la PRV 1 ... Realiza la estimación del SDNN en la PRV
2	PPImed	1	RW	0 ... Ignora la estimación del PPImed en la PRV 1 ... Realiza la estimación del PPImed en la PRV
1	NN50	1	RW	0 ... Ignora la estimación del NN50 en la PRV 1 ... Realiza la estimación del NN50 en la PRV
0	pNN50	1	RW	0 ... Ignora la estimación del pNN50 en la PRV 1 ... Realiza la estimación del pNN50 en la PRV

DIR: 0x03		Caract PRV 2		
Bit	Nombre	Predet	Acceso	Descripción
3	LF	1	RW	0 ... Ignora la estimación del LF en la PRV 1 ... Realiza la estimación del LF en la PRV
2	HF	1	RW	0 ... Ignora la estimación del HF en la PRV 1 ... Realiza la estimación del HF en la PRV
1	SD1	1	RW	0 ... Ignora la estimación del SD1 en la PRV 1 ... Realiza la estimación del SD1 en la PRV
0	SD2	1	RW	0 ... Ignora la estimación del SD1 en la PRV 1 ... Realiza la estimación del SD2 en la PRV

DIR: 0x04		Caract EDA 1		
Bit	Nombre	Predet	Acceso	Descripción
7	ctl90	1	RW	0 ... Ignora la estimación del ctl90 en la EDA 1 ... Realiza la estimación del ctl90 en la EDA
6	ctl75	1	RW	0 ... Ignora la estimación del ctl75 en la EDA 1 ... Realiza la estimación del ctl75 en la EDA
5	ctl50	1	RW	0 ... Ignora la estimación del ctl50 en la EDA 1 ... Realiza la estimación del ctl50 en la EDA
4	ctl25	1	RW	0 ... Ignora la estimación del ctl25 en la EDA 1 ... Realiza la estimación del ctl25 en la EDA
3	sdAmp	1	RW	0 ... Ignora la estimación del sdAmp en la EDA 1 ... Realiza la estimación del sdAmp en la EDA
2	mTL	1	RW	0 ... Ignora la estimación del mTL en la EDA 1 ... Realiza la estimación del mTL en la EDA
1	AmpSCR	1	RW	0 ... Ignora la estimación del AmpSCR en la EDA 1 ... Realiza la estimación del AmpSCR en la EDA
0	NumSCR	1	RW	0 ... Ignora la estimación del NumSCR en la EDA 1 ... Realiza la estimación del NumSCR en la EDA

DIR: 0x05		Caract EDA 2		
Bit	Nombre	Predet	Acceso	Descripción
0	HF	1	RW	0 ... Ignora la estimación del HF en la EDA

				1 ... Realiza la estimación del HF en la EDA
--	--	--	--	--

DIR: 0x06		W <sub>n</sub> Config		
Bit	Nombre	Predet	Acceso	Descripción
7	Auto*	0	RW	0 ... La escritura de cada byte de Matriz FLDn W <sub>n</sub> debe seguirse del envío de su respectivo campo de <i>dirección</i> (0x07h-0x0Ah). 1 ... Permite la escritura consecutiva de los cuatro bytes de Matriz FLDn W <sub>n</sub> sin el envío de sus campos de <i>dirección</i> (0x07h-0x0Ah). Los datos se leen/escriben posterior a la escritura de esta variable W <sub>n</sub> Config y parten de 0x07 hasta 0x0A, tras la acción sobre el último registro 0x0A se limpia este bit a 0.
6:5	W <sub>n</sub>	0	RW	Selecciona la matriz de transformación W <sub>n</sub> al que se escribirá o leerá el dato en los 4 bytes de Matriz FLDn W <sub>n</sub> : 0 ... Matriz de transformación W <sub>1</sub> para el FLD1 1 ... Matriz de transformación W <sub>2</sub> para el FLD2 2 ... Matriz de transformación W <sub>3</sub> para el FLD3 3 ... Matriz de transformación W <sub>4</sub> para el FLD4
4:0	Elemento Mtz	0	RW	Selecciona el elemento de la matriz de transformación W <sub>n</sub> al que se escribirá o leerá el dato en los 4 bytes de Matriz FLD W <sub>n</sub> : 00h ... Elemento 1 de Matriz FLDn W <sub>n</sub> 01h ... Elemento 2 de Matriz FLDn W <sub>n</sub> 14h ... Elemento 21 de Matriz FLDn W <sub>n</sub> 15h ... Elemento 22 de Matriz FLDn W <sub>n</sub> 16h-1Fh ... No utilizar

DIR: 0x07		Matriz FLDn W <sub>n</sub>		
Bit	Nombre	Predet	Acceso	Descripción
7:0	Parte fracc D [7:0]	0	RW	Alberga ocho bits de la parte fraccionaria del elemento de la matriz W <sub>n</sub> expresado en punto fijo Q28.

DIR: 0x08		Matriz FLDn W <sub>n</sub>		
Bit	Nombre	Predet	Acceso	Descripción
7:0	Parte fracc C [15:8]	0	RW	Alberga ocho bits de la parte fraccionaria del elemento de la matriz W <sub>n</sub> expresado en punto fijo Q28.

DIR: 0x09		Matriz FLDn W <sub>n</sub>		
Bit	Nombre	Predet	Acceso	Descripción
7:0	Parte fracc B [24:16]	0	RW	Alberga ocho bits de la parte fraccionaria del elemento de la matriz W <sub>n</sub> expresado en punto fijo Q28.

DIR: 0x0A		Matriz FLDn $W_n$		
Bit	Nombre	Predet	Acceso	Descripción
7:0	Parte fracc A [27:25]/ Parte entera [4:0]	0	RW	Alberga los primeros tres bits de la parte fraccionaria del elemento de la matriz $W_n$ expresado en punto fijo Q28, junto con dos bits de su parte entera y un bit de signo.

DIR: 0x0B		Vect Config		
Bit	Nombre	Predet	Acceso	Descripción
6	Auto*	0	RW	0 ... La escritura de cada byte de Vector $m_k/\pi_k/S$ debe seguirse del envío de su respectivo campo de <i>dirección</i> (0x0Ch-0x0Fh). 1 ... Permite la escritura consecutiva de los cuatro bytes de Vector $m_k/\pi_k/S$ sin el envío de sus campos de <i>dirección</i> (0x0Ch-0x0Fh). Los datos se leen/escriben posterior a la escritura de esta variable Vect Config y parten de 0x0C hasta 0x0F, tras la acción sobre el último registro 0x0F se limpia este bit a 0.
5:4	Sel Vect	0	RW	Selecciona el vector al que se escribirá o leerá el dato en los 4 bytes de Vector $m_k/\pi_k/S$ : 0 ... Vector $m_{k,n}$ 1 ... Vector $\ln(\pi_k)_n$ 2 ... Vector $S_n$ 3 ... No utilizar
3:0	Elemento Vect	0	RW	Selecciona el elemento del vector al que se escribirá o leerá el dato en los 4 bytes de Vector $m_{k,n}/\ln(\pi_k)_n/S_n$ : 0 ... Vector $m_{0,0}/\ln(\pi_0)_0$ Clase No neutro y Vector $S_0$ : Clasificación Neutro/No neutro 1 ... Vector $m_{1,0}/\ln(\pi_1)_0$ : Clase Neutro y Vector $S_1$ : Clasificación EB/EA 2 ... Vector $m_{0,1}/\ln(\pi_0)_1$ : Clase EB y Vector $S_2$ : Clasificación EA→VN/VP 3 ... Vector $m_{0,1}/\ln(\pi_1)_1$ : Clase EA y Vector $S_3$ : Clasificación EB→VN/VP 4 ... Vector $m_{0,2}/\ln(\pi_0)_2$ : Clase EA-VN 5 ... Vector $m_{1,2}/\ln(\pi_1)_2$ : Clase EA-VP 6 ... Vector $m_{0,3}/\ln(\pi_0)_3$ : Clase EB-VN 7 ... Vector $m_{1,3}/\ln(\pi_1)_3$ : Clase EB-VP

DIR: 0x0C		Vector $m_k/\pi_k/S$		
Bit	Nombre	Predet	Acceso	Descripción
7:0	Parte fracc D [7:0]	0	RW	Alberga ocho bits de la parte fraccionaria del elemento del vector $m_k/\pi_k/S$ expresado en punto fijo Q28.

DIR: 0x0D		Vector $m_k/\pi_k/S$		
Bit	Nombre	Predet	Acceso	Descripción
7:0	Parte fracc C [15:8]	0	RW	Alberga ocho bits de la parte fraccionaria del elemento del vector $m_k/\pi_k/S$ expresado en punto fijo Q28.

DIR: 0x0E		Vector $m_k/\pi_k/S$		
Bit	Nombre	Predet	Acceso	Descripción
7:0	Parte fracc B [24:16]	0	RW	Alberga ocho bits de la parte fraccionaria del elemento del vector $m_k/\pi_k/S$ expresado en punto fijo Q28.

DIR: 0x0F		Vector $m_k/\pi_k/S$		
Bit	Nombre	Predet	Acceso	Descripción
7:0	Parte fracc A [27:25]/ Parte entera [4:0]	0	RW	Alberga los primeros tres bits de la parte fraccionaria del elemento del vector $m_k/\pi_k/S$ expresado en punto fijo Q28, junto con dos bits de su parte entera y un bit de signo.

DIR: 0x10		Emoción		
Bit	Nombre	Predet	Acceso	Descripción
2:0	Emoción	0	RO	Emoción clasificada por el sistema: 0 ... Sin asignación de emoción 1 ... Excitación baja – Valencia negativa (EB-VN) 2 ... Excitación baja – Valencia positiva (EB-VP) 3 ... No utilizar 4 ... Excitación alta – Valencia negativa (EA-VN) 5 ... No utilizar 6 ... Excitación alta – Valencia positiva (EA-VP) 7 ... Neutro

DIR: 0x11		Caract Config		
Bit	Nombre	Predet	Acceso	Descripción
4	Auto*	0	RW	0 ... La lectura de cada byte de Caract debe seguirse del envío de su respectivo campo de <i>dirección</i> (0x12h-0x15h). 1 ... Permite la lectura consecutiva de los cuatro bytes de Caract sin el envío de sus campos de <i>dirección</i> (0x12h-0x15h). Los datos se leen posterior a la escritura de esta variable Caract Config y parten de 0x12 hasta 0x15, tras la acción sobre el último registro 0x15 se limpia este bit a 0.
3:0	Sel Caract	0	RW	Selecciona la característica a la que se escribirá o leerá el dato en los 32 bits de Caract: 00h ... pNN50                    0Ch ... NumSCR 01h ... NN50                    0Dh ... AmpSCR

				02h ... PPImed	0Eh ... mTL
				03h ... SDNN	0Fh ... sdAmpl
				04h ... RMSSD	10h ... ct125
				05h ... SD1/SD2	11h ... ct150
				06h ... LF/HF	12h ... ct175
				07h ... VLF	13h ... ct190
				08h ... PRV_LF	14h ... EDA_HF
				09h ... PRV_HF	15h-1Fh ... No utilizar
				0Ah ... SD1	
				0Bh ... SD2	

DIR: 0x12		Caract			
Bit	Nombre	Predet	Acceso	Descripción	
7:0	Parte fracc B [7:0]	0	RO	Alberga el byte inferior de la parte fraccionaria del valor de la característica seleccionada en Sel Caract expresado en punto fijo Q16.	

DIR: 0x13		Caract			
Bit	Nombre	Predet	Acceso	Descripción	
7:0	Parte fracc A [15:8]	0	RO	Alberga el byte superior de la parte fraccionaria del valor de la característica seleccionada en Sel Caract expresado en punto fijo Q16.	

DIR: 0x14		Caract			
Bit	Nombre	Predet	Acceso	Descripción	
7:0	Parte entera B [7:0]	0	RO	Alberga el byte inferior de la parte entera del valor de la característica seleccionada en Sel Caract.	

DIR: 0x15		Caract			
Bit	Nombre	Predet	Acceso	Descripción	
7:0	Parte entera A [15:8]	0	RO	Alberga el byte superior de la parte entera del valor de la característica seleccionada en Sel Caract.	

\*Sólo aplica para comunicaciones con protocolo UART

#### 4.4.2 Comunicaciones

El dispositivo optoelectrónico con el que se comunica el bloque GPIO se implementa mediante un LED RGB. Esta elección es debido a la gama de colores que éste ofrece y con el cual se abastecen los diferentes estados que se busca emitir. De esta forma, se disminuye el número de puertos requeridos en la sección 4.3.2 de siete puertos a tres puertos correspondientes a los pines R, G y B. De esta forma, la codificación de colores sigue el diagrama de la Figura 45 con un amarillo para EA-VP, verde para EB-VP, rojo para EA-VN, azul para EB-VN, blanco para Neutro, cian para el periodo de calibración del sensor biométrico y violeta para el indicador de error. Para este elemento se asignan resistencias de 82  $\Omega$  a fin de garantizar una corriente de  $\approx 20$  mA a través del LED rojo

	EA-VN
	EB-VP
	EB-VN
	EA-VP
	Calibración
	Error
	Neutro

Figura 45. Codificación de colores

y  $\approx 1.25 \text{ mA}$  en los LEDs azul y verde considerando en  $3.1 \text{ V}$  y  $2 \text{ V}$  sus voltajes de umbral respectivamente.

Para la comunicación inalámbrica se escoge el protocolo Bluetooth de baja energía (BLE, por sus siglas en inglés) debido a que cubre con los requerimientos mencionados en la sección 4.3.2 respecto a costo, consumo y distancia, además de resultar de los protocolos más populares para la interconexión de dispositivos portátiles (computadoras, tabletas, teléfonos inteligentes, entre otros). A su vez este medio de fácil comunicación cumple con la especificación marginal dictada en la sección 4.2.2. Asimismo, se opta por utilizar para su implementación el módulo HM10 puesto que soporta el protocolo anterior y brinda al usuario una configuración amigable mediante el uso del protocolo UART y comandos AT que cubren con las funcionalidades básicas de comunicación que se requieren para el trabajo. Estos comandos corresponden a los descritos en la Tabla 15.

Tabla 15. Comandos AT para configuración del módulo HM10

Comando AT	Descripción	Valor establecido
AT+NAME	Modifica el nombre del módulo.	EmoDet_UNAM
AT+NOTI	Habilita el envío de notificaciones escritas al conectarse y desconectarse el módulo. OK+CONN al conectarse y OK+LOST al desconectarse.	1
AT+BAUD	Fija la tasa de baudios con la que se comunica el protocolo UART del módulo con dispositivos externos.	8 $\equiv$ 115200 bps
AT+POWE	Establece la potencia de transmisión del módulo.	2 $\equiv$ 0 dbm

Para realizar la comunicación con el módulo HM10, el protocolo de comunicación serial UART se configura a una tasa de 115200 bps a fin de garantizar una rápida transmisión y recepción de todos los datos, además de implicar un bajo consumo energético dados los breves ciclos de trabajo que conllevan los estados alto y bajo de los bits comunicados. A su vez, se emplea un tamaño de trama de 8 bits para abastecer el envío de caracteres en formato ASCII que satisfagan el manejo de los comandos AT, así como favorecer la estructura de 8 bits del mapa de registros empleado por el subsistema de configuración. Se omite el uso del bit de paridad dado que el propio protocolo BLE ya cuenta con medios de detección de errores durante la transferencia inalámbrica de los datos. Cabe mencionar que se comienza la configuración del módulo HM10 con una tasa de 9600 bps dadas las instrucciones del fabricante.

Para la adquisición de la señal obtenida mediante PPG se elige una tasa de muestro de 128 Hz ya que resulta una de las mínimas tasas viables para el análisis de las diversas características de PRV conforme a los trabajos en [32] y [33]. En cambio, la tasa de muestreo de la señal EDA se fija a 32 Hz puesto que resulta de las menores tasas de muestreo empleadas en artículos asociados a EDA. Ambas elecciones se toman en cuenta con el fin de reducir la carga computacional del sistema, sin embargo, cabe aclarar que se logró una tasa diferente a la dictada en la sección 4.2.2 dado que estas estaban proyectadas para la señal ECG.

En cuanto al sensor biométrico, se opta por utilizar el modelo AS7026GG de la marca AMS dado que cuenta con los módulos necesarios para recabar la señal PPG y EDA empleando las técnicas escogidas en la sección 4.2.4, así como disponer de los elementos de amplificación y digitalización necesarios para las señales. Una justificación a mayor detalle de la elección de este sensor se dará más adelante en la sección 4.4.2.2.

Dada la naturaleza de operación del sensor, se decide emplear el protocolo de comunicación I2C con una tasa de 400 kbps a fin de procurar un bajo consumo de energía, cuya justificación sigue la misma dada para UART respecto al ciclo de trabajo, y satisfacer la velocidad mínima requerida para cumplir las tasas de muestreo establecidas para PPG y EDA. A su vez, para el puerto GPIO asociado a la notificación recibida del sensor biométrico, se configura una estructura en drenaje abierto a fin de permitir el correcto funcionamiento del protocolo I2C a través de éste. Para las resistencias de pull-up del bus se escogen valores de  $4.7\text{ k}\Omega$  a partir de las gráficas de valores máximo y mínimo brindadas en [73] para una corriente de nivel bajo a  $3.3\text{ mA}$ , una capacitancia del bus de  $70\text{ nF}$  y una alimentación del bus a  $3.3\text{ V}$ .

A partir de la elección de los protocolos mencionados, resulta posible detallar aún más la estructura del subsistema de Comunicaciones de la Figura 38 a un nuevo esquema que contempla la adición de dos nuevos bloques UART e I2C como se muestra en el diagrama de la Figura 46. Dentro de estos bloques, se integran dos algoritmos que realizan la calibración del sensor biométrico y la lectura de los datos del módulo HM10. Su funcionamiento parte de los diagramas de flujo ilustrados en la Figura 47 y Figura 48, respectivamente.

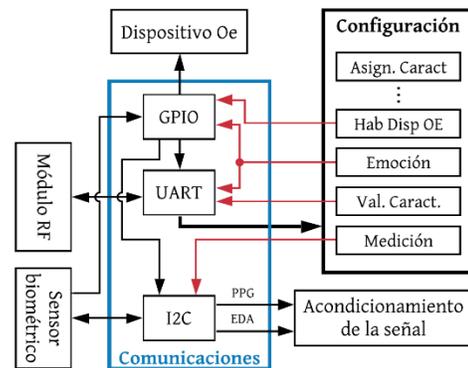


Figura 46. Segundo desglose del subsistema de comunicaciones

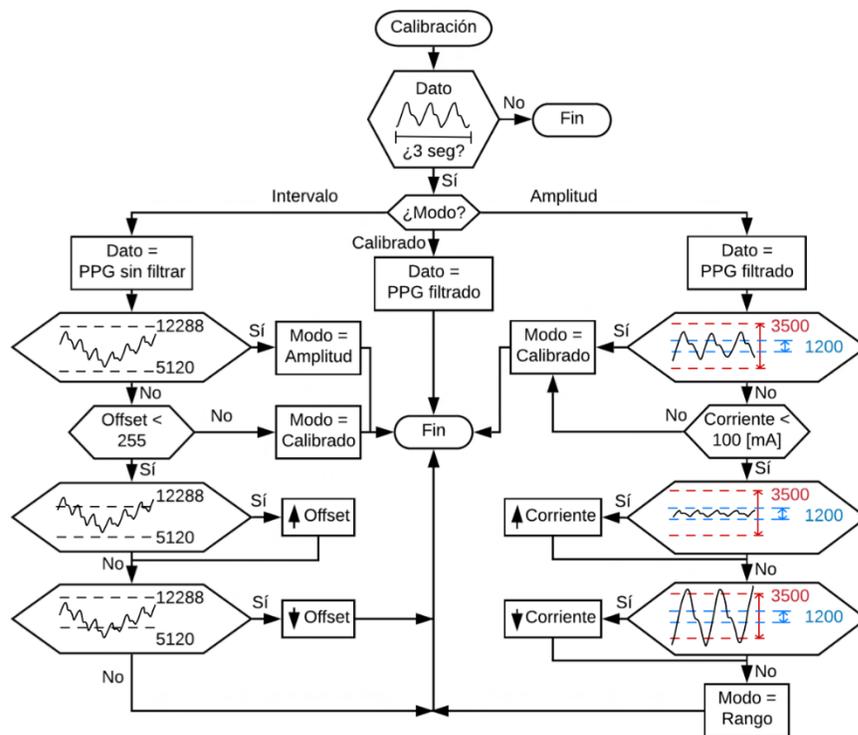


Figura 47. Proceso de calibración del sensor biométrico

(para más información véase la sección 4.4.2.2) en caso de no establecerse la señal sin filtrar (etapa de pre-ganancia) en un intervalo entre 12288 y 5120; esto considerando que el valor máximo que entrega el ADC del sensor AS7026GG es 16383. Otra forma en que se realiza el ajuste es mediante la variación de la corriente suministrada al LED verde del sensor al no acentuarse la amplitud de la señal filtrada (etapa posterior al OFE) entre un valor de 3500 y 1200. Los límites anteriores fueron

El proceso de calibración parte de la necesidad de entregar una señal óptima al usuario, así como manejar la variabilidad de características que se presentan en los individuos que hagan uso del sensor, siendo algunas de ellas el color de la piel, el género, la edad, el entorno, entre otros. Este proceso, ilustrado en la Figura 47, comienza por el análisis de tres segundos de la señal obtenida mediante PPG filtrada o sin filtrar, durante el cual se identifican sus valores máximo y mínimo. Posteriormente, se sigue un proceso de ajuste a través de un offset de corriente

establecidos tras un proceso iterativo de pruebas con el sensor de forma que garantizan la adquisición de la señal a pesar de suscitarse leves movimientos corporales por parte del usuario, así como una adecuada amplitud que permite la correcta adquisición de la PRV. De esta forma, si se cumplen las condiciones anteriores, o si se alcanza el máximo ajuste por corriente o por offset con el que dispone el sensor, la calibración habrá culminado al recabar la señal PPG filtrada.

El proceso de lectura de datos del módulo HM10 ilustrado en la Figura 48 parte de la recepción de información sobre un buffer del cual se identifica la clase de mensaje que se está recibiendo, siendo este un *texto* o un *número*. En caso de ser *texto* se identifica de la cadena de caracteres recibida el carácter de salto de línea, expresado como “\n”, para continuar por identificar el modo actualmente activo del bloque UART. Estos pueden ser en modo *AT* para procurar la correcta configuración del módulo HM10 mediante la escritura de comandos AT, en modo *Conectado* para procurar la conexión del módulo a un dispositivo externo, o en modo *En espera* para identificar la desconexión del módulo HM10 o bien aguardar la escritura del nombre del dispositivo “EM\_DET\_UNAM” para pasar a la lectura de datos numéricos mediante el cambio de la clase de recepción a *número*. De esta forma, para los mensajes cuya clase sea un *número* se procura identificar del primer Byte recibido el alza del bit de “tipo de mensaje” en el campo de dirección y el bit de “continuidad”, ambos descritos en la sección 4.3.6.2. A su vez, se verifica la variable *Auto* que se habilita mediante alguno de los bits con mismo nombre en la Tabla 13. Finalmente, de acuerdo con los estados de los bits anteriores, se ingresa al mapa de registros donde se reciben como parámetros la dirección y el dato y se efectúa una lectura o escritura de alguno de los registros de la Tabla 13.

Para ambos protocolos de comunicación se emplean colas primero en entrar-primero en salir (FIFO, por sus siglas en inglés) para la transmisión y recepción de datos. Éstas cuentan con interrupciones al alcanzarse un nivel de profundidad (>1 Byte) en la FIFO de recepción en el UART y dos niveles de profundidad (>2 Bytes) para la FIFO de recepción en I2C, a fin de evitar la pérdida de datos por cualquier motivo. De esta manera, se busca evaluar los datos recibidos tan pronto ingresen al dispositivo a fin de otorgarle mayor tiempo de operación al procesador, que en caso contrario ocasionaría un retraso dada la espera que implicaría la captación de todo el conjunto de datos y su respectiva evaluación. Asimismo, para cada protocolo se elabora una biblioteca de funciones que

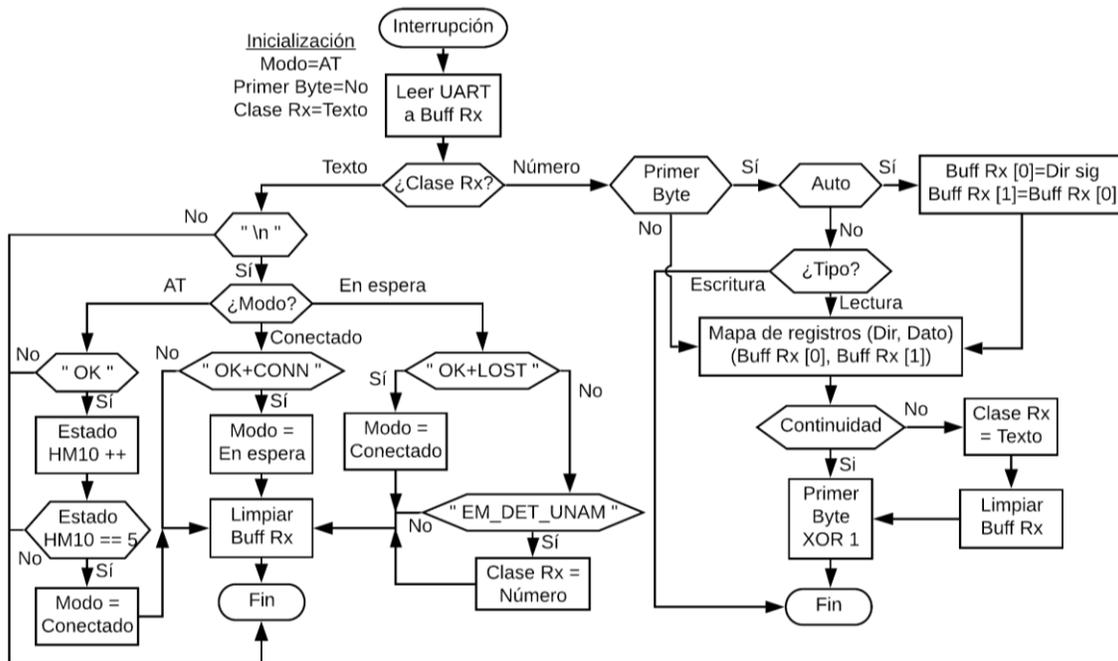


Figura 48. Lectura de los datos del módulo HM10

permitan utilizar de forma sencilla sus módulos respectivos dentro de la tarjeta a elegir, haciendo posible la escritura y lectura de los datos de acuerdo con el registro que le corresponda en memoria.

#### 4.4.2.1 Módulo HM10

El módulo HM10, basado en el chip de Texas Instruments CC2540 con soporte al protocolo BLE, cuenta con un voltaje de operación entre los 3.3-5 V y una corriente de operación de 8.5 mA. Éste dispone de un proceso de configuración inicial, descrito en la Figura 49, el cual permite adaptarlo a las condiciones deseadas para este trabajo. Dicho procedimiento se realiza a través de los valores y comandos AT especificados en la Tabla 15.

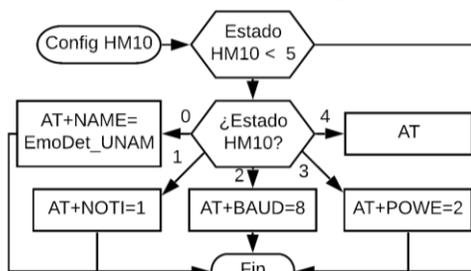


Figura 49. Configuración del módulo HM10.

#### 4.4.2.2 Sensor biométrico AS7026GG

El sensor biométrico AS7026GG de la marca AMS es un módulo que utiliza las técnicas de ECG y PPG para la obtención de señales eléctricas que describen la actividad cardiaca. Asimismo, cuenta con interfaces externas que permiten la adquisición de otras señales fisiológicas, como lo son la temperatura de la piel y la EDA. Dichas señales y técnicas se describen con mayor detalle en la sección 3.1.

El módulo cuenta con diversos componentes electrónicos que hacen posible la obtención de las señales antes descritas. A continuación, se hace un listado de éstos con el objetivo de dar un panorama general de su hardware.

- Interfaz óptica (OFE, por sus siglas en inglés)
  - Cuatro LEDs: dos verdes, uno infrarrojo y uno libre
  - Seis fotodiodos: cuatro con filtros verdes, uno infrarrojo y uno libre
  - Un amplificador de transimpedancia (TIA, por sus siglas en inglés)
  - Un preamplificador
- Amplificador de instrumentación para ECG
- Interfaz eléctrica analógica (EAFE, por sus siglas en inglés)
- Convertidor de luz a frecuencia (LTF, por sus siglas en inglés)
- ADC SAR de 14 bits
- FIFO de 128 Bytes
- Cuatro puertos GPIO
- Interfaz I2C
- Secuenciador

Como puede observarse, el sensor cuenta con el hardware necesario para la adquisición de las dos señales fisiológicas sobre las cuales se sustenta el presente trabajo, es decir, la actividad cardiaca obtenida mediante PPG, así como la señal EDA. De igual forma cuenta con etapas de digitalización, filtrado y amplificación en las interfaces que permiten un primer acondicionamiento a las señales, además de utilizar el tipo de conversión analógica digital estipulada en la sección 4.2.4. Es por esta variedad de motivos que se selecciona el sensor AS7026GG como parte del prototipo.

Cada uno de los cuatro LEDs y los cinco fotodiodos disponibles cuentan con una fuente de corriente programable de hasta 100 mA, tal como se ilustra en la Figura 50. Para este trabajo se fija una corriente inicial de 2.5 mA al LED verde, de modo que durante la calibración de la amplitud de la señal se cuenten con 39 intervalos de corriente de 2.5 mA. En compañía del LED verde, se fijan dos fotodetectores (PD1 y PD2) para la generación de una corriente eléctrica a partir de la incidencia de los fotones reflejados por la piel. Ambos son conectados al amplificador de transimpedancia para permitir la conversión de la corriente a una diferencia de potencial. Además, se asigna un valor de

offset inicial de 70 nA a la fuente de corriente con la que disponen los fotodetectores, pues a partir de diversas pruebas realizadas con el sensor, se observa que dicho valor es un buen punto de partida para comenzar la calibración. De esta forma se evita prolongar el tiempo de calibración a intervalos superiores a los tres minutos, pues cada cambio realizado durante este proceso requiere de tres segundos debido a la ventana de datos adquirida de la señal.

El amplificador de transimpedancia, ilustrado en la Figura 50, dispone de una ganancia ajustable mediante la variación de los valores de

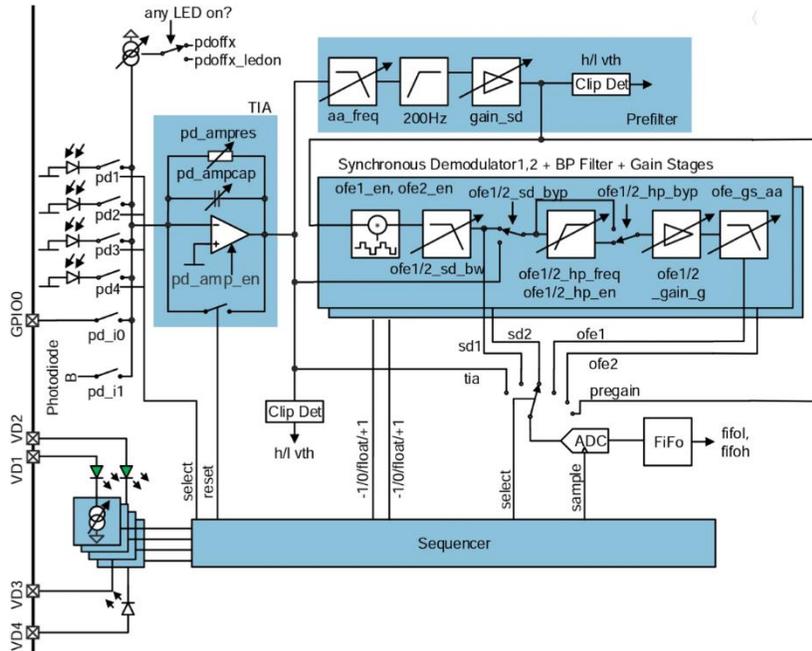


Figura 50. Interfaz óptica del sensor AS7026GG. Imagen modificada de [72].

un capacitor y una resistencia, además de contar con dos modos de operación, ya sea como conversión de corriente a voltaje o como integrador de corriente. De igual manera cuenta con un detector de umbral, una conexión configurable de alimentación entre alguno de los cinco fotodiodos disponibles o el GPIO0, así como una compensación por offset de corriente suministrada a los fotodiodos.

El sensor se encuentra enfocado primordialmente a frecuencias de muestreo superiores a 10 kHz, puesto que dispone de un filtro antialias para acotar el ancho de banda a este intervalo. No obstante, en el presente trabajo se busca utilizar una frecuencia de muestreo de 128 Hz, por lo que es necesario el uso de un modo de bajo ancho de banda para el convertidor de corriente a voltaje, el cual adecua al capacitor y la resistencia variable del amplificador a valores ya establecidos por el fabricante con una ganancia de  $7 V/\mu A$ .

El sensor cuenta con un convertidor analógico-digital de registros de aproximaciones sucesivas (SAR, por sus siglas en inglés) a 14 bits cuya alimentación parte de un reloj de 1 MHz, no obstante, se dispone de un divisor para reducir la frecuencia de funcionamiento del ADC si así se desea. De igual manera dispone de un multiplexor con 12 canales a elegir para realizar su muestreo. Tras cada conversión, las muestras van directamente a la FIFO de 128 Bytes y el multiplexor avanza automáticamente al siguiente canal habilitado. Sin embargo, en caso de utilizarse el secuenciador, el ADC comienza el muestreo con el canal más pequeño. La configuración de este módulo parte de la designación del reloj a 1 MHz junto con la habilitación de un modo de apagado automático en caso de no realizarse una conversión en el ADC. Este modo conlleva la designación de al menos  $64 \mu s$  en el tiempo de establecimiento de la señal en el registro *adc\_settling\_time*. A su vez, se realiza el multiplexado del ADC entre dos canales, ya sea entre la etapa pre-ganancia y la salida post ganancia del modulador sincrónico 1 (etapa posterior al OFE) durante la etapa de calibración, o bien entre la EAFE y la etapa posterior al OFE una vez calibrado el sensor biométrico.

Por su parte, entre las 10 fuentes de interrupción con las que cuenta el sensor, para la FIFO se configura una interrupción cuya habilitación transcurre al escribirse un único dato en ella, es decir, se define una profundidad de 2 Bytes.

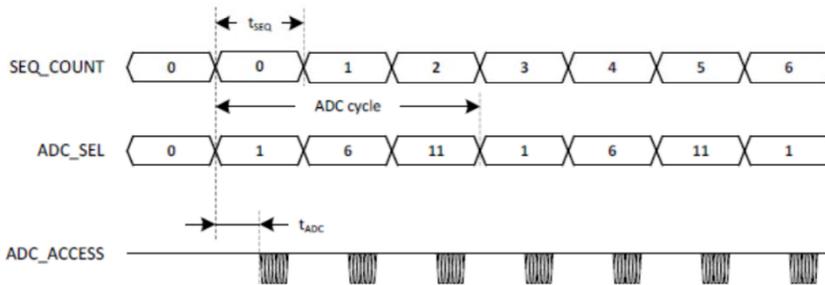


Figura 51. Rotación de los canales seleccionados en el ADC respecto a los periodos transcurridos en el secuenciador. Imagen adquirida de [72].

El secuenciador de la Figura 50 consta de un contador capaz de sincronizar dentro del periodo de una secuencia el tiempo de inyección de corriente a los LEDs, los tiempos de integración, los tiempos del demodulador síncrono y los tiempos de muestreo del ADC. Este módulo funciona con

un reloj interno de 1 MHz, el cual es capaz de alcanzar frecuencias menores gracias a un pre escalador de 8 bits definido mediante el registro *seq\_div*. Con la frecuencia anterior, el registro *seq\_period* define el número de muestras que conforman al periodo de una secuencia.

Es importante mencionar que el multiplexado realizado entre los canales del ADC limita el muestreo del sensor a exclusivamente una señal a lo largo del periodo de una secuencia. No obstante, al concluirse cada periodo del secuenciador, el ADC realiza una rotación automática de los canales restantes habilitados para que en cada secuencia se realice el muestreo de una señal diferente, por lo que, de esta forma, el sensor permite la adquisición de más de una señal. El funcionamiento anterior se ilustra dentro del ejemplo de la Figura 51, donde se realiza una rotación entre los canales 1, 6 y 11 del ADC para cada uno de los periodos del secuenciador.

En este trabajo se establece un valor de 17 dentro del registro *seq\_div* para pre escalar el periodo del reloj a 18  $\mu s$ . Asimismo, se define un valor de 217 dentro de *seq\_period* para que, en compañía de *seq\_div*, conforme un periodo de 3.906 ms para el secuenciador. De esta forma, se adquiere un muestreo de la señal EDA y la señal obtenida mediante PPG a 256.016 Hz, sin embargo, el sensor entrega un muestreo real de cada señal a 128 Hz al considerarse el efecto ocasionado por el multiplexado del ADC. Cabe mencionar que la tasa de muestreo de EDA a 32 Hz se alcanza mediante un remuestreo dentro del subsistema de Comunicaciones, donde se lee únicamente el último de cada cuatro datos recibidos por I2C de la señal. Con base en la configuración anterior, se designan los tiempos descritos en la Tabla 16 cuyo valor de 8 entre el inicio y final de la inyección de corriente a los LEDs (equivalente a 144  $\mu s$ ) parte del tiempo necesario para lograr la conversión completa de los 14 bits del ADC mediante su método SAR, lo cual implica un tiempo de 25  $\mu s$  considerando el periodo del ADC a 1  $\mu s$ , y del tiempo solicitado por el fabricante de al menos 64  $\mu s$  para habilitar el modo de apagado automático del ADC, lo que en conjunto entrega un tiempo mínimo requerido de 89  $\mu s$ . A partir de este requisito se fijan finalmente 144  $\mu s$  dada la efectividad que presentó este tiempo tras realizarse pruebas con el sensor. Esta selección resulta conveniente para la habilitación del LED, pues los fabricantes suelen recomendar la alimentación de LEDs verdes con corrientes de hasta 100 mA sólo si se maneja una alimentación de voltaje mediante PWM con un bajo ciclo de trabajo. Dado que la elección anterior otorga un ciclo de trabajo del 3.69%, no se posee ninguna dificultad para manejar corrientes de 100 mA.

Tabla 16. Configuración temporal del secuenciador del sensor AS7026GG.

Registro	Tiempo	Registro	Tiempo	Registro	Tiempo
seq_led_start	109	seq_itg_stop	0	seq_sdm1_start	109
seq_led_stop	117	seq_sdp1_start	1	seq_sdm1_stop	117
seq_itg_start	1	seq_sdp1_stop	9	seq_adc	109



elecciones anteriores se tomaron con base en pruebas y los trabajos revisados en el Estado del Arte.

La interfaz eléctrica analógica, ilustrada en la Figura 53, se comunica tanto con los cuatro GPIOs del sensor, como con el pin ECG\_REF del módulo ECG siempre que éstos se encuentren en modo analógico. Con estas entradas pueden adoptarse diversas topologías de amplificación ajustables a las necesidades del usuario. En el presente trabajo, la configuración del módulo sigue las recomendaciones dadas por el fabricante para la detección de señales EDA [74], que parte de la conformación de un circuito divisor de voltaje, considerando la resistencia de la piel como el componente variable, acompañado de un filtro paso bajas RC como se ilustra en el esquemático de la Figura 54. Para el filtro se considera una resistencia de  $10\text{ M}\Omega$  y un capacitor de  $220\text{ nF}$  a fin de brindar una frecuencia de corte de  $2.2\text{ Hz}$ , de esta forma se conforma un filtro antialias para la señal EDA. El circuito se alimenta a través del GPIO1 con  $1.9\text{ V}$  adquiridos del DAC interno del sensor. Asimismo, se lleva a cabo la lectura del divisor mediante el GPIO0 que atraviesa el amplificador no inversor interno, ubicado frente al ADC, con referencia a tierra. Ambos GPIOs se configuran en su modo analógico. Para la señal adquirida del GPIO0, se designa una ganancia unitaria para evitar la saturación de la señal a causa de la amplificación de la línea base con la que cuenta ésta.

Para llevar a cabo la configuración del sensor se conforma una función en cuyo interior se realiza la configuración individual de cada uno de los módulos antes descritos, lo cual se realiza a través de subfunciones. Cada una de ellas solicita como parámetro de entrada la dirección I2C del sensor que corresponde a  $0x30$ .

Debido a que no se dispone de ningún modulo integrado que haga uso del sensor AS7026GG bajo las necesidades de ergonomía buscadas para este proyecto, se toma la decisión de elaborar un circuito impreso que lo garantice. Para su diseño se parte nuevamente de las recomendaciones brindadas por el fabricante al final de [75] para conformar el esquemático de la Figura 54. En dicho circuito se realizan las conexiones necesarias para integrar los módulos utilizados en este trabajo, en adición de

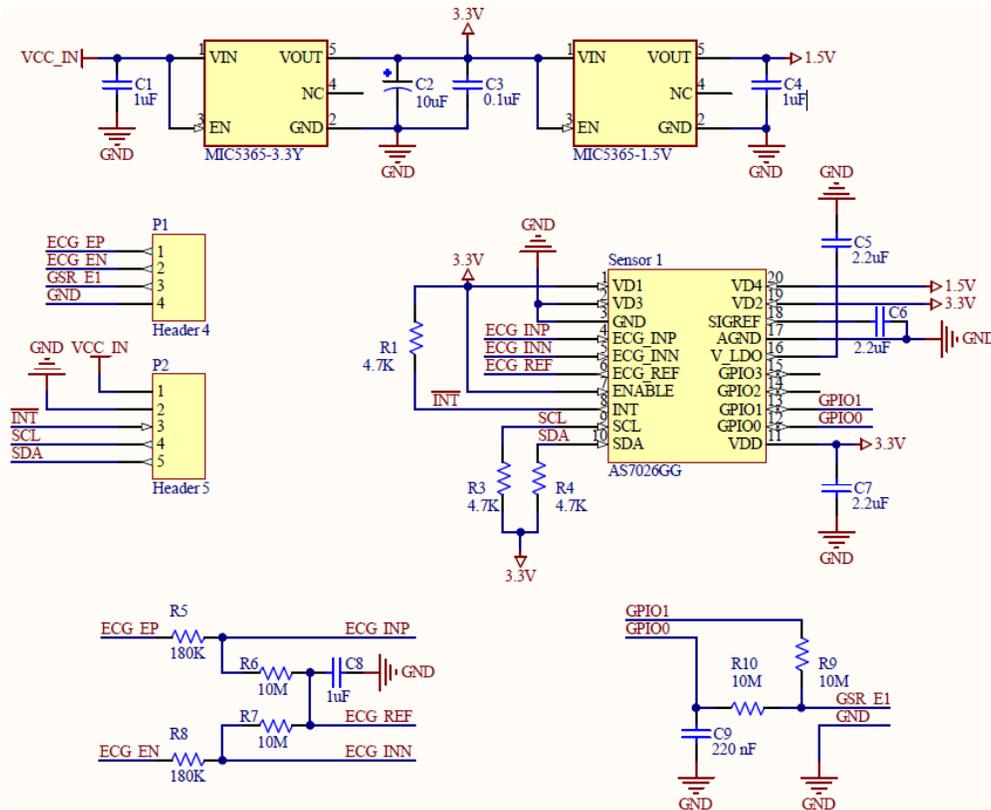


Figura 54. Esquemático del módulo AS7026GG.

algunos adicionales como el módulo ECG y la habilitación del LED infrarrojo dado que se pretende explorar dichas modalidades en futuros trabajos. El valor de las componentes se rige con base en las recomendaciones del fabricante para el funcionamiento correcto del sensor.

#### **4.4.3 Acondicionamiento de la señal**

Para atenuar el ruido durante la etapa de acondicionamiento de las señales fisiológicas se opta por utilizar dos filtros digitales FIR, uno paso bajas con frecuencia de corte de 2.1 Hz en el caso de la EDA y otro paso banda con frecuencias de corte de 0.67 y 3.33 Hz para la señal obtenida a partir de la PPG. En el caso de la EDA, el filtro es diseñado únicamente para atenuar el ruido de alta frecuencia tomando como base el intervalo estipulado en la sección 3.1.4. Por otra parte, en el caso de la banda de paso elegida para la señal obtenida a partir de la PPG, se toma en consideración un intervalo de frecuencias que permita la extracción adecuada de la señal eléctrica tomando en cuenta un ritmo cardiaco máximo de 200 bpm y mínimo de 40 bpm, tal como se indica en la sección 3.2.3.1, eliminando de igual forma el ruido de alta frecuencia y las oscilaciones de la línea base. La razón por la que no se utiliza un filtro paso banda en la señal EDA es debido a que el método empleado para la obtención del SCR elimina por sí solo la línea base, lo cual se explica a detalle en la sección 4.4.4.2.

Si bien en la sección 4.4.2.2 se mencionó el uso de filtros internos al momento de adquirir las señales, existen algunas razones por las que se elige implementar un filtrado externo además del suministrado por el propio sensor biométrico. En el caso de la señal EDA, es imprescindible al menos un filtro paso bajas FIR, pues el filtro analógico implementado en el circuito impreso es susceptible a cambios por factores físicos como puede ser la temperatura, siendo poco confiable para señales susceptibles a ruido, tal como se mencionó en la sección 4.2.4. En cuanto a la señal adquirida mediante PPG, tras las pruebas realizadas con el filtro FIR se obtuvo una señal más limpia y fiel a lo que se busca representar, obteniendo una mejor SNR y cumpliendo con los requerimientos propuestos en las Especificaciones objetivo, lo que no hubiese sido posible utilizado únicamente el filtrado interno del sensor. Los resultados de las SNRs están descritos con mayor detalle en la sección 6.2.2.

Respecto al diseño de los filtros, éstos deben generarse mediante el método de ventanas a partir de una ventana Hamming, pues su respuesta en frecuencia presenta lóbulos laterales menores que evitan la existencia de rizados en la banda de paso y la banda suprimida, obteniendo así una adecuada atenuación sin afectar el intervalo de frecuencias útil, sin embargo, la desventaja de esta ventana es su banda de transición ancha producto de su amplio lóbulo principal. Para mitigar esto, ambos filtros deben ser diseñados con un orden de 100 o superior, pues el ancho del lóbulo principal es inversamente proporcional a la longitud de la ventana utilizada.

Los filtros FIR deben implementarse en su forma directa y de fase lineal, explotando así la simetría de la respuesta al impulso y reduciendo a la mitad el número de coeficientes. Además, estos últimos se transformarán a un formato de punto fijo en 16 bits para minimizar el espacio en memoria requerido.

Considerando lo anterior, es necesario implementar dos funciones dedicadas al filtrado de las señales de interés, las cuales deben recibir como argumento un dato de la respectiva señal y, posteriormente, retornar un valor filtrado. Cabe resaltar que, al utilizar filtros FIR, los datos retornados corresponden a muestras anteriores debido al retraso que se introduce durante el filtrado, el cual equivale a un retardo en muestras igual a la mitad del orden del filtro. Asimismo, se opta por manejar las señales sin dimensiones antes y después del filtrado, pues las características extraídas serán igual de relevantes para el clasificador, además de evitar el uso de la FPU durante el desarrollo del algoritmo.

#### **4.4.4 Extracción de características**

Para la descripción de los componentes de este subsistema, primero se ahonda en la técnica utilizada para la generación de la señal PRV, seguido de la descripción de los elementos que permiten la identificación de las características temporales en las señales SCR y PRV, así como las no lineales en PRV. Finalmente, se describe la metodología a seguir para la adquisición de las características en el

dominio de la frecuencia de ambas señales, así como las consideraciones que se realizan sobre sus ventanas de datos para alcanzar un análisis secuencial entre ellas.

#### 4.4.4.1 Generación de PRV

Tal como es establecido en la sección 4.3.4, previo a la extracción de características es indispensable estimar la PRV de la señal (sección 3.2.3.1), lo cual es realizado mediante la detección de picos sistólicos y, posteriormente, el cálculo de los IPP. En la sección 3.1.3 se citaron diversos algoritmos que han presentado buenos resultados, sin embargo, algunos de éstos se basan en transformaciones que implican el uso excesivo de recursos computacionales, lo cual afecta al tiempo de respuesta y reduce el espacio en memoria. Por otro lado, dentro de los algoritmos estudiados se encuentran aquellos dirigidos a aplicaciones en tiempo real, tal como el trabajo citado en [42], el cual demostró ser ligero, rápido y se acopla a las especificaciones planteadas para el prototipo portátil. Debido a esto, el método empleado en la presente tesis se basa en dicho algoritmo, denominado como *Mountaineer's method*. Este principio parte de la detección de flancos ascendentes en la señal de PPG, donde se considera que la última muestra que conforma el flanco ascendente es el pico sistólico (como si fuera la cima de una montaña). El primer paso es comparar dos muestras sucesivas, para lo cual es aplicada la diferenciación finita hacia atrás, tal como se muestra en la ecuación (62).

$$d = x[n] - x[n - 1] \quad (62)$$

Si  $d$  adquiere un valor mayor a cero, es decir que  $x[n]$  es mayor a  $x[n - 1]$ , indica que la señal aumentó de amplitud y podría ser el comienzo de un flanco ascendente. Para ser considerado un pico sistólico, el flanco ascendente previo al mismo debe sobrepasar un umbral definido, es decir, las diferencias  $d$  mayores a cero sucesivas deben superar un número establecido. En este trabajo se selecciona un umbral de 25 muestras (a una frecuencia de muestreo de 128 Hz), el cual es determinado mediante pruebas con el sensor biométrico y tomando en cuenta características fisiológicas inherentes a la frecuencia cardíaca, tal como el periodo sistólico ventricular.

En la Figura 55 se ilustra la detección de picos sistólicos, en la cual los flancos ascendentes resaltados en verde son aquellos que sobrepasan el umbral establecido, por lo que la última muestra que los conforma es considerada un pico sistólico. Por el contrario, los flancos ascendentes resaltados en rojo, tal como el flanco previo a un pico diastólico o ruido presente en la señal, son descartados debido a que no sobrepasan el umbral.

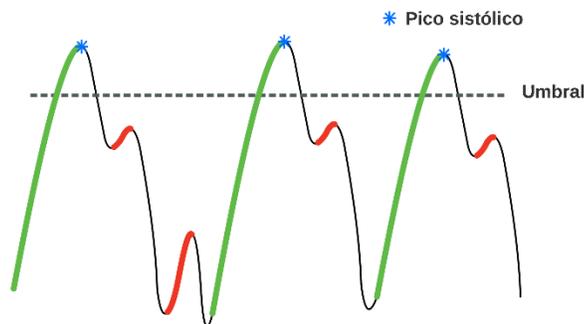


Figura 55. Detección de picos sistólicos mediante un umbral

Como se vio en la sección 3.1.3, la señal de PPG es propensa a diversos artefactos de movimiento, lo cual propicia que en ocasiones sea irreconocible un pico o que, al presentarse ruido de alta amplitud, se detecten picos falsos. Para lidiar con ambos casos se desarrolla un método basado en la estimación del promedio de los IPP, donde éste es calculado conforme es analizada la señal fisiológica. Por lo tanto, cuando se obtiene un nuevo intervalo, si éste difiere en un 25% o más del valor promedio calculado hasta el momento, se concluye que hubo un error en su estimación. Cuando esto sucede, simplemente se descarta el IPP y se continúa analizando la señal hasta encontrar el siguiente pico.

Si bien basta con la adquisición de dos puntos de la señal obtenida mediante PPG para la identificación de los IPP, la ventana de datos necesaria para la PRV resulta de longitud variable, pues su extensión depende directamente de la frecuencia cardiaca de cada individuo. Por ende, se contempla registrar un máximo de 853 puntos ya que resulta en un caso extremo donde la obtención de los valores IPP se da a una tasa de 0.3 s, lo cual corresponde a una frecuencia máxima de 200 bpm. Considerando lo anterior, se adquiere una ventana de 256 s tras una interpolación de los datos de la PRV. La selección del tamaño de la ventana, así como la necesidad de realizar una interpolación, se explica con mayor detalle en la sección 4.4.4.5.

#### 4.4.4.2 Obtención de SCR

Una vez filtrada la señal de EDA por el subsistema de acondicionamiento de la señal, es necesario obtener su componente fásica (SCR), de la cual son extraídas las características de interés. Esto se lleva a cabo empleando un método similar al descrito en la sección 3.2.3.2, en donde se deriva la señal de EDA mediante la ecuación (62) y, posteriormente, se realiza la convolución con una ventana Bartlett de 32 puntos, también conocida como ventana Triangular, la cual está definida por la ecuación (63).

$$w(n) = \begin{cases} \frac{2n}{N} & 0 \leq n \leq N/2 \\ 2 - \frac{2n}{N} & N/2 < n \leq N \end{cases} \quad (63)$$

Una vez obtenida la señal deseada, los SCR son detectados mediante la identificación de cruces por cero. El cruce que indica el comienzo de un evento es de negativo a positivo, mientras que el cruce de positivo a negativo indica el final del mismo. La amplitud del SCR se obtiene al encontrar el valor máximo entre ambos cruces. En la Figura 56 se ilustran los componentes de los SCR presentes en la señal.

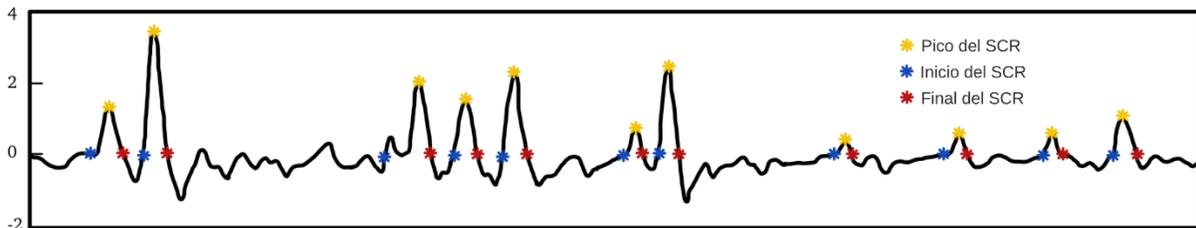


Figura 56. Detección de SCR en la señal de EDA filtrada

Para esta señal se selecciona una ventana de datos de 64 segundos dada la gran efectividad demostrada en un estudio para la detección de estrés en un individuo. En éste se realizó una comparación con ventanas de 30, 60, 180 y 300 segundos [76]. De igual forma, dicha ventana brinda facilidad para el análisis en frecuencia de la señal mediante el método de Burg, el cual emplea una FFT radix 2.

Una vez generada tanto la PRV como la detección de los SCR, se procede a la estimación de características que, tal como es ilustrado en la Figura 40 de la sección 4.3.4, consta de 3 bloques correspondientes al método no lineal de la gráfica de Poincaré, análisis en el dominio del tiempo y en el dominio de la frecuencia. A continuación, se describen las expresiones y consideraciones para calcular cada una de las características seleccionadas en el presente trabajo.

#### 4.4.4.3 Dominio del tiempo

Las características correspondientes al dominio del tiempo, cuya selección es justificada en la sección 4.2.4, se enlistan en la Tabla 17 para PPG y en la Tabla 18 para EDA.

Tabla 17. Características correspondientes al dominio del tiempo para PPG.

Descripción	Parámetro
Media de los IPP	$PPI_{med}$
Desviación estándar de los IPP	$SDNN$
RMS de las diferencias sucesivas de los IPP	$RMSSD$
Número de IPP adyacentes que varían por más de 50ms	$NN50$
NN50 dividido por el número total de IPP	$pNN50$

Tabla 18. Características correspondientes al dominio del tiempo para EDA.

Descripción	Parámetro
Número de SCR presentes en el segmento	$NumSCR$
Media de las amplitudes de los SCR	$AmpSCR$
Desviación estándar de las amplitudes de los SCR	$sdAmpl$
Media de los tiempos de levantamiento	$mTL$
Cuantil al 25%	$ctl25$
Cuantil al 50%	$ctl50$
Cuantil al 75%	$ctl75$
Cuantil al 90%	$ctl90$

Cabe resaltar que, a pesar de emplear la PPG para la estimación de intervalos, por convención, el nombre designado a las características es el mismo que se emplea para el ECG.

#### Métodos estadísticos

La mayoría de las características pertenecientes al dominio del tiempo seleccionadas para ambas señales fisiológicas, corresponden a estimaciones estadísticas presentadas en la sección 3.2.1.1, tal como la media aritmética, la RMS y la desviación estándar, las cuales son calculadas mediante las ecuaciones (2), (3) y (4), respectivamente. En la Tabla 19 se enlistan dichas características, al igual que sus correspondientes expresiones estadísticas, donde  $N$  es el número total de IPP o de ER-SCR presentes en el segmento de PRV y SCR, respectivamente.

Mediante la tabla es posible observar que las estimaciones se calculan de manera similar, ya que en todas se realiza la adición de elementos, para posteriormente dividirse entre el total de los mismos.

Tabla 19. Características y sus correspondientes ecuaciones estadísticas.

Ecuación estadística	Característica equivalente	
	PPG	EDA
$m_x = \frac{1}{N} \sum_{n=0}^{N-1} x[n]$	$PPI_{med} = \frac{1}{N} \sum_{n=0}^{N-1} PPI[n]$	$Amp_{med} = \frac{1}{N} \sum_{n=0}^{N-1} Amp[n]$ $TL_{med} = \frac{1}{N} \sum_{n=0}^{N-1} TL[n]$
$s = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} (x[n] - m_x)^2}$	$SDNN = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} (PPI[n] - PPI_{med})^2}$	$AmpSD = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} (Amp[n] - Amp_{med})^2}$
$RMS = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} x[n]^2}$	$RMSSD = \sqrt{\frac{1}{N-1} \sum_{n=1}^N (PPI[n] - PPI[n-1])^2}$	-
$ctl_k = \frac{n * i}{100}$	-	$ctl_k = \frac{numSCR * i}{100}$

Los intervalos adyacentes que varían por más de 50ms ( $NN50$ ), son definidos mediante la siguiente desigualdad:

$$abs(IPP[n] - IPP[n - 1]) \geq 50ms$$

Una vez obtenido, se calcula  $pNN50$  empleando la ecuación (31).

#### 4.4.4.4 Método no lineal de la gráfica de Poincaré

Tal como se aborda en la sección 3.2.3.1, para obtener las características derivadas de la gráfica de Poincaré, primeramente se calculan las proyecciones de los puntos ( $IPP_n, IPP_{n+1}$ ) sobre las rectas descritas por las ecuaciones (32) y (33), las cuales se estiman mediante las siguientes expresiones:

$$dist1[n] = \frac{1}{\sqrt{2}}IPP[n] + \frac{1}{\sqrt{2}}IPP[n + 1] \quad (64)$$

$$dist2[n] = -\frac{1}{\sqrt{2}}IPP[n] + \frac{1}{\sqrt{2}}IPP[n + 1] \quad (65)$$

Una vez obtenidas las proyecciones correspondientes a cada eje, se determinan las desviaciones estándar de cada una, tal como se muestra en (66) y (67):

$$SD1 = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} (dist1[n] - Distmed)^2} \quad (66)$$

$$SD2 = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} (dist2[n] - Distmed)^2} \quad (67)$$

#### 4.4.4.5 Dominio de la frecuencia

Explorando la teoría plasmada en la sección 3.2.2.2, es evidente que no es posible realizar de forma directa la estimación de la PSD sobre la señal PRV mediante el método de Burg, pues ésta parte del análisis de señales con tasa de muestreo constante, como se observa en término  $k$  de la expresión (9), y en realidad la naturaleza de la PRV refleja por sí misma una tasa de muestreo variable, tal como se explica en la sección 3.2.3.1. Por ende, para realizar su análisis en frecuencia se requiere de un remuestreo de la señal a una tasa constante mediante la interpolación de los datos que conforman la PRV. Para este trabajo se escoge el método de interpolación por trazadores cúbicos con terminación natural, pues su algoritmo de estimación puede adaptarse a uno que evite la reestimación completa de la interpolación tras la adquisición de un nuevo dato, además de resultar uno de los algoritmos más rápidos para la estimación de espectros mediante modelos AR [77]. Para la interpolación se toma una tasa de 8 Hz con el objetivo de cumplir con el criterio de Nyquist que posee la señal PRV, considerando una frecuencia máxima del ritmo cardiaco de 200 bpm [78] (equivalente a 3.33 Hz), además de facilitar el proceso de desplazamiento de datos, el cual se explicará más adelante en esta sección. De esta forma, se escoge una ventana de 256 muestras de la señal PRV interpolada (equivalente a 256 s de la señal adquirida de la PPG) a fin de permitir la estimación de la PSD de Burg mediante una FFT radix 2, dando lugar a un total de 2048 componentes espectrales y una resolución de 0.003906 Hz, la cual concuerda con el valor mínimo de 0.003 Hz para el intervalo VLF en la PRV, para más información véase la sección 3.2.3.1. Asimismo, para la estimación de la PSD únicamente se requiere del logaritmo base diez de 102 valores del espectro dado que con ellos se abastecen los intervalos de frecuencia deseados para PRV (VLF, LF y HF).

Para la estimación de la PSD de la señal SCR se mantiene la ventana de datos de 64 s, la cual, en conjunto con su tasa de muestreo de 32 Hz, da lugar a un total de 2048 componentes espectrales estimadas mediante una FFT radix 2, la cual cuenta con una resolución en frecuencia de 0.015625 Hz por componente espectral. Para esta estimación solo se requiere del logaritmo base diez de 22 valores del espectro, pues con ellos se abastece el intervalo de frecuencia deseado (HF). El resumen de las elecciones anteriores puede observarse en la Tabla 20.

Tabla 20. Propiedades del análisis en frecuencia de las señales PRV y SCR.

Señal	Fs [Hz]	Ventana [s]	NFFT	Resolución [mHz]
PRV	8	256	2048	3.90625
SCR	32	64	2048	15.625

Debido a la diferencia en el tamaño de las ventanas entre las señales PPG y SCR, resulta imposible el análisis secuencial de ambos para su extracción de características. Por este motivo, se opta por analizar secuencialmente sólo la señal SCR mientras que la señal PRV adopta un mecanismo de desplazamiento que permite realizar su análisis en periodos constantes de tiempo, los cuales vayan a la par de la ventana de datos de la señal SCR. De esta forma, la señal SCR se analiza cada 64 segundos, mientras que la señal PRV también es analizada cada 64 segundos considerando los 256 s previos de su ventana. No obstante, debido a que la señal PRV es de longitud variable, se implementa otro mecanismo que permita identificar el tamaño que va adquiriendo ésta a fin de posibilitar el desplazamiento adecuado de los datos, así como dar pie al inicio de las estimaciones respectivas en frecuencia. El funcionamiento de este mecanismo se ilustra en la Figura 57, partiendo, en los pasos 1 y 2, de la adquisición de al menos 256 s de la señal PRV calculada de forma acumulativa mediante la variable “UpAcum”. Posteriormente, se realiza la interpolación de los datos en el paso 3 desde el índice “Low

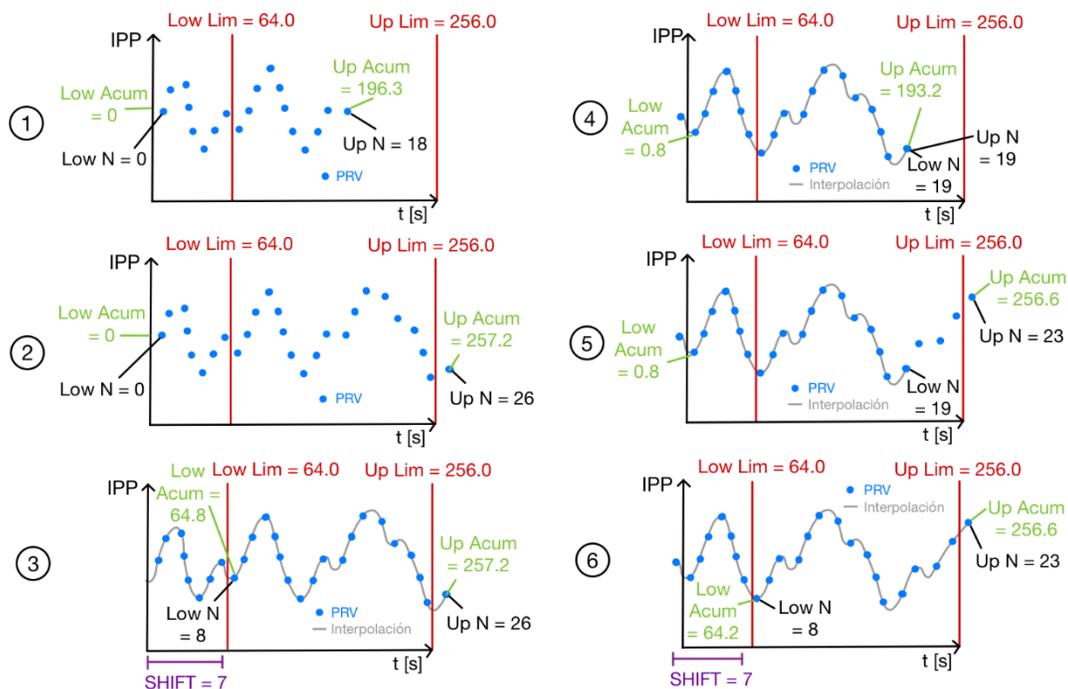


Figura 57. (1) Primera adquisición de datos de PRV (2) Adquisición completa de 256 s de datos de la PRV (3) Interpolación para remuestreo de la señal y estimación de la longitud del traslape para PRV considerando un traslape de 64 s (4) Segunda adquisición de datos de PRV a partir del último PRV adquirido (5) Adquisición completa de 256 s de los datos de la PRV (6) Interpolación de nuevos datos adquiridos para remuestreo de la señal y estimación de la longitud del traslape para PRV considerando un traslape de 64 s

N” hasta el índice “Up N” del paso 2, seguido de la estimación del número de elementos por desplazar en la variable SHIFT para lograr un desplazamiento en la señal de 64 s en el paso 4. Esto último se logra gracias a la variable acumulativa “Low Acum” en el paso 3. Más tarde, se sigue la adquisición de los siguientes 64 s de la señal tras el desplazamiento de los datos PRV en el paso 5. Y finalmente, se realiza otra interpolación con los nuevos datos adquiridos, realizándolo desde el índice “Low N” hasta “Up N” en el paso 6. Los pasos 1 a 3 suceden al inicio del proceso de adquisición de los datos PRV, mientras que los pasos 4 a 6 se repiten continuamente conforme se adquieren más datos de la señal PRV.

En teoría la interpolación por trazadores cúbicos solicita la resolución de un sistema de ecuaciones cuya matriz de constantes resulta de tamaño  $(N - 2) \times 3$ , así como un vector de incógnitas y un vector de constantes, ambos de tamaño  $N - 2$ . Sin embargo, el solicitar como parámetro de entrada los datos de las ordenadas en forma de intervalos entre datos sucesivos  $(x_1 - x_0, \dots, x_{i-1} - x_{i-2}, x_i - x_{i-1})$ , en lugar de una secuencia creciente de valores  $(x_0, \dots, x_{i-1}, x_i, x_{i+1})$ , así como la selección de una terminación natural para la interpolación, permiten la reducción de la matriz de constantes a un tamaño  $N - 2$ .

La adaptación del algoritmo de interpolación mencionada con anterioridad, la cual evita la reestimación completa de la función interpolada tras la adquisición de un nuevo dato, se comprende gracias a que el método realiza la interpolación de los datos a través una función por partes. Cada una de ellas se conforma por una función cúbica  $f_n(x)$  que parte del uso de una vecindad cerrada de tres puntos y, como requisito, cada función estimada debe ser continua en su segunda derivada  $f_n''(x)$  con las funciones cúbicas adyacentes. Con esto, dentro del algoritmo de interpolación se aprovecha que en el paso de la sustitución hacia atrás, realizada tras completarse la eliminación de Gauss de la matriz tridiagonal conformada, existe una dependencia de las funciones lineales adquiridas  $f_n''(x)$  con su función lineal anterior  $f_{n-1}''(x)$ , es decir, que la adición de un punto al conjunto de datos involucra que la función cúbica conformada con éste altere la función lineal anterior, que a la vez cambiará a su anterior y así sucesivamente, hasta modificar la primera función lineal  $f_0''(x)$ . No obstante, estos cambios disminuyen conforme se avanza en la sustitución hacia atrás, de forma que se identifica una convergencia de las modificaciones realizadas hacia los resultados previamente obtenidos sin el nuevo dato. De esta forma, en el presente trabajo se establece un límite en la sustitución hacia atrás del algoritmo, siendo hasta que las nuevas modificaciones alcancen un valor inferior al 1% de las funciones lineales obtenidas previamente sin el nuevo dato en el conjunto.

Para lograr el desplazamiento de los datos se recurre al uso de canales DMA dada la facilidad que éstos brindan para realizar el traslado de información de una localidad de memoria a otra, siendo a una gran velocidad y mediante una ejecución en paralelo que no afecta al código principal en el procesador. No obstante, además de las abscisas de PRV y los intervalos de PRV, también se considera el desplazamiento de la señal interpolada y todos los elementos que conforman al sistema de ecuaciones de la interpolación por trazadores cúbicos, es decir, el vector de coeficientes, el vector de incógnitas y el vector de constantes, así como la señal interpolada. De esta forma resulta necesario el uso de al menos seis canales DMA. La configuración de éstos se realizará dentro de los archivos de este subsistema.

Para la estimación de la PSD mediante el método de Burg se conforman dos funciones, una primera enfocada a la estimación de los coeficientes AR de la señal, cuyos parámetros de entrada son los datos de la señal por analizar, el orden y el tipo de señal a ingresar, ya sea PRV o EDA. La segunda función se encarga de realizar la estimación de la PSD a partir de los coeficientes AR previos, solicitando como parámetros de entrada el valor de rho y la frecuencia de muestreo de la señal. Dentro de la función anterior se conforma otra función que realiza la estimación de la FFT radix 2 con decimación en el tiempo utilizando un formato de punto flotante a 32 bits, siendo sus parámetros de entrada la señal por analizar y el número de puntos de la FFT por efectuar, además de retornar un apuntador con el vector que contenga el espectro en frecuencia. Para esta última función se utiliza una unidad

trigonométrica que permite optimizar la velocidad de ejecución del algoritmo pues proporciona la estimación de las funciones seno y coseno mediante unidades de hardware.

A su vez, se elabora una función que permita realizar la interpolación de los datos de la PRV a través del método de trazadores cúbicos. Ésta maneja como parámetros de entrada los intervalos de las ordenadas y las abscisas de la señal por interpolar, además de retornar un apuntador al arreglo con los datos interpolados. Dentro de esta función se realizan los pasos 3 y 6 de la Figura 57, con los cuales se estiman los parámetros SHIFT, Low N y Low Acum del mecanismo de desplazamiento ahí ilustrados.

Finalmente, se constituye una función general que haga uso de las funciones anteriores y sus respectivos resultados para realizar la estimación y asignación de las características en frecuencia correspondientes, las cuales son accesibles mediante el registro “Caract Config” de la Tabla 14. Los parámetros de entrada de esta función corresponden a los intervalos de las ordenadas, las abscisas y el tipo de señal a ingresar, esto dado que PRV requiere del ingreso de ambas entradas, mientras que EDA únicamente requiere del ingreso de la entrada de las abscisas.

#### **4.4.5 Clasificación**

El subsistema de clasificación debe implementarse por separado en dos funciones que permitan realizar la reducción y clasificación de las características extraídas tomando en consideración cada etapa del esquema EMDC, tal como se abordó en la sección 4.3.5.

La función enfocada a FLD debe recibir un argumento que permita identificar la etapa del esquema donde se está utilizando el algoritmo, pues es necesario identificar la selección de características de la SFFSn correspondiente. Sumado a esto, la función debe recibir el valor de las características a comprimir y su respectiva matriz  $W_n$  tomando en consideración los requerimientos indicados en la sección 4.3.5. Dado que el esquema EMDC está conformado por clasificaciones binarias y existe la condición dimensional de FLD expresada en el apartado 3.4.3, la salida de cada FLDn será un vector unidimensional.

La función enfocada a LDA de igual forma debe recibir un argumento que especifique la etapa del esquema EMDC en que se está ejecutando el algoritmo, pues es necesario identificar correctamente los parámetros  $m_{k,n}$ ,  $\ln(\pi_k)_n$  y  $S_n$  para realizar las estimaciones probabilísticas de cada LDAn. Del mismo modo, la función debe recibir el vector unidimensional proveniente de FLDn para realizar la clasificación pertinente.

Finalmente, es necesario ensamblar el esquema dentro de una función general EMDC que recibe el vector de características a clasificar. La salida de dicha función debe contener un valor codificado que permita discriminar la emoción detectada con base en los resultados de las clasificaciones de neutralidad, excitación y valencia. El valor será escrito en el registro “Emoción” indicado en la Tabla 13 utilizando la codificación estipulada para el mismo.

#### **4.4.6 Selección del dispositivo comercial**

Con base en las características mencionadas con anterioridad para los diversos subsistemas del proyecto, se identifica la necesidad de un dispositivo que disponga de al menos una interfaz I2C, una interfaz UART, siete canales DMA, así como una unidad de punto flotante y una unidad matemática trigonométrica para el cálculo eficiente de funciones trigonométricas. Para esto, se elige trabajar con un procesador digital de señales (DSP, por sus siglas en inglés) dado que este tipo de dispositivos cuentan con dichos módulos integrados con el procesador, además de disponer de una amplia gama de productos a un precio reducido. Se descartan los MCU’s y los FPGA’s dado que los primeros no cuentan con las unidades matemáticas necesarias y los segundos debido a que primordialmente se enfocan hacia aplicaciones con un alto grado de paralelismo, lo cual no es el fin particular de este

proyecto. Por ende, para el siguiente trabajo se escoge el DSP TMS320F28379D de la familia C2000 producida por Texas Instruments, el cual está dentro de su kit de desarrollo LAUNCHXL-F28379D. El motivo de su elección es debido a que cumple con las necesidades anteriores, además de ser uno de los elementos disponibles dentro del equipo de trabajo. No obstante, se resalta al lector que existen otras opciones que pueden satisfacer igualmente las necesidades mencionadas.

El DSP TMS320F28379D, orientado a las necesidades del detector, cuenta con dos CPUs TMS320C28x con arquitectura RISC, una unidad de punto flotante de precisión simple con respecto a la IEEE 754, una unidad matemática trigonométrica, un reloj interno de hasta 200 MHz, seis canales DMA, dos interfaces I2C y cuatro interfaces de comunicación serial (SCI/UART).

De esta forma, el esquemático que integra a los módulos seleccionados con el kit de desarrollo LAUNCHXL-F28379D puede observarse a través de la Figura 58.

#### 4.4.7 Sistema en la computadora

Como se mencionó en la sección 4.3.6., el sistema desarrollado en la computadora ejecuta funciones de gran utilidad para el detector de emociones, las cuales buscan reducir la carga computacional del sistema embebido y proporcionar herramientas que faciliten su uso. Es por este motivo que resulta necesario hacer un análisis detallado de su funcionamiento, especificando las particularidades de cada subsistema y algoritmo implementado.

##### 4.4.7.1 Interfaz gráfica de usuario

A continuación, son descritos a detalle los elementos enlistados en la sección 4.3.6.1, así como su interacción con el resto de los subsistemas en la computadora y el detector de emociones. En la Figura 59 se muestra el esquema general de la interfaz gráfica de usuario, donde es ilustrada la disposición de los elementos y secciones planteadas con anterioridad.

**Conexión:** la interfaz cuenta con un botón para enlazar, mediante el subsistema de Comunicación RF, la computadora con el prototipo. Inicialmente, los campos de selección y los botones de envío de datos se encuentran deshabilitados, éstos se activarán en cuanto se establezca la conexión una vez presionado el botón de “Conectar”.

**Medición:** se implementa el botón “Iniciar medición” con el propósito de habilitar la calibración del sensor biométrico, así como la adquisición de las señales fisiológicas. Para esto, primeramente, se debe establecer la conexión con el detector de emociones.

**Sección de configuración:** en este bloque se realiza la selección de algunos de los parámetros descritos en la sección 4.4.1, tal como la selección de modalidad (entrenamiento o prueba), la opción de habilitar el dispositivo optoelectrónico y la transmisión automática de datos. Una vez seleccionados los parámetros correspondientes, se debe presionar el botón de “Enviar configuración”.

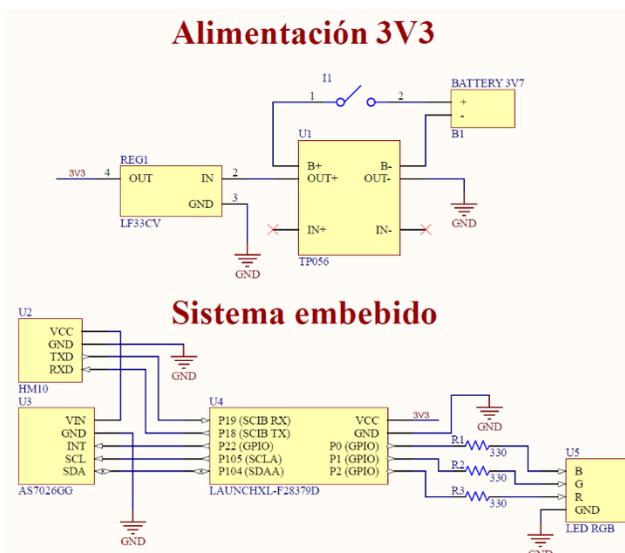


Figura 58. Esquemático del detector de emociones

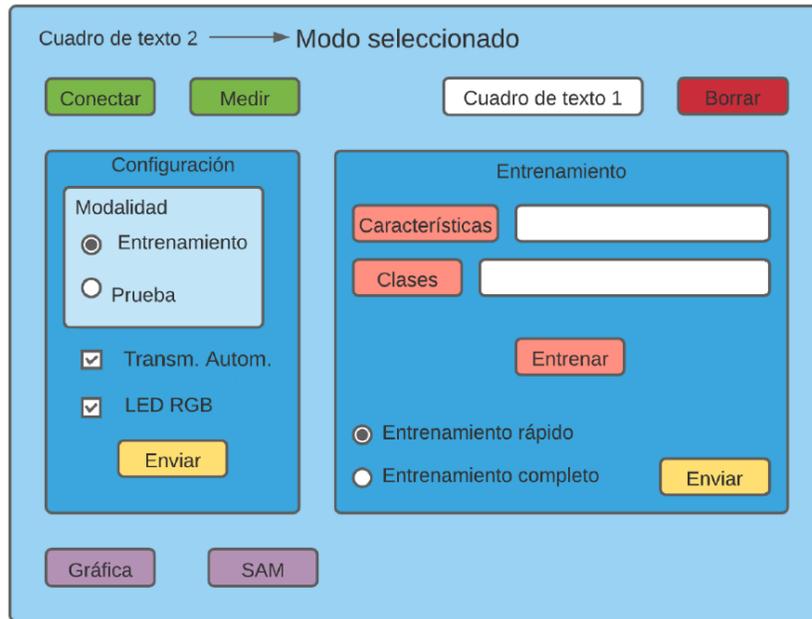


Figura 59. Esquema general de la interfaz gráfica de usuario.

**Sección de entrenamiento:** para realizar la reducción de características y entrenar al clasificador, es necesario seleccionar dos archivos: uno que contiene los vectores de características y otro en el cual se encuentren las clases correspondientes a cada vector. Para esto, se cuenta con dos botones que despliegan la ventana del explorador de archivos de la computadora, donde se podrán seleccionar los documentos previamente mencionados. Ambos archivos deben ser elegidos antes de presionar el botón de entrenamiento, de lo contrario, el sistema no será entrenado y se mostrará una ventana indicando los documentos que faltan de ser seleccionados. Además, en esta sección es posible elegir uno de los dos tipos de entrenamiento disponibles (*rápido* o *completo*), cuya descripción a detalle se da en la sección 4.4.7.3. Una vez realizado lo anterior, es preciso presionar el botón “Entrenar” para enviar los datos al subsistema SFFS y, una vez obtenidos los parámetros correspondientes mediante los subsistemas LDA y FLD, éstos son enviados al detector de emociones presionando el botón de “Enviar parámetros”. También se incluye el botón “Borrar”, el cual elimina los datos contenidos en los archivos de características y clases seleccionados con anterioridad.

**Maniquí de autoevaluación:** la interfaz permite visualizar el maniquí mediante el botón “Esquema SAM”, el cual abre una ventana donde el participante debe seleccionar los niveles correspondientes al grado de valencia y excitación que experimentó durante la presentación del estímulo. Posteriormente, el participante debe capturar los valores presionando el botón “Registrar” encontrado en la parte inferior de la ventana.

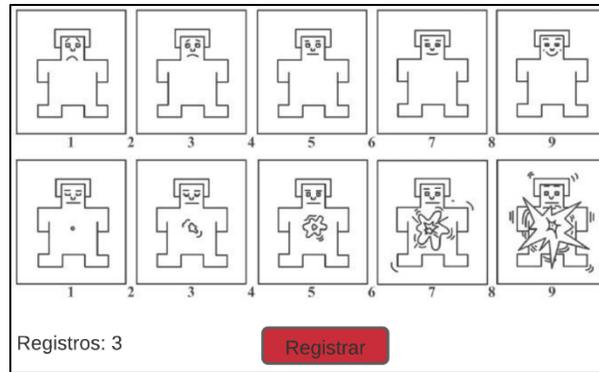


Figura 60. Ventana del esquema SAM

**Visualización de resultados:** para representar los datos de manera ilustrativa y permitir al investigador interpretar los resultados de la mejor forma, se cuenta con una gráfica conformada por cinco barras, donde cuatro corresponden a cada cuadrante del modelo dimensional valencia-excitación y la quinta al estado neutro. Esta herramienta brinda una visualización clara de la ocurrencia de las emociones a lo largo de un intervalo de tiempo determinado. Para generarla es necesario presionar el botón “Mostrar gráfica”, el cual abre una ventana similar a la que se ilustra en la Figura 61.

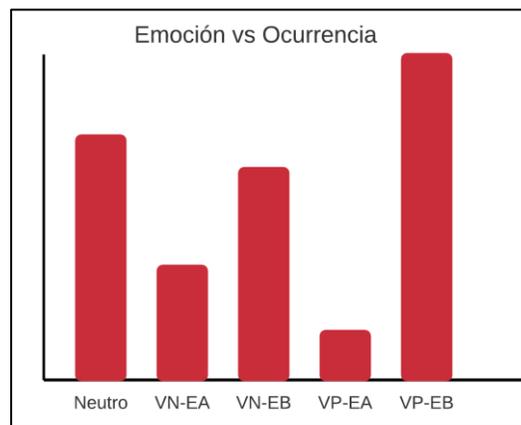


Figura 61. Gráfica de barras de Emoción vs Ocurrencia

**Cuadros de texto:** se cuenta con dos cuadros de texto con el objetivo de facilitar al usuario el manejo de la interfaz. El primero indica el procedimiento que debe seguirse para realizar la calibración del sensor biométrico, mientras que el segundo muestra la modalidad actual del sistema. El modo predeterminado es el de *entrenamiento*.

#### 4.4.7.2 Comunicación RF

Para facilitar el uso de este subsistema con el resto de los subsistemas integrados en la computadora se realiza su implementación a través de una clase cuyos atributos y métodos sean públicos. De esta forma, la creación de su objeto permite realizar la manipulación de la comunicación BLE, la escritura a dos archivos de la computadora y la configuración de los registros del detector de emociones. Los atributos de dicha clase corresponden a la dirección única universal (UUID, por sus siglas en inglés) de la característica del servicio que disponga el módulo HM10 para la recepción y transmisión de datos mediante el protocolo BLE, dos marcos de datos a los archivos respectivos con las características extraídas de las señales fisiológicas y las emociones clasificadas por el detector, así como los registros del detector de emociones de la Tabla 13. Los métodos de la clase permiten establecer la conexión con el módulo BLE del detector de emociones, conceder la transmisión y recepción de datos mediante el protocolo BLE y, a su vez, que los datos recibidos sean escritos a los archivos

correspondientes en la computadora. Asimismo, permite la transmisión de datos a los registros del detector de emociones, siguiendo el flujo de lectura de datos del módulo HM10 descrito en la Figura 48, mediante el uso de los botones en la GUI de la computadora.

#### 4.4.7.3 Selección secuencial hacia delante flotante

Tal como se mencionó en el Diseño en el nivel sistema, el algoritmo SFFS se ejecuta en su modalidad de envoltorio utilizando el clasificador LDA, por lo que el objetivo es hallar las características que maximicen la exactitud de la clasificación en cada etapa del esquema EMDC. Si bien en la sección 3.4.2 se estipula que debe definirse inicialmente un valor  $K$  que especifique la dimensión del espacio de características reducido, en este trabajo se plantea una modificación al algoritmo que permita al mismo decidir el número de características adecuado sin necesidad de fijarlo previamente. En la Figura 62 se presenta un diagrama que utiliza la misma simbología de la sección antes mencionada y expone las diferencias entre ambos algoritmos, siendo (A) el original y (B) la modificación propuesta. Como puede observarse, el funcionamiento del nuevo algoritmo propone realizar la adición de características hasta que la función objetivo  $J(\cdot)$  produzca un resultado menor al que se obtuvo en la iteración anterior, garantizando así un máximo desempeño con el subespacio seleccionado.

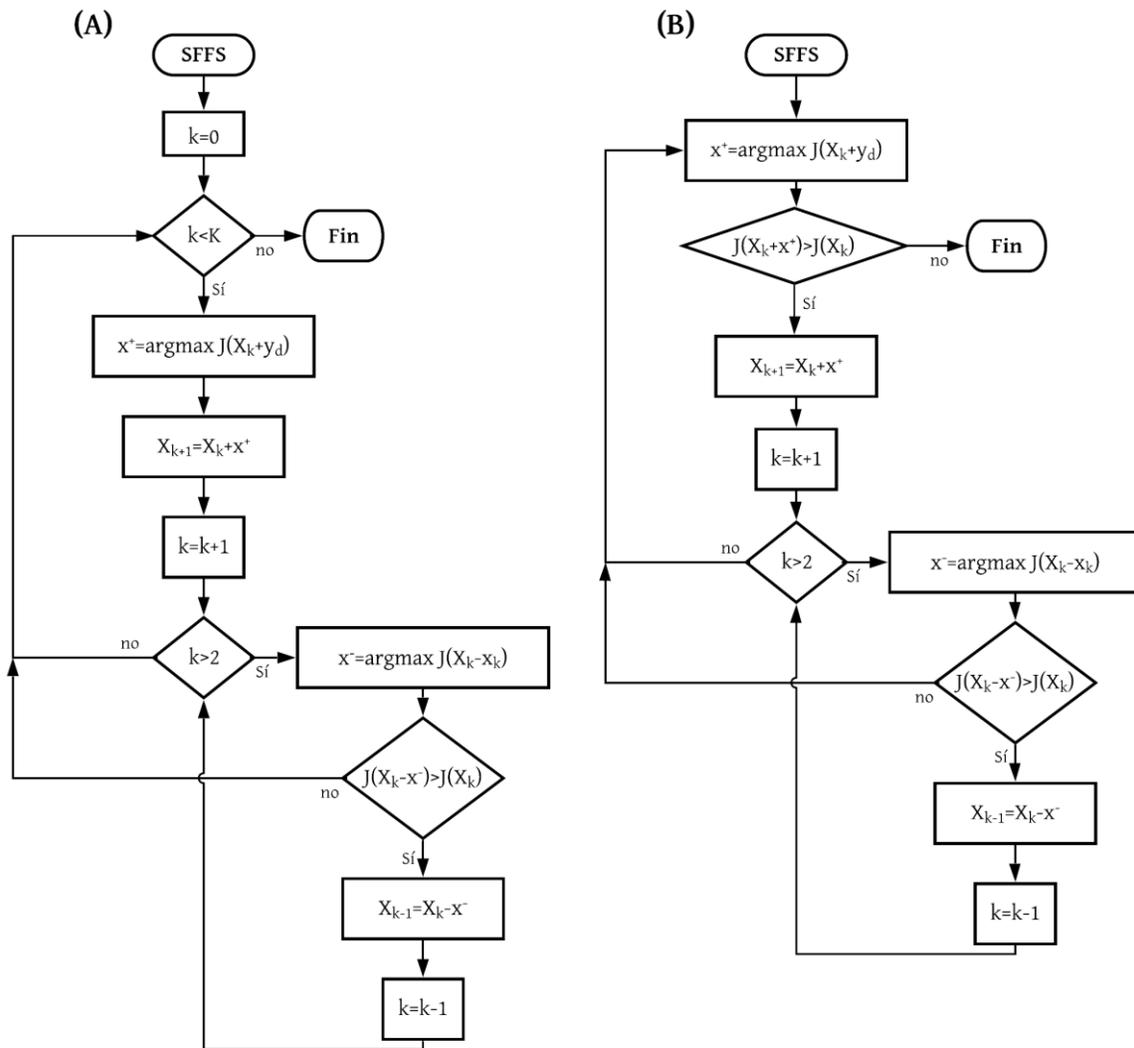


Figura 62. Modificación de SFFS

Asimismo, se elige utilizar dos tipos de validaciones para las clasificaciones realizadas dentro de la SFFS, la validación simple y la validación cruzada dejando uno fuera, mismas que se exponen en la sección 3.3.5. Cabe aclarar que para ambas se opta por hacer caso omiso del conjunto de validación, pues solo se está valorando la efectividad de las características seleccionadas y no el desempeño del clasificador en sí, pues este último ya está definido. La razón por la cual se elige implementar ambas como opciones es para proporcionar al usuario una modalidad de *entrenamiento rápido* y una modalidad de *entrenamiento completo*, cuya diferencia radica en la variabilidad de los resultados y el tiempo de ejecución, siendo este último definido por el total de vectores de características recolectados. Ambas modalidades son válidas, siendo la de *entrenamiento completo* la más efectiva, sin embargo, se optó por incorporar el *entrenamiento rápido* para evitar grandes periodos de espera durante las evaluaciones del dispositivo en trabajos futuros. Cabe resaltar que mientras menor sea el número de vectores de características con los que se cuente, peor desempeño tendrá este último tipo de entrenamiento.

Por último, para satisfacer los requerimientos del subsistema de configuración, la selección de características efectuada en cada SFFSn se guarda de forma codificada en 4 grupos de 8 bits donde, tal como se muestra en los registros “Caract PRV 1”, “Caract PRV 2”, “Caract EDA 1” y “Caract EDA 2” de la Tabla 14, un ‘1’ indica que la característica fue seleccionada y un ‘0’ lo contrario.

#### 4.4.7.4 Discriminante lineal de Fisher

Para el entrenamiento del algoritmo FLD se opta por considerar su versión de pLDA (véase la sección 3.4.3.1) con el fin de evitar errores en caso de que las matrices de covarianza  $S_W$  o  $S_B$  sean singulares (no invertibles). Asimismo, al ser clasificaciones binarias, el entrenamiento de cada FLDn produce una matriz de transformación  $W_n$  de dimensiones  $D_n \times K$ , donde  $D_n$  es la dimensión de las características seleccionadas por SFFSn y  $K$  la dimensión a la que se reduce el vector, siendo dicho valor igual a 1 dado que se busca una compresión a una dimensión como se describió en la sección 4.4.5.

Para satisfacer los requerimientos del subsistema de configuración, cada matriz de transformación  $W_n$  se guarda de forma codificada en un número entero de 32 bits con formato de punto fijo Q28 separado en 4 grupos de 8 bits, tal como se observa en los registros “Matriz FLDn  $W_n$ ” de la Tabla 14.

#### 4.4.7.5 Análisis discriminante lineal

El entrenamiento de cada algoritmo LDAn es realizado con vectores de características unidimensionales producto de la reducción del FLDn correspondiente. Con esto, se obtienen una serie de parámetros  $m_{k,n}$ ,  $\ln(\pi_k)_n$  y  $S_n$  cuyas dimensiones son  $k \times 1$ ,  $k \times 1$  y  $1 \times 1$ , respectivamente, donde  $k$  es la clase en cuestión. Cabe recordar que el esquema EMDC está conformado por clasificaciones binarias, por lo que  $k = 2$ .

Para satisfacer los requerimientos del subsistema de configuración, cada parámetro  $m_{k,n}$ ,  $\ln(\pi_k)_n$  y  $S_n$  se guarda de forma codificada en un número entero de 32 bits con formato de punto fijo Q28 separado en 4 grupos de 8 bits, tal como se visualiza en los registros “Vector  $m_k/\pi_k/S$ ” de la Tabla 14.

## 5 Implementación

La descripción del código de los programas para el sistema embebido puede encontrarse en el repositorio git público con nombre “Emotion\_Detector\_UNAM” cuyo direccionamiento también puede realizarse mediante el uso del código QR proporcionado en la Figura 63. Los programas ahí dispuestos yacen descritos en idioma inglés a fin de brindar de globalidad al mismo en caso de emplearse por fuentes externas.



Figura 63. Código QR del repositorio Emotion\_Detector\_UNAM

A continuación, los archivos manejados dentro del trabajo son expresados mediante comillas junto con su extensión respectiva “EjemArchivo.c”, las variables se denotan en cursivas *EjemVariable* y las funciones mediante negritas **EjemFuncion ()**.

### 5.1 Configuración

El archivo que compone al programa de este subsistema puede encontrarse en la raíz “CCS/Emotion\_Detector\_CPU1/Configuration” dentro del repositorio antes señalado, cuyo acceso se da en el código QR de la Figura 63.

Dentro del archivo “Configuration.c” se declaran todas las variables que corresponden a los diferentes registros de la Tabla 13. La escritura o lectura a cada uno de éstos se realiza mediante la función **VariablesMap ()**, cuya ejecución sigue los pasos descritos en el pseudocódigo de la Tabla 21. Cabe mencionar que, dentro de algunos registros de esta función, asociados a las características extraídas de las señales fisiológicas, se realiza una conversión de una aritmética en punto flotante simple a una en punto fijo Q16 en 32 bits para brindar así a éstas de un intervalo dinámico de  $-32768 \leq ID \leq 32767.9999847$  con una resolución de  $1.525879 \times 10^{-5}$ . Dicho Qi se elige tras visualizar los valores comunes que arrojan las características adquiridas con el sistema embebido. Y, a su vez, se realiza lo anterior con los parámetros asociados al subsistema de clasificación a una aritmética de punto fijo Q28 pero su justificación se da más adelante en la sección 5.8.6.

Asimismo, se conforman dos funciones de nombre **Write\_Emotion ()** y **Send\_Features ()** cuyo propósito es realizar la transmisión de los datos de la emoción clasificada y las características extraídas por el detector (esta última durante la modalidad de entrenamiento), a través del bloque UART que comunica los datos mediante BLE. A su vez, la función **Write\_Emotion ()** realiza la visualización de la emoción clasificada a través del LED RGB siguiendo la codificación de colores descrita en la Figura 45. Igualmente, la ejecución de las funciones anteriores sigue los pseudocódigos correspondientes de la Tabla 21.

Tabla 21. Pseudocódigo de las funciones del bloque de configuración.

Pseudocódigo	
<b>VariablesMap (Var_Addr, Data)</b>	
1.	Guardar los primeros 5 bits de <i>Var_Addr</i> en la variable <i>Address</i> .
2.	Si <i>Var_Addr</i> contiene el bit de lectura entonces <ol style="list-style-type: none"> <li>a. Transmitir por BLE el valor de <i>Address</i></li> <li>b. Según <i>Address</i> hacer               <ol style="list-style-type: none"> <li>i. Leer de memoria el valor de las variables respectivas al registro seleccionado por <i>Address</i> para concatenarlos en una sola trama de 8 bits, siguiendo su estructura de acuerdo con la Tabla 13, y transmitirlos por BLE.</li> </ol> </li> <li>c. Fin Según</li> </ol>
3.	SiNo <ol style="list-style-type: none"> <li>a. Según <i>Address</i> hacer</li> </ol>

<ul style="list-style-type: none"> <li>i. Leer cada uno de los campos de <i>Data</i>, correspondientes a su registro respectivo en la Tabla 13, de acuerdo con la dirección <i>Var_Addr</i> y guardarlos en variables con su mismo nombre. Algunos campos conllevan la acción sobre algún elemento del sistema por lo que durante su escritura se ejecuta la acción correspondiente.</li> <li>b. Fin Según</li> </ul>
<p>4. Fin Si</p>
<p><b>Send_Features (Number)</b></p>
<ul style="list-style-type: none"> <li>1. Si <i>AutoTx</i>=True entonces <ul style="list-style-type: none"> <li>a. Si <i>Modality</i>=0 (Prueba) entonces <ul style="list-style-type: none"> <li>i. Realizar una solicitud de escritura al registro <i>Caract Config</i> con el dato <i>Number</i> en la función <b>VariablesMap()</b>.</li> <li>ii. Realizar una solicitud de lectura a los registros de <i>Valor Caract</i> en la función <b>VariablesMap()</b> para transmitir sus valores a través de BLE.</li> </ul> </li> <li>b. Fin Si</li> </ul> </li> </ul>
<p>2. Fin Si</p>
<p><b>Write_Emotion ( )</b></p>
<ul style="list-style-type: none"> <li>1. Si <i>RGB_En</i>=True entonces <ul style="list-style-type: none"> <li>a. Habilitar LED RGB siguiendo la codificación de colores de la Figura 45.</li> </ul> </li> </ul>
<p>2. Fin Si</p>
<ul style="list-style-type: none"> <li>3. Si <i>AutoTx</i>=True entonces <ul style="list-style-type: none"> <li>a. Realizar una solicitud de lectura al registro de <i>Emoción</i> en la función <b>VariablesMap()</b> para transmitir su valor a través de BLE.</li> </ul> </li> </ul>
<p>4. Fin Si</p>

## 5.2 Comunicaciones

Con el objetivo de entablar la comunicación con el LED RGB, el sensor biométrico AS7026GG y el módulo HM10, se conforman tres funciones de nombre **Config\_Ports ( )**, **Config\_I2C ( )** y **Config\_SCIB ( )** encargadas de realizar la configuración de los módulos respectivos GPIO, I2C y SCI de la tarjeta TMS320F28379D, así como una de nombre **Config\_DMA ( )** orientada a la configuración de los canales DMA que permiten realizar el desplazamiento de los datos de PRV y aquellos involucrados con la interpolación por trazadores cúbicos, como se explica en la sección 4.4.4.5. A su vez, se atienden cuatro rutinas de interrupción (ISR, por sus siglas en inglés) respectivas a un canal DMA, cuyo origen se explica más adelante, el GPIO, los puertos I2C y los puertos UART. De las anteriores existe un orden de prioridad preestablecido por la tarjeta donde predomina el módulo GPIO, seguido del canal DMA, luego el módulo I2C y por último el módulo SCI.

Para comprender el funcionamiento de las ISR del DMA y el GPIO es suficiente su interpretación a través de los pseudocódigos descritos en la Tabla 22. En cambio, la ISR para el módulo I2C, cuyo pseudocódigo se encuentra en la misma Tabla 22, parte de la atención conmutada de las señales PPG y EDA recopiladas del sensor biométrico debido al funcionamiento del secuenciador visto en la sección 4.4.2.2, junto con el filtrado de ambas mediante los elementos del subsistema de acondicionamiento y el proceso de calibración de la señal obtenida mediante PPG siguiendo el algoritmo de la Figura 47. Cabe mencionar que posterior al filtrado de la señal se añade una condición que espera el transcurso de 6.25 s a fin de permitir el establecimiento de la misma tras el retraso ocasionado por el filtro FIR. Asimismo, la ISR del módulo SCI sigue el algoritmo estipulado en la Figura 48, en cuya lectura de caracteres recibidos se emplean funciones de la biblioteca “string.h”.

Cabe mencionar que, a causa del prolongado tiempo de ejecución de las características en frecuencias, por encima del periodo de muestreo de la señal EDA debido a su tiempo de ejecución alrededor de los 1.133 s, y dado que para ellas se requiere del vector completo de 2048 datos de la señal (64 s),

se opta por extender el arreglo de la señal SCR hasta 2208 datos (69 s) a fin de evitar corromper la información previamente adquirida con la nueva por adquirir. Esta extensión permite almacenar el excedente de datos recabados durante un periodo adicional de cinco segundos. La manera en cómo se logran reubicar los nuevos datos adquiridos dentro de los primeros 2048 considerables para las estimaciones en frecuencia de la SCR es a través de un canal DMA. Sin embargo, a falta de canales disponibles dentro de la tarjeta TMS320F28379D, se opta por reutilizar el tercer canal DMA, destinado a la señal PRV interpolada de nombre *CubSpl*, en cuya interrupción se realiza el intercambio de la transferencia por realizar a los datos del arreglo SCR.

Cabe mencionar que, tras pruebas realizadas con el LED RGB y el DSP seleccionado, se identificó que los GPIOs del DSP ya cuentan con una resistencia interna pequeña que permite limitar la corriente a través de los LEDs sin necesidad de utilizar las resistencias de  $82\ \Omega$  mencionadas en la sección 4.4.2, por lo que se omite su uso dentro del diseño final del detector de emociones.

Los archivos de las funciones descritas previamente pueden encontrarse en el módulo con raíz “CS/Emotion\_Detector\_CPU1/Communications” del repositorio disponible a través del código QR de la Figura 63. Asimismo, en el archivo yacen integradas dos bibliotecas denominadas “SCI\_UART.c” e “I2C.c”, las cuales ponen en marcha los protocolos de comunicación serial UART e I2C con los que dispone la tarjeta TMS320F28379D, así como una biblioteca “Biom\_AS7026GG.c” asociada al sensor biométrico y cuya explicación se brinda en la siguiente sección.

Tabla 22. Pseudocódigos de algunas ISR del subsistema de Comunicaciones.

<b>Pseudocódigo</b>
<b>Inter_DMACH3 ( )</b>
<ol style="list-style-type: none"> <li>1. Configuración del canal DMA número 3 para realizar el desplazamiento de 64 s de la señal interpolada <i>CubSpl</i>.</li> <li>2. Ejecución del canal DMA para comenzar el desplazamiento de los datos.</li> </ol>
<b>Inter_XINT2 ( )</b>
<ol style="list-style-type: none"> <li>1. Realizar una solicitud de lectura de los registros <i>FIFOL</i> y <i>FIFOH</i> a través del módulo I2C.</li> </ol>
<b>Inter_I2CA ( )</b>
<ol style="list-style-type: none"> <li>1. Si <i>Conmut</i>==1 (Señal PPG) entonces             <ol style="list-style-type: none"> <li>a. Leer la FIFO de recepción del módulo I2C y guardar el valor como <i>Biom1.GR_LED</i></li> <li>b. Si <i>Clb_Mode</i>==2 (Intervalo) hacer                 <ol style="list-style-type: none"> <li>i. Determinar el máximo y mínimo de la señal dentro de las variables <i>Clb_Max</i> <i>Clb_Min</i> comparando los datos previamente recolectados de la señal con el recién adquirido en <i>Biom1.GR_LED</i>.</li> </ol> </li> <li>c. Fin Si</li> <li>d. Si <i>Clb_Mode</i>==0 (Calibrado) hacer                 <ol style="list-style-type: none"> <li>i. Pasar el dato <i>Biom1.GR_LED</i> a través del filtro FIR a través de la función <b>FIR_PPG ( )</b> y guardarlo dentro de la variable <i>PPG</i>.</li> <li>ii. Si han transcurrido 6.25 s de la señal PPG entonces</li> <li>iii. Si la bandera <i>PRVShift</i> está habilitada entonces                     <ol style="list-style-type: none"> <li>1. Configurar y ejecutar los canales DMA para realizar el proceso de desplazamiento de datos sobre los arreglos <i>PRV_h</i> y <i>PRV_y</i>.</li> <li>2. Deshabilitar la bandera <i>PRVShift</i>.</li> </ol> </li> <li>iv. Fin Si</li> <li>v. Realizar la estimación de la PRV y sus características en tiempo mediante la función <b>PPI_Estimation ( )</b> enviándole como parámetros el valor de PPG y las direcciones de los vectores <i>PRV_y</i> y <i>PRV_hready</i>. Su valor de retorno se guarda en la variable <i>WDW_ready</i>.</li> </ol> </li> </ol> </li> </ol>

- vi. Fin Si
- e. Fin Si
- 2. SiNo (Señal EDA)
  - a. Leer la FIFO de recepción del módulo I2C y guardar el valor como *Biom1.int\_EDA*
  - b. Si *Clb\_Mode==1* (Amplitud) hacer
    - i. Determinar el máximo y mínimo de la señal dentro de las variables *Clb\_Max Clb\_Min* comparando los datos previamente recolectados de la señal con el recién adquirido en *Biom1.int\_EDA*.
  - c. Fin Si
  - d. Si *Clb\_Mode==0* (Calibrado) y *Clb\_Peak==1* (Se detecta el primer pico) hacer
    - i. Realizar el remuestreo de la señal EDA a 32 Hz al ejecutar los pasos siguientes sólo cuando se adquiriera el cuarto valor del dato recabado en *Biom1.int\_EDA* a 128 Hz.
    - ii. Pasar el dato *Biom1.EDA* a través del filtro FIR a través de la función **FIR\_EDA ( )** y guardarlo dentro del elemento *i* del arreglo *SCR*.
    - iii. Si  $i > 2048$  entonces
      - 1. Incrementar el valor de la variable *k* para determinar la cantidad de nuevos datos adquiridos en el arreglo *SCR* mientras yace ejecutándose la estimación en frecuencia.
    - iv. Fin Si
    - v. Si  $i = 2048$  entonces
      - 1. Incrementar el valor de la bandera *sum\_flg*.
    - vi. Fin Si
    - vii. Si la bandera *SCRShift* está habilitada entonces
      - 1. Configurar y ejecutar los canales DMA para realizar el desplazamiento de los *k* nuevos datos adquiridos en el arreglo *SCR*. Es decir, los datos adquiridos tras completarse la ventana de 64 s de la señal *SCR*.
      - 2. Deshabilitar la bandera *SCRShift*.
    - viii. Fin Si
  - e. Fin Si
- 3. Fin Si
- 4. Ejecutar el proceso de calibración descrito por el algoritmo de la Figura 47.
- 5. Conmutar el valor de *Conmut*.

### 5.2.1 Sensor biométrico AS7026GG

Para el diseño de la tarjeta PCB se emplea el software Altium Designer con el cual se procura un tamaño compacto de 16x17 mm con una estructura de doble capa para acomodar todos los



Figura 64. PCB del sensor biométrico AS7026GG

componentes necesarios ilustrados en el esquemático de la Figura 54, resultando en el diseño de la Figura 64. La impresión de la tarjeta fue realizada mediante la compañía BitShakeMx y los componentes fueron soldados a mano. Cabe mencionar que en el diseño se realizó el remplazo de las resistencias pull up para el bus I2C por dos resistencias de  $820\Omega$ , pues las anteriores no garantizan el tiempo de levantamiento inferior adecuado para el protocolo bajo una tasa de 400 kbps. Además, se aplica una capa de aislante (barniz) a los pines GSR que sobresalen de la tarjeta a fin de evitar el contacto con la piel una vez que se coloca el dedo encima del sensor, ya que con ello se realiza la medición de la señal obtenida mediante PPG.

Con respecto a la configuración del sensor, se constituye la función **Biom\_Config** ( ) encargada de realizar la configuración de cada uno de los componentes electrónicos del OFE, los LEDs, los fotodiodos, la TIA, el preamplificador, la EAFE, el ADC, la FIFO, los puertos GPIO y el secuenciador, de acuerdo con los parámetros estipulados en la sección 4.4.2.2. A su vez, se realizó la implementación de cuatro funciones de nombre **AS7026GG\_Start** ( ) y **AS7026GG\_Stop** ( ) que arrancan o detienen el funcionamiento del secuenciador, junto con **AS7026GG\_ADC\_Channel** ( ) y **Biom\_Calibration** ( ) que son utilizadas dentro del subsistema de Comunicaciones al momento de efectuarse el proceso de calibración. El funcionamiento de estas últimas puede observarse en los pseudocódigos de la Tabla 23 y su implementación se encuentra en el archivo “Biom\_AS7026GG.c” dentro del repositorio indicado para el subsistema de comunicaciones.

Tabla 23. Pseudocódigo de algunas funciones utilizadas con el sensor AS7026GG.

<b>Pseudocódigo</b>	
<b>AS7026GG_ADC_Channel (Channel)</b>	
1.	Si <i>Canal</i> =0 entonces <ol style="list-style-type: none"> <li>a. Seleccionar los canales OFE1 After Pregain y Electriclcal Front End un cero al atributo <i>seq_start</i> del registro <i>SEQ_START</i> del sensor para detener el funcionamiento del secuenciador del sensor.</li> <li>b. Asignar el valor de <i>Offset_Curr</i> al registro <i>PDOFFX_LEDON</i> para ajustar el offset de corriente suministrado al LED.</li> <li>c. Escribir un uno al atributo <i>seq_start</i> del registro <i>SEQ_START</i> del sensor para detener el funcionamiento del secuenciador del sensor.</li> </ol>
2.	SiNo Si <i>Offset_Curr</i> <39 entonces <ol style="list-style-type: none"> <li>a. Escribir un cero al atributo <i>seq_start</i> del registro <i>SEQ_START</i> del sensor para detener el funcionamiento del secuenciador del sensor.</li> <li>b. Escribir un cero al atributo <i>led1_en</i> del registro <i>SEQ_CFG</i> del sensor para deshabilitar la salida del LED1.</li> <li>c. Asignar el valor de <i>Offset_Curr</i> a los registros <i>LED1_CURRL</i> y <i>LED1_CURRH</i> para ajustar el nivel de corriente suministrado al LED.</li> <li>d. Escribir un uno al atributo <i>led1_en</i> del registro <i>SEQ_CFG</i> del sensor para habilitar la salida del LED1.</li> <li>e. Escribir un uno al atributo <i>seq_start</i> del registro <i>SEQ_START</i> del sensor para detener el funcionamiento del secuenciador del sensor.</li> </ol> <p style="text-align: center;">Fin Si</p>
<b>Biom_Calibration (Mode, Offset_Curr)</b>	
1.	Si <i>Modo</i> =1 y <i>Offset_Curr</i> <256 entonces <ol style="list-style-type: none"> <li>a. Escribir un cero al atributo <i>seq_start</i> del registro <i>SEQ_START</i> del sensor para detener el funcionamiento del secuenciador del sensor.</li> <li>b. Asignar el valor de <i>Offset_Curr</i> al registro <i>PDOFFX_LEDON</i> para ajustar el offset de corriente suministrado al LED.</li> </ol>

- c. Escribir un uno al atributo *seq\_start* del registro *SEQ\_START* del sensor para detener el funcionamiento del secuenciador del sensor.
2. SiNo Si *Offset\_Curr*<39 entonces
  - a. Escribir un cero al atributo *seq\_start* del registro *SEQ\_START* del sensor para detener el funcionamiento del secuenciador del sensor.
  - b. Escribir un cero al atributo *led1\_en* del registro *SEQ\_CFG* del sensor para deshabilitar la salida del LED1.
  - c. Asignar el valor de *Offset\_Curr* a los registros *LED1\_CURRL* y *LED1\_CURRH* para ajustar el nivel de corriente suministrado al LED.
  - d. Escribir un uno al atributo *led1\_en* del registro *SEQ\_CFG* del sensor para habilitar la salida del LED1.
  - e. Escribir un uno al atributo *seq\_start* del registro *SEQ\_START* del sensor para detener el funcionamiento del secuenciador del sensor.
3. Fin Si

### 5.3 Acondicionamiento de la señal

Tomando en consideración los requerimientos y funcionalidades establecidos en el Diseño en el nivel sistema y posteriormente refinados en el Diseño de detalle, se construye la etapa de acondicionamiento enfocándose únicamente en el filtrado de las señales por medio de filtros digitales FIR, pues fue seleccionado un sensor biométrico que cuenta con módulos de amplificación y digitalización propios.

Ambos filtros son generados y evaluados mediante MATLAB tomando en consideración las frecuencias de muestreo y corte indicadas en la sección 4.4.2 y 4.4.3, respectivamente. Cabe resaltar que la herramienta utilizada para el diseño de filtros FIR considera que las frecuencias de corte son aquellas cuya ganancia normalizada es de -6dB, por lo que se modificaron las frecuencias de diseño con el objetivo de tener una ganancia de aproximadamente -3dB en las frecuencias de corte estipuladas anteriormente, resultando en 2.16 Hz en el caso de la EDA y 0.4 Hz y 3.58 Hz para la señal obtenida a partir de la PPG.

El orden de los filtros se elige de 200 en ambas señales, pues la memoria requerida es ligera y sus bandas de transición se consideran adecuadas para la aplicación en cuestión. Dicho orden implica un

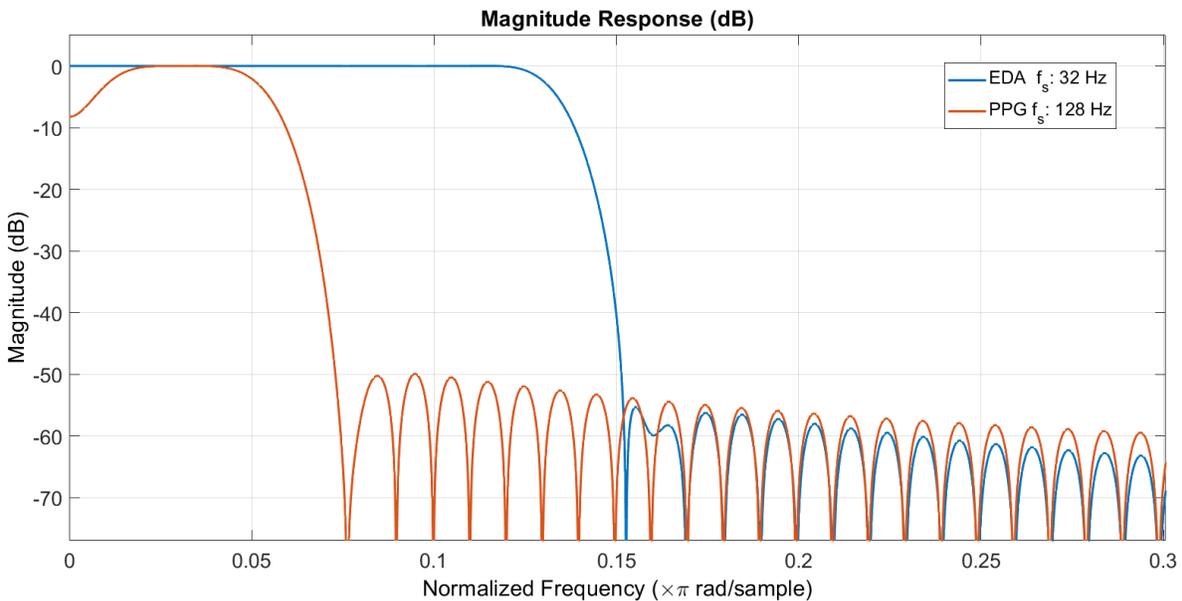


Figura 65. Respuesta en frecuencia de filtros FIR

retraso de 100 muestras en el filtrado de cada una de las señales, el cual se traduce a aproximadamente 0.78 s para la señal obtenida mediante PPG y 3.13 s para la EDA. Sin embargo, dicho retraso es insignificante dado el tamaño de las ventanas señalado en la sección 4.4.4.5.

Al emplearse una ventana Hamming, la banda suprimida de cada filtro se mantiene con una atenuación por encima de los 50 dB, tal como se ilustra en la Figura 65. Dicha gráfica se obtuvo mediante la herramienta de análisis FVTool, la cual permite plasmar la respuesta en frecuencia de los filtros diseñados. Cabe resaltar que la frecuencia mostrada es normalizada para obtener un intervalo entre 0 y 1, donde 1 es la mitad de la frecuencia de muestreo.

Inicialmente los coeficientes de cada filtro son obtenidos en un formato de punto flotante de doble precisión, sin embargo, para hacer uso de éstos en el procesador digital de señales y con el objetivo de mantener un intervalo adecuado de memoria, son transformados a un formato de punto fijo Q15 en 16 bits. Dicho Qi fue elegido tras un análisis del mínimo y máximo coeficiente que es necesario representar. Finalmente, cada valor generado es guardado en un archivo de cabecera para su posterior implementación.

El programa encargado de realizar las funciones antes descritas puede ser encontrado en la raíz “Matlab/FIR.m” del repositorio “Emotion\_Detector\_UNAM” disponible a través del código QR de la Figura 63.

Tal como se mencionó en la sección 4.4.3, los filtros FIR son implementados en su forma directa y de fase lineal, por lo que se desarrolla un algoritmo utilizando una mezcla de código en lenguaje C y ASM que implementa las operaciones necesarias para filtrar las señales fisiológicas. Para ello se utiliza un buffer de tamaño  $n + 1$ , donde  $n$  es el orden del filtro, que guarda dentro de su primera posición la muestra  $x$  de la señal en cuestión. Posteriormente, se realiza una multiplicación y acumulación de cada elemento del buffer con cada coeficiente del filtro. Para cumplir dicha tarea se implementa un doble ciclo con el objetivo de aprovechar la simetría de los coeficientes. Finalmente, se realiza un corrimiento del buffer de modo que quede el primer espacio vacío a la espera de un nuevo dato de entrada. Dicho algoritmo es ejecutado por medio de la función **firFilter** ( ), la cual recibirá los parámetros correspondientes a cada señal fisiológica por medio de las funciones **FIR\_PPG** ( ) o **FIR\_EDA** ( ).

El código del subsistema de acondicionamiento puede ser encontrado dentro de la raíz “CCS/Emotion\_Detector\_CPU1/Conditioning” del repositorio antes señalado. A continuación, se muestra una descripción paso a paso de las funciones más relevantes de los archivos “FIR\_filter.c” y “FIR.asm”.

Tabla 24. Pseudocódigo de función realizada en C para el filtrado de las señales fisiológicas.

<b>Pseudocódigo</b>	
<b>FIR_PPG (x) / FIR_EDA (x)</b>	
1.	Creación de variable local $y$ .
2.	Llamada de función <b>firFilter</b> ( ) dándole como argumento la muestra obtenida $x$ , el número de coeficientes del filtro, la dirección de los coeficientes y la dirección del buffer. Retorna la señal filtrada y lo guarda en $y$ .
3.	Se retorna $y$ .

Tabla 25. Pseudocódigo de la función realizada en ASM para ejecutar el filtro FIR en su forma directa y de fase lineal.

<b>Pseudocódigo</b>	
<b>firFilter (sample, Num_coef, coef_Adress, buffer_Address)</b>	
1.	Utilizando la dirección indicada por el <i>buffer_Address</i> se apunta a un buffer de datos por medio de dos registros.
2.	Se guarda el valor de <i>sample</i> en la primera localidad del buffer.
3.	Para $i = \text{Num\_coef} - 1$ hasta $i = 0$ hacer:
a.	Se realiza una multiplicación y acumulación de cada valor de la primera mitad del buffer con los coeficientes del filtro (PPG o EDA).
4.	Fin Para
5.	Para $i = \text{Num\_coef} - 2$ hasta $i = 0$ hacer:
a.	Se realiza una multiplicación y acumulación de cada valor de la segunda mitad del buffer con los coeficientes invertidos del filtro (PPG o EDA).
6.	Fin Para
7.	Se almacena el resultado de los dos ciclos anteriores.
8.	Se realiza un corrimiento de 15 bits a la derecha sobre el resultado para obtener un número entero de 16 bits en formato Q15.
9.	Se realiza un corrimiento de todos los valores en el buffer para dejar un espacio vacío al inicio de éste.
10.	Se retorna el resultado.

## 5.4 Extracción de características

Se realiza la subdivisión de la extracción de características en distintos rubros, que por cuestiones de eficiencia y estructura del código, se han dividido de la siguiente manera

### 5.4.1 Generación de PRV

Los archivos que componen a los programas para la generación de PRV y de la estimación de características pueden encontrarse en la raíz “CCS/Emotion\_Detector\_CPU1/Time\_Extraction” del repositorio “Emotion\_Detector\_UNAM” cuyo acceso se da en el código QR de la Figura 63.

Primeramente, la función **PPI\_Estimation()** recibe las muestras de la señal de PPG conforme se adquieren del sensor biométrico y, tal como es descrito en la sección 4.4.4.1, se comparan dos muestras sucesivas empleando la ecuación (62) con el propósito de identificar el comienzo de un flanco ascendente y así, el pico sistólico correspondiente. Una vez que se han detectado dos picos sistólicos consecutivos es posible calcular el IPP, el cual es almacenado en memoria para ser posteriormente empleado en las funciones dirigidas a la obtención de características.

### 5.4.2 Dominio del tiempo y gráfica de Poincaré de PRV

Observando tanto la Tabla 19 de la sección 4.4.4.1 como las ecuaciones (66) y (67), es claro que para obtener el valor de las características es necesario contar con el número total de IPP, es decir  $N$ , el cual se conoce tras analizarse el segmento completo. Sin embargo, no es necesario esperar la adquisición de todo el segmento para comenzar los cálculos, ya que es posible estimar parcialmente las características de PRV conforme se obtienen los IPP. Para esto, cuando un nuevo intervalo es calculado por la función **PPI\_Estimation()**, las funciones **Poincare\_Graph()** y **Time\_Features()** son empleadas para agregarlo a las sumas correspondientes. Una vez concluido el segmento de 64 s, dichas funciones estiman el valor final de las características empleando el número de IPP registrados y los intervalos acumulados previamente. Cabe resaltar que ciertas características requieren de dos IPP adyacentes para comenzar su cálculo, como es el caso de las desviaciones estándar de la gráfica de

Poincaré, la RMSSD y NN50, por lo que las funciones **Poincare\_Graph()** y **Time\_Features()** son empleadas a partir de que se han obtenido los dos primeros IPP válidos.

Debido a la diferencia de tamaño que existe entre los segmentos de PRV y SCR, se ha optado por transmitir las características de ambas señales cada 64 segundos, considerando para PRV los últimos 256 s registrados, lo cual se explica a detalle en la sección 4.4.4.5. En la Figura 66 se ilustra el mecanismo empleado para solventar la adquisición de las características de la PRV de forma simultánea con las de EDA.

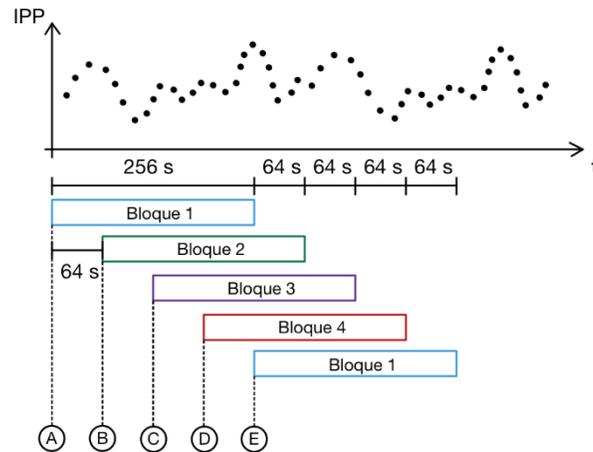


Figura 66. Mecanismo de adquisición de características de PRV.

Los bloques ilustrados en la imagen representan la estimación de características correspondientes a 256 segundos de la PRV, donde cada uno emplea los IPP calculados en el periodo de tiempo que le corresponde. Durante la etapa *A* se adquieren los primeros intervalos, los cuales son acumulados únicamente por el Bloque 1, mientras que la etapa *B* comienza sus estimaciones después de transcurrir los primeros 64 s, donde los IPP estimados en dicho periodo son destinados tanto al Bloque 1 como al Bloque 2. Al transcurrir nuevamente 64 s, comienza el Bloque 3 y, al alcanzar la etapa *D*, inicia el Bloque 4. En la etapa *E* han transcurrido los primeros 256 s, donde son estimadas las características derivadas de los IPP correspondientes al Bloque 1 y, a su vez, dicho bloque reinicia sus valores y comienza desde cero a acumular los IPP. A partir de esta etapa (256 s), dicho proceso se repite, por lo que cada 64 segundos se obtienen las características de cada bloque, tal como se ilustra en la Figura 66.

El desplazamiento de datos mediante los canales DMA, explicado en la sección 4.4.4.5, podría ocasionar la pérdida de datos si un IPP es obtenido al mismo tiempo que se realiza el corrimiento, por lo que se ha implementado una forma de evitar dicha situación. En cuanto comienza el desplazamiento de datos, la bandera RUNSTS del registro de control del DMA es activada, por lo que si en ese periodo de tiempo es calculado un IPP, éste se guarda en un buffer temporal, en vez de agregarlo al conjunto de datos que en dicho momento están siendo desplazados. Una vez finalizado el corrimiento, la misma bandera es limpiada, lo cual indica al código la necesidad de transferir los IPP almacenados en el buffer al arreglo correspondiente.

El pseudocódigo que describe tanto la generación de PRV como el procedimiento anterior se desglosa en la Tabla 26.

Tabla 26. Pseudocódigo de las funciones para generar la PRV y calcular sus características

<b>Pseudocódigo</b>	
<b>PPI_Estimation(x_PPG, &amp;X, &amp;Y)</b>	
1.	Si se completaron los 256 segundos del segmento ( $sum\_flg > 3$ ) entonces

- a. Se calculan las características correspondientes a la gráfica de Poincaré mediante la función **Poincare\_Graph()**.
- b. Se calculan las características correspondientes al dominio del tiempo y métodos estadísticos mediante la función **Time\_Features()**.
- c. Retorna 0
2. SiNo
  - a. Si la muestra actual es mayor que la muestra previa ( $PPG[n] > PPG[n-1]$ ) entonces
    - i. El contador de muestras ascendentes (*subs\_up\_steps*) aumenta en 1.
  - b. Sino
    - i. Si se superó el umbral ( $subs\_up\_steps > umbral$ ) entonces
      1. El número de picos detectados (*detected\_peaks*) aumenta en 1
    - ii. Fin Si
    - iii. Si el número de picos detectados es mayor a uno ( $detected\_peaks > 1$ ) entonces
      1. Si el IPP calculado se encuentra dentro del intervalo aceptado de acuerdo con el promedio estimado hasta el momento entonces
        - a. Si el DMA se encuentra realizando el desplazamiento entonces
          - i. El IPP es almacenado en el buffer temporal.
        - b. Si el DMA finalizó el desplazamiento entonces
          - i. Los IPP almacenados en el buffer temporal son transferidos a los arreglos X e Y.
        - c. Si el DMA no se encuentra activo
          - i. Se guarda el intervalo en los arreglos X e Y para su posterior uso en el cálculo de las características en el dominio de la frecuencia.
        - d. Fin Si
        - e. Se calcula el promedio de los intervalos adquiridos hasta el momento.
      2. SiNo
        - a. Se considera que se perdió un pico o que se detectó uno falso.
      3. Fin Si
      4. Si ya se estimó más de un IPP (*número de IPP estimados* > 0) entonces
        - a. Se realizan las sumas correspondientes a las características de la gráfica de Poincaré mediante **Poincare\_Graph()** con el IPP calculado.
        - b. Se realizan las sumas correspondientes a las características del dominio del tiempo mediante **Time\_Features()** con el IPP calculado.
      5. Fin Si
    - iv. Fin Si
  - c. Fin Si
3. Fin Si
4. Si ya se detectó un pico ( $detected\_peaks > 0$ )
  - a. El contador de muestras *indx\_dist* se aumenta en uno
5. Fin Si
6. Retorna variable booleana *WR*
7. Fin

**PFeat\_PreEstimation(&PPI, &numPPI, &sumSD1, &sumSD2, &meanSD1, &meanSD2)**

1. Se calcula la proyección de los intervalos del Bloque empleando las ecuaciones (64) y (65).
2. El resultado de la proyección se acumula en las variables de suma que se emplearán en el cálculo de *SD1* y *SD2* cuando finalice el Bloque.

3. Aumenta en uno el número de IPP del Bloque ( $numPPI$ ).
<b>PFeat_Estimation(&amp;numPPI, &amp;sumSD1, &amp;sumSD2, &amp;meanSD1, &amp;meanSD2 )</b>
1. Se estiman las características de la gráfica de Poincaré empleando las ecuaciones (66) y (67). ( $SD1, SD2, SD1\_SD1$ ).
<b>Poincare_Graph(&amp;PPI, sum_flg)</b>
1. Si ya transcurrieron 256 segundos ( $sum\_flg > 3$ ) entonces <ul style="list-style-type: none"> <li>a. Si <math>cont=0</math> entonces <ul style="list-style-type: none"> <li>i. Se realiza el procedimiento descrito en <b>PFeat_Estimation( )</b> para el Bloque 1.</li> <li>ii. <math>cont</math> se aumenta en 1.</li> </ul> </li> <li>b. Si <math>cont=1</math> entonces <ul style="list-style-type: none"> <li>i. Se realiza el procedimiento descrito en <b>PFeat_Estimation( )</b> para el Bloque 2.</li> <li>ii. <math>cont</math> se aumenta en 1.</li> </ul> </li> <li>c. Si <math>cont=2</math> entonces <ul style="list-style-type: none"> <li>i. Se realiza el procedimiento descrito en <b>PFeat_Estimation( )</b> para el Bloque 3.</li> <li>ii. <math>cont</math> se aumenta en 1.</li> </ul> </li> <li>d. Si <math>cont=3</math> entonces <ul style="list-style-type: none"> <li>i. Se realiza el procedimiento descrito en <b>PFeat_Estimation( )</b> para el Bloque 4.</li> <li>ii. <math>cont = 0</math>.</li> </ul> </li> <li>e. Fin Si</li> </ul>
2. SiNo <ul style="list-style-type: none"> <li>a. Se realiza el procedimiento descrito en <b>PFeat_PreEstimation( )</b> para el Bloque 1.</li> <li>b. Si ya transcurrieron los primeros 64 segundos (<math>sum\_flg &gt; 0</math>) entonces <ul style="list-style-type: none"> <li>i. Se realiza el procedimiento descrito en <b>PFeat_PreEstimation( )</b> para el Bloque 2.</li> </ul> </li> <li>c. Fin Si</li> <li>d. Si ya transcurrieron los primeros 128 segundos (<math>sum\_flg &gt; 1</math>) entonces <ul style="list-style-type: none"> <li>i. Se realiza el procedimiento descrito en <b>PFeat_PreEstimation( )</b> para el Bloque 3.</li> </ul> </li> <li>e. Fin Si</li> <li>f. Si ya transcurrieron los primeros 192 segundos (<math>sum\_flg &gt; 2</math>) entonces <ul style="list-style-type: none"> <li>i. Se realiza el procedimiento descrito en <b>PFeat_PreEstimation( )</b> para el Bloque 4.</li> </ul> </li> </ul>
3. Fin Si
<b>TFeat_PreEstimation( &amp;PPI, &amp;numPPI, &amp;preNN50, &amp;preSDNN, &amp;preRMSSD, &amp;sumPPI )</b>
1. Si los intervalos adyacentes $PPI[n]$ y $PPI[n - 1]$ difieren por más de 50ms entonces <ul style="list-style-type: none"> <li>a. La característica <math>NN50</math> se aumenta en uno.</li> </ul>
2. El intervalo recibido se acumula a las variables de suma que se emplearán en el cálculo de $RMSSD, SDNN$ y $meanPPI$ cuando finalice el segmento.
3. Aumenta en uno el número de PPI ( $numPPI$ ).
<b>TFeat_Estimation( &amp;PPI, &amp;numPPI, &amp;preNN50, &amp;preSDNN, &amp;preRMSSD, &amp;sumPPI)</b>
1. Se estiman las características en el dominio del tiempo $meanPPI, RMSSD, SDNN$ y $pNN50$ empleando las ecuaciones (2), (3), (4), y (31), respectivamente.
<b>Time_Features(&amp;PPI, sum_flg)</b>
1. Si ya transcurrieron 256 segundos ( $sum\_flg > 3$ ) entonces <ul style="list-style-type: none"> <li>a. Si <math>cont=0</math> entonces <ul style="list-style-type: none"> <li>i. Se realiza el procedimiento descrito en <b>TFeat_Estimation( )</b> para el Bloque 1.</li> <li>ii. <math>cont</math> se aumenta en 1.</li> </ul> </li> </ul>

- b. Si  $cont=1$  entonces
    - i. Se realiza el procedimiento descrito en **TFeat\_Estimation()** para el Bloque 2.
    - ii.  $cont$  se aumenta en 1.
  - c. Si  $cont=2$  entonces
    - iii. Se realiza el procedimiento descrito en **TFeat\_Estimation()** para el Bloque 3.
    - iv.  $cont$  se aumenta en 1.
  - d. Si  $cont=3$  entonces
    - iii. Se realiza el procedimiento descrito en **TFeat\_Estimation()** para el Bloque 4.
    - iv.  $cont = 0$ .
  - e. Fin Si
2. SiNo
- a. Se realiza el procedimiento descrito en **TFeat\_PreEstimation()** para el Bloque 1.
  - b. Si ya transcurrieron los primeros 64 segundos ( $sum\_flg > 0$ ) entonces
    - ii. Se realiza el procedimiento descrito en **TFeat\_PreEstimation()** para el Bloque 2.
  - c. Fin Si
  - d. Si ya transcurrieron los primeros 128 segundos ( $sum\_flg > 1$ ) entonces
    - ii. Se realiza el procedimiento descrito en **TFeat\_PreEstimation()** para el Bloque 3.
  - e. Fin Si
  - f. Si ya transcurrieron los primeros 192 segundos ( $sum\_flg > 2$ ) entonces
    - ii. Se realiza el procedimiento descrito en **TFeat\_PreEstimation()** para el Bloque 4.
3. Fin Si

### 5.4.3 Obtención de SCR

Primeramente, la función **SCR\_Extraction()** recibe las muestras de la señal de EDA conforme se adquieren del subsistema de acondicionamiento y, tal como es descrito en la sección 4.4.4.2, se realiza la diferenciación y posterior convolución con una ventana Bartlett de 32 puntos. Este procedimiento es descrito mediante los pseudocódigos mostrados en la Tabla 27 y la Tabla 28.

Tabla 27. Pseudocódigo para la obtención de la componente SCR de la señal EDA.

Pseudocódigo
<b>SCR_Extraction(x_EDA, rdy)</b>
1. Se realiza la diferenciación empleando la ecuación (62) y el resultado es almacenado en $x\_Diff$ .
2. La función <b>convol()</b> es llamada dándole como argumento el resultado $x\_Diff$ , el número de coeficientes de la ventana Bartlett, la dirección de los coeficientes y la dirección del buffer. Retorna el resultado de la convolución y lo guarda en $x\_SCR$ .
3. Se envían las muestras $x\_SCR$ a la función <b>SCR_Detection()</b> .

Tabla 28. Pseudocódigo de la función realizada en ASM para ejecutar la convolución de la señal diferenciada con la ventana Bartlett.

Pseudocódigo
<b>convol(x_Diff, Num_window, coef_Address, buffer_Address)</b>
1. Utilizando la dirección indicada por el $buffer\_Address$ se apunta a un buffer de datos por medio de dos registros.
2. Se guarda el valor de $x\_Diff$ en la primera localidad del buffer.
3. Para $i=Num\_window-1$ hasta $i=0$ hacer: <ul style="list-style-type: none"> <li>a. Se realiza una multiplicación y acumulación del buffer con los coeficientes de la ventana.</li> </ul>
4. Fin Para

5. Se almacena el resultado del ciclo anterior.
6. Se realiza un corrimiento de 15 bits a la derecha sobre el resultado para obtener un número entero de 16 bits en formato Q14.
7. Se realiza un corrimiento de todos los valores en el buffer para dejar un espacio vacío al inicio de éste.
8. Se retorna el resultado.

#### 5.4.4 Dominio del tiempo de SCR

Los archivos que componen a los programas para la obtención de las características de EDA pueden encontrarse en la raíz “CCS/Emotion\_Detector\_CPU1/Time\_Extraction” del repositorio “Emotion\_Detector\_UNAM” cuyo acceso se da en el código QR de la Figura 63.

Una vez obtenida la señal de SCR, la función **SCR\_Detection()** compara dos muestras sucesivas con el propósito de identificar los cruces por cero y así, el comienzo de un evento. Una vez detectado el SCR, tanto su amplitud como tiempo de levantamiento son almacenados en memoria para ser posteriormente empleados en las funciones dirigidas a la obtención de características.

A diferencia de las características extraídas a partir de la señal de PPG, no es posible estimar la desviación estándar ni las medias correspondientes a los SCR conforme se adquieren, debido a que, tal como es especificado en la sección 3.2.3.2, los SCR con amplitudes menores al 10% de la amplitud máxima del segmento son descartados. De lo anterior se deduce que es preciso analizar el segmento completo para determinar la amplitud máxima y con base en ello, descartar las amplitudes que no cumplan con dicho criterio. Una vez realizado lo anterior, se emplean las funciones **Amplitude\_Estimation()** y **Rise\_Time\_Mean()** para estimar las características estadísticas del dominio del tiempo.

Para el cálculo del tiempo de levantamiento es necesario almacenar las muestras de la señal desde que comienza el SCR hasta que la amplitud es identificada. Esto es debido a que, tal como se define en la sección 3.2.1.1, éste es el lapso que transcurre desde que la señal se encuentra en el 10% de su valor máximo, hasta el 90% del mismo. Por lo tanto, una vez identificada la amplitud, es preciso recorrer las muestras para encontrar tanto la muestra correspondiente al 10% como el 90% de la amplitud y determinar el tiempo que transcurrió entre dichas muestras.

Como es descrito en la sección 3.2.1.1, la estimación de cuantiles requiere ordenar de forma ascendente los elementos, que en el caso de EDA corresponden a las amplitudes de los SCR. Para ello, se emplea la función **MergeSort()**, la cual implementa un algoritmo dirigido al ordenamiento de datos de manera eficiente. Una vez obtenido el arreglo de amplitudes, se estiman los diversos cuantiles con la función **Quantiles()**. El pseudocódigo que describe el procedimiento tanto de la detección de SCR como el cálculo de las características es desglosado en la Tabla 29.

Tabla 29. Pseudocódigo para la obtención de las características provenientes de los SCR

<b>Pseudocódigo</b>	
<b>SCR_Detection(x_SCR, rdy)</b>	
1.	Si se presenta un cruce por cero de negativo a positivo entonces
a.	Se considera que comenzó un SCR y se almacenan las muestras en el arreglo <i>det_scr</i> .
b.	Si se encontró el máximo de la señal entonces
i.	Se almacena la amplitud en el arreglo <i>Amp</i> .
ii.	Se calcula en tiempo de levantamiento con <b>Rise_Time_Mean()</b> y se almacena el valor en el arreglo <i>mTL</i> .
c.	Fin Si
2.	Fin Si
3.	Si se completaron los 64 segundos del segmento ( <i>rdy = 1</i> ) entonces

<ul style="list-style-type: none"> <li>a. Se ordenan de forma ascendente las amplitudes y los tiempos de levantamiento con la función <b>MergeSort()</b>.</li> <li>b. Se descartan las amplitudes y tiempos de levantamiento correspondientes a los SCR que no cumplen con el criterio del 10%.</li> <li>c. Se calculan las características correspondientes a las amplitudes de los SCR válidos mediante la función <b>Amplitude_Estimations()</b>.</li> <li>d. Se calculan las características finales correspondientes al tiempo de levantamiento mediante la función <b>Rise_Time_Mean()</b>.</li> <li>e. Se calculan los cuantiles con la función <b>Quantiles()</b>.</li> </ul>
<p>4. Sino</p> <p style="text-align: center;"><b>Amplitude_Estimations(Amp, numSCR, rdy)</b></p> <ul style="list-style-type: none"> <li>1. Se calcula la desviación estándar de las amplitudes de los SCR.</li> <li>2. Se calcula la media de las amplitudes.</li> </ul>
<p style="text-align: center;"><b>Rise_Time_Mean(&amp;scr, Amp, numSCR, rdy)</b></p> <ul style="list-style-type: none"> <li>1. Se calcula el 10% y el 90% de la amplitud del SCR.</li> <li>2. Si se completaron los 64 segundos del segmento (<math>rdy = 1</math>) entonces <ul style="list-style-type: none"> <li>a. Se calcula la media del tiempo de levantamiento de los SCR detectados en el segmento.</li> </ul> </li> <li>3. SiNo <ul style="list-style-type: none"> <li>a. Para <math>i=0</math> hasta <math>N_{scr}</math> hacer <ul style="list-style-type: none"> <li>i. Si la muestra <math>scr[i]</math> es la más cercana al 10% de la amplitud entonces <ul style="list-style-type: none"> <li>1. Se define como el 10% de la amplitud (<math>indx_{10} = i</math>).</li> </ul> </li> <li>ii. Si la muestra <math>scr[i]</math> es la más cercana al 90% de la amplitud entonces <ul style="list-style-type: none"> <li>2. Se define como el 90% de la amplitud (<math>indx_{90} = i</math>).</li> <li>1. Se calcula el tiempo de levantamiento del SCR como <math display="block">TL = (indx_{90} - indx_{10})/f_s.</math> </li> </ul> </li> <li>iii. Fin Si</li> </ul> </li> <li>b. Fin Para</li> </ul> </li> <li>4. Fin Si</li> </ul>
<p style="text-align: center;"><b>Quantiles(Amp, numSCR)</b></p> <ul style="list-style-type: none"> <li>1. Se obtiene la posición <math>n</math> del cuantil 25</li> <li>2. Si <math>n</math> es un número entero <ul style="list-style-type: none"> <li>a. El cuantil 25 es el promedio de las amplitudes en la posición <math>n</math> y <math>n+1</math></li> </ul> </li> <li>3. SiNo <ul style="list-style-type: none"> <li>a. El cuantil 25 es la amplitud en la posición <math>int(n+1)</math></li> </ul> </li> <li>4. Fin Si</li> <li>5. Se obtiene la posición <math>n</math> del cuantil 50</li> <li>6. Si <math>n</math> es un número entero <ul style="list-style-type: none"> <li>a. El cuantil 50 es el promedio de las amplitudes en la posición <math>n</math> y <math>n+1</math></li> </ul> </li> <li>7. SiNo <ul style="list-style-type: none"> <li>a. El cuantil 50 es la amplitud en la posición <math>int(n+1)</math></li> </ul> </li> <li>8. Fin Si</li> <li>9. Se obtiene la posición <math>n</math> del cuantil 75</li> <li>10. Si <math>n</math> es un número entero <ul style="list-style-type: none"> <li>a. El cuantil 75 es el promedio de las amplitudes en la posición <math>n</math> y <math>n+1</math></li> </ul> </li> <li>11. SiNo <ul style="list-style-type: none"> <li>a. El cuantil 75 es la amplitud en la posición <math>int(n+1)</math></li> </ul> </li> </ul>

12. Fin Si
13. Se obtiene la posición  $n$  del cuantil 90
14. Si  $n$  es un número entero
  - a. El cuantil 90 es el promedio de las amplitudes en la posición  $n$  y  $n+1$
15. SiNo
  - a. El cuantil 90 es la amplitud en la posición  $int(n+1)$
16. Fin Si

#### 5.4.5 Dominio de la frecuencia

Los archivos que componen a los programas para la extracción de características en el dominio de la frecuencia pueden encontrarse en la raíz “CCS/Emotion\_Detector\_CPU1/Freq\_Extraction” del repositorio “Emotion\_Detector\_UNAM” cuyo acceso se da en el código QR de la Figura 63.

Primeramente, dentro del archivo “Cubic\_Spline.c” se realiza la interpolación de la señal PRV a través del método de trazadores cúbicos, con base en la teoría descrita en [56] y siguiendo el pseudocódigo de la Tabla 30, el cual contempla la resolución de un sistema de ecuaciones a través del algoritmo de Thomas, que resulta una simplificación del método de Gauss aplicado para matrices tridiagonales. En éste se contemplan tanto la reducción para la matriz de constantes a un vector, como la definición de un límite en la sustitución hacía atrás para los datos futuros interpolados en el paso (4.e.i), tal como se explicó en la sección 4.4.4.5. No obstante, en lugar de adquirir los coeficientes A, B, C y D de todas las funciones cúbicas que integran a la función interpolada tras finalizar la sustitución hacía atrás, se opta realizar la interpolación de dicho intervalo a 8 Hz al instante de estimar sus coeficientes. Por ende, dicha interpolación se realiza a partir del límite de 256 s de la ventana de datos de la señal PRV hacía atrás hasta interpolar todos los puntos posibles dentro de ese intervalo, para luego continuar así sucesivamente con el resto de los intervalos faltantes. De esta manera, se evita el almacenamiento de todos los coeficientes de las funciones cúbicas que conforman a la señal interpolada.

Tabla 30. Pseudocódigo del método de interpolación por trazadores cúbicos.

<b>Pseudocódigo</b>	
<b>Cubic_Spline (&amp;h, &amp;Y)</b>	
1.	Establecer condiciones iniciales para el proceso de interpolación, fijando el <i>Umbral</i> a partir del cual comenzará la interpolación hacía atrás de la señal.
2.	Para $i=Low\_N$ hasta $Up\_N$ hacer <ol style="list-style-type: none"> <li>a. Estimar el valor de un elemento del vector de coeficientes <i>Coeff_A</i> y su elemento del vector de constantes <i>Const_B</i> de acuerdo con las ecuaciones respectivas de la interpolación por trazadores cúbicos, empleando para ello el valor del elemento <math>i</math> del arreglo <math>h</math> e <math>Y</math>.</li> <li>b. Comenzar el barrido hacía adelante del algoritmo de Thomas, que conlleva la eliminación de los elementos recién estimados con sus anteriores.</li> </ol>
3.	Fin Para
4.	Para $i=Up\_N$ hasta $-1$ hacer <ol style="list-style-type: none"> <li>a. Realizar la sustitución hacía atrás del algoritmo de Thomas.</li> <li>b. Si se alcanza el límite del 1% de las nuevas estimaciones con respecto a las estimaciones previamente realizadas entonces               <ol style="list-style-type: none"> <li>i. Fin Para</li> </ol> </li> <li>c. Fin Si</li> <li>d. Estimar los coeficientes <math>A, B, C</math> y <math>D</math> de la función cúbica del intervalo <math>h[i]</math></li> <li>e. Mientras <math>Umbral \geq 0</math> y <math>j \geq 0</math> entonces</li> </ol>

- i. Interpolar la señal a 8 Hz dentro del arreglo  $CubSpl[j]$  a partir del valor de  $Umbral$ .
  - ii. Disminuir el valor de  $Umbral$  en 0.125 (correspondiente al inverso de 8 Hz) y de  $j$  en 1.
- f. Fin Mientras
- g. Añadir el residuo de  $Umbral$  al siguiente intervalo de la señal  $h[i - 1]$
- 5. Fin Para
- 6. Estimar el número de elementos que conforman al desplazamiento de valores considerando una ventana de 64 s y siguiendo los pasos descritos en la Figura 57.
- 7. Configuración de los canales DMA para los valores  $Coeff\_A$ ,  $Const\_B$  y  $S$ .
  - a. Retorno de la primera dirección del arreglo  $CubSpl$  con la señal interpolada.

Originalmente se planteó estimar la PSD de las señales fisiológicas a través del método paramétrico de Burg, cuyo código puede encontrarse en el archivo “PSD\_Burg.c”, sin embargo, tras su implementación y puesta a prueba en el DSP TMS320F28379D se identificó un error en los resultados entregados por el algoritmo conforme se aumentaba el orden del modelo AR. La causa de ello es una falta de precisión en la aritmética en punto flotante a 32 bits con la que dispone la tarjeta, ya que se realizó la comparación de la misma señal procesada en la tarjeta con una implementación a través del software matemático MATLAB, cuyas operaciones se realizan en punto flotante a 64 bits, del cual se adquieren resultados correctos. De esta forma, se analizó el método no paramétrico de Welch del cual se identificó que para satisfacer la precisión suficiente para los intervalos de frecuencia mencionados en la sección 4.2.4, se requiere de la segmentación de la señal en diversas ventanas de menor tamaño que permitan realizar el traslape solicitado por el algoritmo. Este aspecto podría solventarse al incrementar aún más el tamaño de la ventana original por encima de los 64 s, sin embargo, se declinó esta solución debido a que la latencia de los resultados terminaría siendo cada 256 s, lo cual se considera poco práctico para cualquier aplicación por realizar. Por ende, se opta por llevar a cabo la estimación de la PSD mediante el método no paramétrico del periodograma que permite garantizar la resolución espectral solicitada. El fundamento teórico de este método puede encontrarse en [50] y su programa implementado yace dentro del archivo “PSD\_Periodogram.c”.

Para garantizar una ejecución rápida de la FFT necesaria en el periodograma se realiza su implementación mediante lenguaje ensamblador dentro del archivo “FFT.asm” tomando provecho de la arquitectura RISC de la que dispone el procesador digital de señales para hacer uso del pipeline de las instrucciones de la TMU. A su vez, se evita la estimación de la raíz cuadrática de las componentes espectrales al final del algoritmo, tras adquirirse la magnitud de las componentes real e imaginarias de cada espectro, ya que el periodograma solicita el cuadrado de cada una de ellas. Cabe mencionar que dentro de la implementación anterior se aprovecha el proceso de reversión de bits en la FFT para realizar un ventaneo Hann a la señal para reducir los efectos de derrame espectral. La elección de esta ventana es debido a que dispone de una buena atenuación en sus lóbulos laterales, por debajo de los -31 dB, y de un lóbulo central angosto en 0.001953 Hz (dado el análisis con 2048 datos) que concuerda con la resolución deseada de 0.003906 Hz para el espectro, como se menciona en la sección 4.4.4.5. Además de la facilidad con la que puede estimarse su expresión matemática empleando las unidades matemáticas trigonométricas de la tarjeta TMS320F28379D.

Empleando las funciones anteriores, se conforma una función general, ubicada en el archivo “Freq\_Extraction.c”, encargada de realizar la extracción de las características para las señales PRV y SCR siguiendo el pseudocódigo descrito en la Tabla 31 que considera los intervalos de frecuencia establecidos para cada una de ellas en la sección 4.2.4.

Tabla 31. Pseudocódigo de la función para extracción de características en el dominio de la frecuencia.

<b>Pseudocódigo</b>	
<b>Freq_Extraction (&amp;InputX, &amp;InputY, Type)</b>	
1.	Si $Type=PRV$ entonces <ol style="list-style-type: none"> <li>a. Interpolan los puntos de los arreglos <math>InputX</math> e <math>InputY</math> mediante el método de trazadores cúbicos para obtener el apuntador a un arreglo <math>Ptr</math> con la señal interpolada. Dentro de la función de interpolación se estima el desplazamiento por realizar a los datos tal como se ejemplifica en la Figura 57.</li> <li>b. Habilitar los canales DMA para el vector de coeficientes <math>Coeff\_A</math>, el vector de incógnitas <math>S</math> y el vector de constantes <math>Const\_B</math> para realizar el desplazamiento de datos de acuerdo con la estimación realizada en el paso anterior.</li> <li>c. Estimar la PSD de la señal PRV interpolada en el apuntador <math>Ptr</math> mediante el método del periodograma. Su resultado se adquiere nuevamente a través del mismo apuntador <math>Ptr</math> al arreglo con la PSD.</li> <li>d. Calcular las características en frecuencia de la señal PRV correspondientes a los intervalos VLF, LF, HF y la razón LF/HF a partir de las componentes espectrales de <math>Ptr</math>.</li> </ol>
2.	Si No <ol style="list-style-type: none"> <li>a. Estimar la PSD de la señal SCR, correspondiente a <math>InputY</math>, mediante el método del periodograma. Su resultado se adquiere nuevamente a través del mismo apuntador <math>Ptr</math> al arreglo con la PSD.</li> <li>b. Calcular la característica HF en frecuencia de la señal SCR.</li> <li>c. Fin Si</li> </ol>

## 5.5 Clasificación

Para la clasificación de las características extraídas se desarrolla un algoritmo utilizando una mezcla de código en lenguaje C y ASM. Éste implementa las operaciones necesarias para realizar la identificación del estado emocional de una persona utilizando el esquema EMDC descrito en la sección 4.3.5. El subsistema de clasificación consta de dos funciones individuales, denominadas **FLD** () y **LDA** (), las cuales reducen y clasifican las características de acuerdo con los parámetros obtenidos en sus homólogos del sistema en la computadora, tal como se describió en el Diseño en el nivel sistema y en el Diseño de detalle. En el caso de la función enfocada a FLD, se realiza una multiplicación y acumulación de cada una de las características extraídas (tomando en consideración los índices de aquellas seleccionadas por el algoritmo SFFSn) con su correspondiente elemento de la matriz de reducción  $W_n$ , obteniendo así un vector unidimensional dadas las razones expuestas en la sección 4.4.7.4. En el caso de la función enfocada a LDA, se realiza el cálculo de las probabilidades de pertenencia a cada clase por medio de dos funciones lineales que utilizan el vector de características reducido y los parámetros  $m_{k,n}$ ,  $\ln(\pi_k)_n$  y  $S_n$ . De esta forma, se realiza una comparación para identificar la clase con la mayor probabilidad y, posteriormente, retornar un valor de ‘0’ o ‘1’, el cual indica la pertenencia a una u otra clase. Dichas funciones son integradas dentro de una función general que ejecuta al esquema EMDC, donde en cada etapa se realiza la reducción (FLD) y clasificación (LDA) utilizando los parámetros correspondientes a cada una de éstas, los cuales son adquiridos del subsistema de Configuración (véase la Tabla 13). Asimismo, dentro de dicho esquema se emplea un sistema de comparaciones con los resultados obtenidos internamente, esto con el objetivo de identificar el cuadrante del modelo dimensional en el que se encuentra el individuo por medio de las clasificaciones a nivel neutralidad, excitación y valencia, dicha estructura es explicada a detalle dentro de la sección 4.3.5.

El código utilizado para el subsistema de clasificación puede ser encontrado dentro de la raíz “CCS/Emotion\_Detector\_CPU1/Classification” del repositorio disponible a través del código QR de

la Figura 63. A continuación, se muestra una descripción paso a paso de las funciones más relevantes del archivo “Classification.c”.

Tabla 32. Pseudocódigo de las funciones del subsistema de clasificación.

<b>Pseudocódigo</b>	
<b>EMDC (&amp;features)</b>	
1.	Utilizando la información de los registros <i>Asig_car</i> se crea un vector con cuatro valores de 32 bits que indican, de forma unificada, las características seleccionadas en cada etapa del esquema EMDC.
2.	Se realiza la reducción del vector de características por medio de la función <b>FLD</b> (). Ésta recibe como argumento el vector <i>features</i> , la matriz $W_0$ y el primer valor de índices codificados. El resultado se guarda en la variable <i>x</i> .
3.	Se realiza la estimación de la clase durante la etapa de neutralidad por medio de la función <b>LDA</b> () y los parámetros $m_{k,1}$ , $\ln(\pi_k)_1$ y $S_1$ . El resultado se guarda en <i>prediction_1</i> .
4.	Si <i>prediction_1=1</i> entonces
a.	Se retorna el valor 7 que indica la clase Neutral.
5.	SiNo
a.	Se realiza la reducción del vector de características por medio de la función <b>FLD</b> (). Ésta recibe como argumento el vector <i>features</i> , la matriz $W_1$ y el segundo valor de índices codificados. El resultado se guarda en la variable <i>x</i> .
b.	Se realiza la estimación de la clase durante la etapa de excitación por medio de la función <b>LDA</b> () y los parámetros $m_{k,2}$ , $\ln(\pi_k)_2$ y $S_2$ . El resultado se guarda en <i>prediction_2</i> .
c.	Si <i>prediction_2=1</i> entonces
i.	Se realiza la reducción del vector de características por medio de la función <b>FLD</b> (). Ésta recibe como argumento el vector <i>features</i> , la matriz $W_2$ y el segundo valor de índices codificados. El resultado se guarda en la variable <i>x</i> .
ii.	Se efectúa la estimación de la clase durante la etapa de valencia por medio de la función <b>LDA</b> () y los parámetros $m_{k,3}$ , $\ln(\pi_k)_3$ y $S_3$ . El resultado se guarda en <i>prediction_3</i> .
iii.	Si <i>prediction_3=1</i> entonces
1.	Se retorna el valor 6 que indica la clase Excitación Alta – Valencia Positiva.
iv.	SiNo
1.	Se retorna el valor 4 que indica la clase Excitación Alta – Valencia Negativa.
v.	Fin Si
d.	SiNo
i.	Se realiza la reducción del vector de características por medio de la función <b>FLD</b> (). Ésta recibe como argumento el vector <i>features</i> , la matriz $W_3$ y el segundo valor de índices codificados. El resultado se guarda en la variable <i>x</i> .
ii.	Se efectúa la estimación de la clase durante la etapa de valencia por medio de la función <b>LDA</b> () y los parámetros $m_{k,4}$ , $\ln(\pi_k)_4$ y $S_4$ . El resultado se guarda en <i>prediction_4</i> .
iii.	Si <i>prediction_3=1</i> entonces
1.	Se retorna el valor 2 que indica la clase Excitación Baja – Valencia Positiva.
iv.	SiNo
1.	Se retorna el valor 1 que indica la clase Excitación Baja – Valencia Negativa.
v.	Fin Si
e.	Fin Si
6.	Fin Si
7.	

<b>FLD (&amp;features, &amp;W, index)</b>	
1.	Mientras el número de características sea menor a 21 entonces <ol style="list-style-type: none"> <li>a. Si la característica en cuestión fue seleccionada por la SFFSn entonces           <ol style="list-style-type: none"> <li>i. Se multiplica la característica por el valor correspondiente de la matriz <math>W_n</math> y posteriormente se acumula en la variable <math>y</math>.</li> </ol> </li> </ol>
2.	Fin Mientras
3.	Se retorna el valor de $y$ .
<b>LDA (&amp;MeanVect, &amp;ApriVect, &amp;Cov_S, entry, level)</b>	
1.	Se calcula la probabilidad de la primera clase mediante la función lineal respectiva del algoritmo LDA. El resultado se guarda en $y_1$ .
2.	Se calcula la probabilidad de la segunda clase mediante la función lineal respectiva del algoritmo LDA. El resultado se guarda en $y_2$ .
3.	Si $y_1 < y_2$ entonces <ol style="list-style-type: none"> <li>a. Retorna 1.</li> </ol>
4.	Si no <ol style="list-style-type: none"> <li>a. Retorna 0.</li> </ol>

## 5.6 Ahorro de energía

Se elige una frecuencia de reloj de 10 MHz con el objetivo de mantener una velocidad que permita un bajo consumo energético sin sacrificar el desempeño del dispositivo.

Asimismo, se implementa una característica estándar de la familia de CPUs C28x denominada IDLE, esto con el objetivo de satisfacer la especificación marginal sobre las técnicas para el ahorro de energía indicada en la sección 4.2.2. Dicha modalidad detiene el reloj del CPU hasta que ocurra alguna interrupción por parte de los periféricos, pues en el caso de éstos sus relojes siguen en funcionamiento. Una vez llegado el evento, el reloj del CPU regresa a su estado normal mientras no sea ejecutada nuevamente la instrucción que habilite el estado IDLE.

En el detector de emociones se utiliza dicha modalidad para que el CPU conserve energía durante periodos donde no se esté ejecutando alguna tarea, por ejemplo, al estar apagado el módulo HM10 del dispositivo o en los periodos donde no se realice la extracción de características. Su uso puntual puede visualizarse en la raíz “CCS/Emotion\_Detector\_CPU1/main.c” del repositorio disponible a través del código QR de la Figura 63.

## 5.7 Constitución final del sistema embebido

La consolidación final del sistema embebido como parte del detector de emociones se realiza a través de los distintos elementos mencionados a lo largo de la sección 4.4, los cuales comprenden los costos enlistados en la Tabla 33 que dan lugar a un costo de componentes del sistema de \$1159.38 MXN. A su vez, el conjunto total de las dimensiones del sistema no cumple con las especificaciones marginales dictadas en la sección 4.2.2 a excepción de la batería.

Tabla 33. Costo y dimensiones de los elementos del sistema embebido

Elemento	Modelo	Dimensiones ( $mm^2$ )	Costo (MXN)
Tarjeta de desarrollo	LAUNCHXL-F28379D	130 x 57	\$670.56
Módulo RF	HM10	26.9 x 13	\$72.00
Sensor biométrico + PCB	AS7026GG	16 x 17	\$356.95
Dispositivo optoelectrónico	LED RGB	15 x 19	\$8.52
Batería + Regulación	No aplica	37 x 19	\$51.35
<b>Total</b>		9019.7	\$1159.38

El empleo de dos electrodos de referencia de plata/cloruro de plata (Ag/AgCl) da lugar a que el dispositivo pueda emplearse en la palma de la mano, cumpliendo con la especificación marginal del número de zonas del cuerpo para uso del dispositivo en la sección 4.2.2. En este trabajo se utiliza la superficie volar en falanges mediales para el estudio de la señal EDA.

Para la alimentación del sistema embebido se selecciona una batería de 400 mAh dado que se cuenta con un consumo constante de 60.8 mA durante la ejecución continua del sistema, contemplando la adquisición y procesamiento de las señales fisiológicas, y de 61.3 mA al realizarse la transmisión de datos mediante BLE al sistema en la computadora. Cabe mencionar que las mediciones anteriores fueron adquiridas a través de un multímetro UNI-T UT61C. De esta forma, el sistema brinda una autonomía de 6 hrs y 30 min bajo un régimen de operación continua, por lo que se cumple con el valor marginal descrito en las especificaciones objetivo de la sección 4.2.2. A su vez, se integra un módulo TP4056 con el propósito de brindar protección por sobre carga, sobre descarga y sobre corriente al sistema, así como un regulador lineal de 3.3 V para abastecer del voltaje adecuado de la batería a la tarjeta de desarrollo, el módulo HM10 y el sensor biométrico.

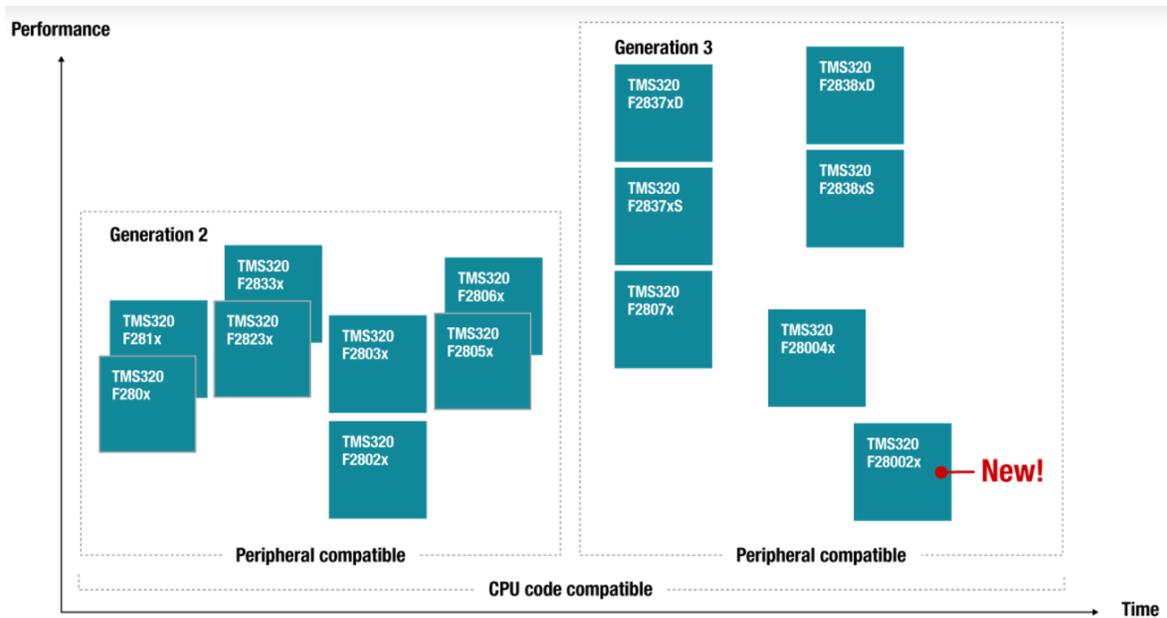


Figura 68. Familia de dispositivos compatibles con el DSP TMS320F2837xD.  
 Imagen adquirida de [106]

El número de familias en los que puede replicarse el sistema embebido resulta de al menos otra adicional del mismo fabricante del DSP seleccionado, siendo esta la familia TMS320F2807x como se ilustra en la Figura 68. Con esta y la familia de dispositivos TMS320F2837x se alcanza un total de 2 familias de dispositivos en los que el sistema puede replicarse, cumpliendo de esta forma con la especificación ideal planteada dentro de la sección 4.2.2. En la Figura 67 puede observarse la integración del sistema embebido completo mediante conexiones en una protoboard.

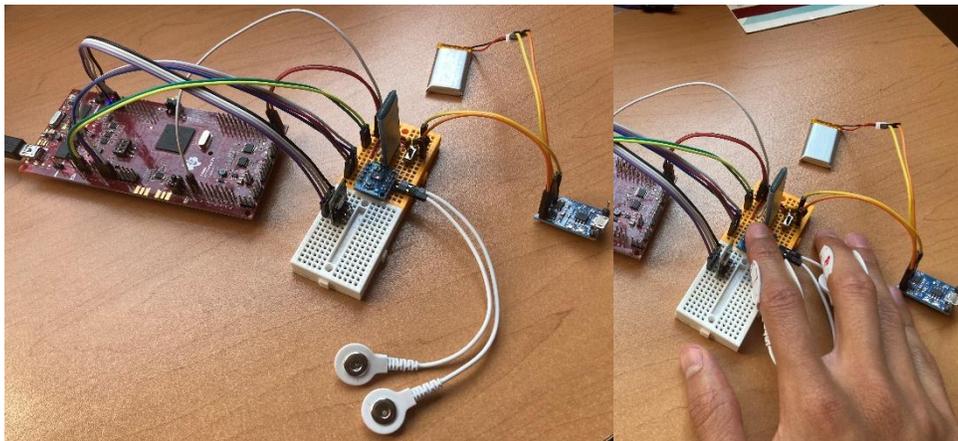


Figura 67. Integración final del sistema embebido.

## 5.8 Sistema en la computadora

El sistema se implementa a través del lenguaje de programación Python en su versión 3.8.5 empleando para ello el entorno de desarrollo integrado Spyder dentro de un sistema operativo Windows 10. La razón por la que se utiliza dicho lenguaje es debido a la versatilidad que lo distingue dada su naturaleza multiparadigma, lo cual le permite adecuarse a distintas necesidades como lo puede ser el desarrollo de los elementos detallados en la sección 4.4.7. Dentro de su repertorio se encuentran herramientas para la creación de interfaces gráficas, comunicación utilizando BLE e implementación de algoritmos que ameriten análisis de datos y cálculos matriciales, permitiendo así desarrollar

algoritmos de Aprendizaje Automático con soltura. Éste último rasgo ha propiciado que Python sea uno de los lenguajes más utilizados en dicha área. A continuación, se describe la implementación de cada uno de los subsistemas desarrollados dentro de este lenguaje, las bibliotecas empleadas en ellos y su ubicación dentro del repositorio mostrado en la sección 5.

### 5.8.1 Interfaz Gráfica de Usuario

Tomando en cuenta los requerimientos planteados en el proceso de diseño y con el objetivo de cumplir con una interfaz gráfica que brinde al usuario una visualización clara de los datos recibidos y transmitidos, la GUI es implementada mediante la biblioteca *Tkinter*. Esta herramienta permite crear elementos interactivos como botones, cuadros de texto, graficas, entre otros; por lo que cumple con las características necesarias para representar la información y gestionar el sistema en la computadora correctamente. Dado lo anterior, se cumplen las especificaciones objetivo planteadas en la sección 4.2.2, referentes a la fácil interpretación de los resultados y una interfaz sencilla para el usuario. A continuación, se detalla cada una de las secciones y botones, así como los pasos a seguir para emplearla correctamente.

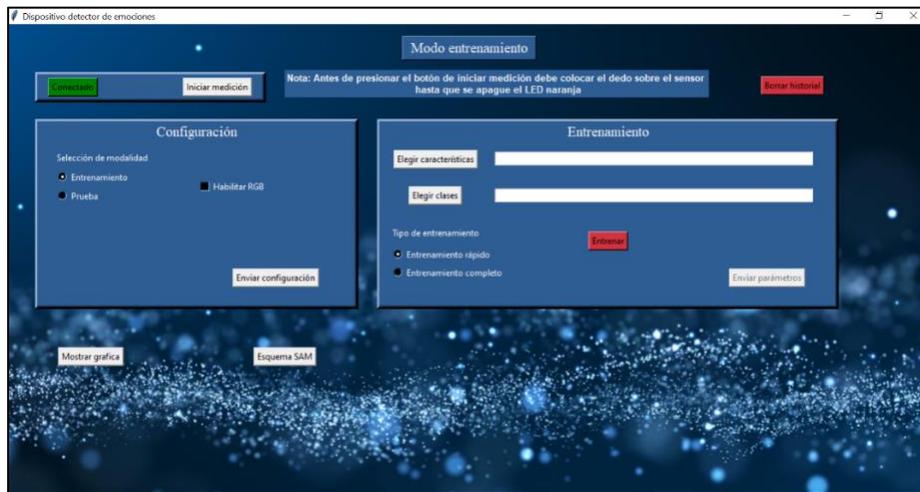


Figura 69. Ventana principal de la Interfaz Gráfica de Usuario

- *Conexión con el dispositivo y el sensor biométrico:*

Tanto el botón de “Conectar” como el de “Iniciar medición” se encuentran en la parte superior izquierda de la ventana. Con el propósito de ofrecer una visualización más clara, ambos botones modifican su color indicando el estado del sistema, tal como se muestra en la Figura 70.



Figura 70. Estado de los botones "Conectar" e "Iniciar medición".

Inicialmente, todos los botones de la interfaz se encuentran desactivados, por lo que es necesario realizar la conexión con el detector de emociones para que se habiliten. Además, al comenzar la medición los botones de envío de parámetros son desactivados.

- *Reducción de características y entrenamiento:*

Para realizar el entrenamiento del sistema, primeramente deben seleccionarse los archivos que contienen los vectores de características y sus clases correspondientes, empleando para ello los botones “Elegir características” y “Elegir clases”, respectivamente. Éstos se encuentran en la sección de entrenamiento de la ventana principal, tal como se ilustra en la Figura 71.



Figura 71. Botones para la selección de archivos

Al presionar el botón “Elegir características”, el explorador de archivos de la computadora se despliega y debe seleccionarse un archivo de extensión .csv con las características a emplear, el cual debe cumplir con la estructura descrita en la sección 5.8.6. Así mismo, si se presiona el botón “Elegir clases”, el archivo a seleccionar en el explorador debe contener las clases por utilizar, cumpliendo con la extensión y estructura mencionada con anterioridad.

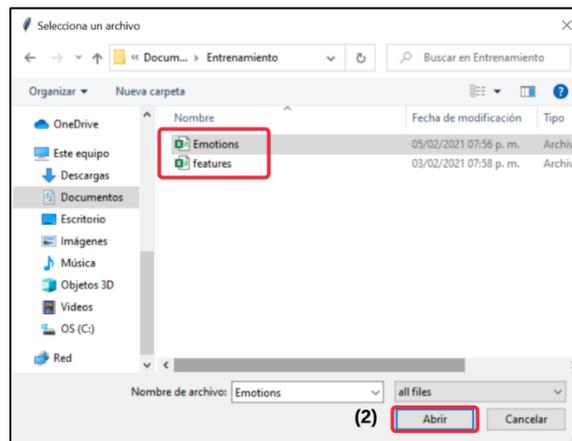


Figura 72. Explorador de archivos de la computadora

A continuación, debe elegirse un tipo de entrenamiento (rápido o completo) y entrenar al sistema mediante el botón “Entrenar”. Si se han seleccionado ambos archivos se mostrará una ventana indicando que el entrenamiento ha sido exitoso, en caso contrario el sistema indicará los archivos faltantes.

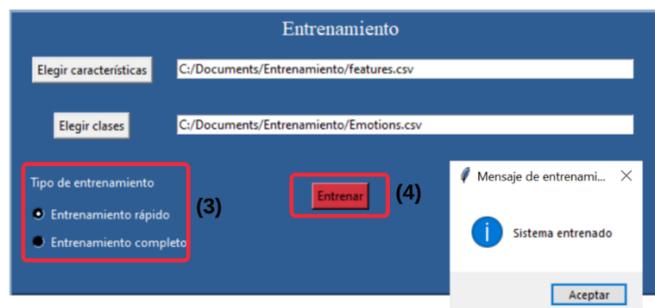


Figura 73. Realización del entrenamiento

Una vez entrenado el sistema, las características son enviadas al detector de emociones presionando el botón “Enviar parámetros”, mostrado en la Figura 74. Éste se encuentra desactivado al abrir la interfaz y es necesario entrenar al sistema para activarlo. Una vez presionado, la interfaz preguntará

al usuario si desea enviar los datos, por lo que debe seleccionarse *aceptar* en caso de estar seguro y cerrar la ventana en caso contrario.



Figura 74. Envío de parámetros de entrenamiento al detector de emociones

Por último, si se desean eliminar los datos contenidos en los archivos “features.csv” y “Emotion.csv”, es necesario presionar el botón “Borrar historial” encontrado en la parte superior derecha de la ventana principal de la GUI.

- *Configuración:*

La sección de configuración permite seleccionar si el LED RGB es activado o desactivado, realizar la transmisión automática, además de elegir entre las dos modalidades disponibles, entrenamiento o prueba. Una vez definida la configuración deseada, dichos parámetros son enviados al detector de emociones presionando el botón “Enviar configuración” y posteriormente *aceptar*, tal como se ilustra en la Figura 75.



Figura 75. Sección de configuración.

- *Esquema SAM y gráfica de barras:*

Tanto la gráfica de barras como el esquema SAM son visualizados a través de las ventanas mostradas en la Figura 76, las cuales son abiertas mediante los botones “Mostrar gráfica” y “Esquema SAM”, encontrados en la parte inferior izquierda de la ventana principal.

En el esquema SAM deben asignarse los niveles correspondientes al grado de valencia y excitación experimentados durante el estímulo, lo cual se realiza seleccionando los botones radiales enumerados del 1 al 9. Por último, debe presionarse el botón “Registrar” a fin de almacenar dichos niveles, los cuales son asignados a un determinado conjunto de características.

El archivo que compone al programa de este subsistema puede encontrarse en la raíz “Python/Emotion\_Detector\_PY/User\_Interface.py” del repositorio dirigido en el código QR de la Figura 63.

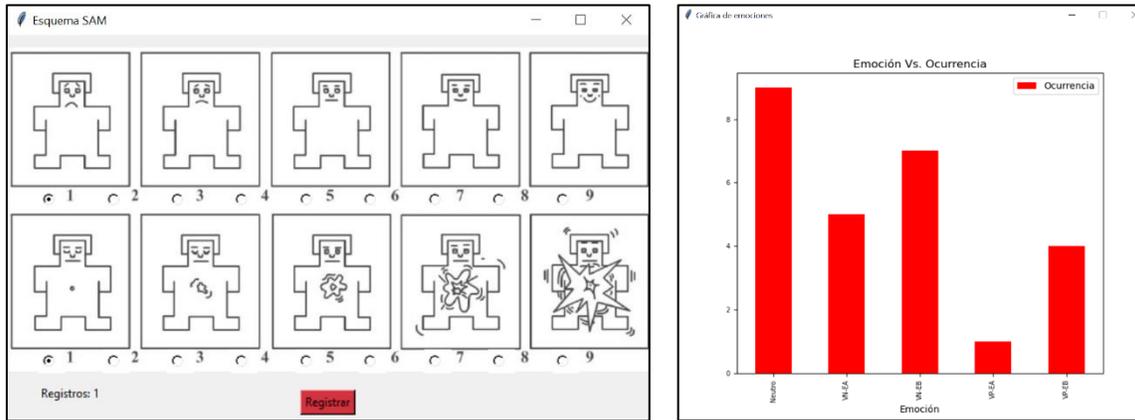


Figura 76. Ventanas emergentes del esquema SAM y la gráfica de emociones

### 5.8.2 Comunicación RF

El archivo que compone al programa de este subsistema puede encontrarse en la raíz “Python/Emotion\_Detector\_PY/RF\_Communications.py” dentro del repositorio “Emotion\_Detector\_UNAM” cuyo acceso se da en el código QR de la Figura 63.

De primera mano, se hace uso del módulo *asyncio*, ubicado dentro de la biblioteca *aioconsole* versión 0.3.1, que permite la programación del subsistema a través de un esquema en concurrencia. Esto dado que la recepción y transmisión de datos corresponden a eventos asíncronos que, de otra manera, tendrían que consultarse constantemente a través de un método de polling. Para esto se solicita la creación de un ente denominado *loop* dentro del cual es posible asignar diversas tareas que serán ejecutadas de forma concurrente por el CPU. La clase que engloba a los atributos y métodos de este subsistema, como se mencionó en la sección 4.4.7.2, se denota mediante el nombre *RF\_COMS* y su constructor toma como parámetros el *loop* y el UUID del módulo HM10 del sensor biométrico.

Para la manipulación de la comunicación BLE se modifica el trabajo realizado en [79], el cual emplea la biblioteca *Bleak* en su versión 0.9.1 y facilita el uso de funciones para generar un cliente BLE en la computadora. De este trabajo se suprimen algunas funciones irrelevantes y se modifican otras a fin de optimizar su funcionamiento.

Para el acceso a archivos en la computadora se emplea la biblioteca *Pandas* con la cual se conforman dos marcos de datos de los archivos “features.csv” y “Emotions.csv” como atributos de la clase *RF\_COMS*. A su vez, se declaran dos atributos que adquieren en su valor la ubicación de la última entrada a cada uno de los archivos, permitiendo, de esta forma, continuar su edición tras ejecutarse nuevamente el programa en distintas sesiones.

Para la configuración de los registros del detector de emociones se conforman varios atributos con el mismo nombre y valor hexadecimal a 8 bits, correspondientes a los registros de la Tabla 13. Cabe mencionar que los atributos *Chars\_Asig*, *FLD\_W*, *Pik\_Vect*, *Mean\_Vect* y *Cov\_S* solo adquieren su valor tras la ejecución de las etapas de clasificación en los subsistemas SFFS, FLD y LDA. La excepción a estos tamaños son los registros con la emoción clasificada y las características extraídas de las señales fisiológicas, ya que el primero se guarda como un valor entero y el segundo dentro de un vector de 21 elementos en punto flotante, cuyo contenido sigue el orden descrito en el registro *Caract Config* para el campo *Sel Caract*. A su vez, se constituyen cinco métodos que permiten la comunicación de ciertos registros hacia el detector de emociones mediante el protocolo BLE, la descripción de sus pseudocódigos puede observarse en la Tabla 34.

Tabla 34. Pseudocódigo de los métodos de la clase RF\_COMS para la configuración de registros del detector de emociones.

<b>Pseudocódigo</b>
<b>async Connect_EmotSensor ( )</b>
1. Transmisión de caracteres “EM_DET_UNAM” mediante la función <b>BLE_Data_Tx ( )</b> .
<b>async Send_Config ( )</b>
1. Concatenar el valor del registro <i>Config</i> con la dirección 0x40 correspondiente a este registro con el bit de continuidad.
2. Transmisión de mensaje concatenado mediante la función <b>BLE_Data_Tx ( )</b> .
<b>async Send_Clasif_Parameters ( )</b>
17. Para $n \leftarrow 0$ hasta 3 (respectivo a las 4 etapas del esquema EMDC) hacer: <ul style="list-style-type: none"> <li>a. En el registro <i>CC Config</i> habilitar el bit <i>Auto</i> y asignar el valor de <math>n</math> a <i>Sel CC</i></li> <li>b. Concatenar los valores de los registros <i>Caract EDA 2</i>, <i>Caract EDA 1</i>, <i>Caract PRV 2</i>, <i>Caract PRV 1</i> correspondientes a la etapa <math>n</math> junto con el registro <i>CC Config</i> y el valor 0x41, correspondiente a la dirección de este último registro con el bit de continuidad.</li> <li>c. Transmisión de mensaje concatenado mediante la función <b>BLE_Data_Tx ( )</b>.</li> </ul>
18. Fin Para
19. Para $n \leftarrow 0$ hasta 3 (respectivo a las 4 etapas del esquema EMDC) hacer: <ul style="list-style-type: none"> <li>a. Para <math>m \leftarrow 0</math> hasta 20 (respectivo a las 21 características) hacer <ul style="list-style-type: none"> <li>i. En el registro “<math>W_n</math> Config” habilitar el bit <i>Auto</i>, asignar el valor de <math>n</math> a <math>W_n</math> y <math>m</math> a <i>Elemento Mtz</i></li> <li>ii. Si <math>m &lt;</math> número de elementos de <math>W_n</math> entonces <ul style="list-style-type: none"> <li>1. Concatenar los valores de los registros <i>Matriz FLDn <math>W_n</math></i> respectivo a la etapa <math>n</math> y el elemento <math>m</math> junto con el registro <i>CC Config</i> y la dirección 0x46, correspondiente a la dirección de este último registro con el bit de continuidad</li> </ul> </li> <li>iii. SiNo <ul style="list-style-type: none"> <li>1. Concatenar valores de cero a los registros <i>Matriz FLDn <math>W_n</math></i> respectivos a la etapa <math>n</math> y el resto de los elementos faltantes.</li> </ul> </li> <li>iv. Fin Si</li> <li>v. Transmisión de mensaje concatenado mediante la función <b>BLE_Data_Tx ( )</b>.</li> </ul> </li> <li>b. Fin Para</li> </ul>
20. Fin Para
21. Para $n \leftarrow 0$ hasta 3 (respectivo a las 4 etapas del esquema EMDC) hacer: <ul style="list-style-type: none"> <li>a. Para <math>k \leftarrow 0</math> hasta 1 (respectivo a las medias de cada clase) hacer <ul style="list-style-type: none"> <li>i. En el registro <i>Vect Config</i> habilitar el bit <i>Auto</i>, asignar el Vector <math>m_{k,n}</math> con un 0 en <i>Sel Vect</i> y asignar el valor de <math>(2 * n + m)</math> a <i>Elemento Vect</i>.</li> <li>ii. Concatenar los valores de los registros <i>Vector <math>m_{k,n}/\ln(\pi_k)_n/S_n</math></i> respectivos a la etapa <math>n</math> y el elemento <math>k</math> junto con el registro <i>Vect Config</i> y el valor 0x4B, correspondiente a la dirección de este último registro con el bit de continuidad</li> <li>iii. Transmisión de mensaje concatenado mediante la función <b>BLE_Data_Tx ( )</b>.</li> </ul> </li> <li>b. Fin Para</li> </ul>
22. Fin Para
23. Para $n \leftarrow 0$ hasta 3 (respectivo a las 4 etapas del esquema EMDC) hacer: <ul style="list-style-type: none"> <li>a. Para <math>k \leftarrow 0</math> hasta 1 (respectivo a las medias de cada clase) hacer <ul style="list-style-type: none"> <li>i. En el registro <i>Vect Config</i> habilitar el bit <i>Auto</i>, asignar el Vector <math>\ln(\pi_k)_n</math> con un 1 en <i>Sel Vect</i> y asignar el valor de <math>(2 * n + m)</math> a <i>Elemento Vect</i>.</li> </ul> </li> </ul>

- ii. Concatenar los valores de los registros *Vector*  $m_{k,n}/ln(\pi_k)_n/S_n$  respectivos a la etapa  $n$  y el elemento  $k$  junto con el registro *Vect Config* y el valor 0x4B, correspondiente a la dirección de este último registro con el bit de continuidad
  - iii. Transmisión de mensaje concatenado mediante la función **BLE\_Data\_Tx** ( ).
- b. Fin Para
24. Fin Para
25. Para  $n \leftarrow 0$  hasta 3 (respectivo a las 4 etapas del esquema EMDC) hacer:
- a. En el registro *Vect Config* habilitar el bit *Auto*, asignar el Vector  $S_n$  con un 2 en *Sel Vect* y asignar el valor de  $n$  a *Elemento Vect*.
  - b. Concatenar los valores de los registros *Vector*  $m_{k,n}/ln(\pi_k)_n/S_n$  respectivo a la etapa  $n$  junto con el registro *Vect Config* y la dirección 0x4B, correspondiente a este último registro con el bit de continuidad.
26. Fin Para

#### Rx\_Menu (Addr, Data)

1. Si Addr=0x10 entonces
  - a. Escribir Data al atributo *Emotion*.
  - b. Decodificar el valor de *Emotion* a los niveles de neutralidad, valencia y excitación.
  - c. Escribir valor decodificado de *Emotion* a archivo “Emotions.csv”
2. SiNo Si Addr=0x11 entonces
  - a. Guardar el valor recibido en *Sel Caract* en una variable *index* y verificación del bit *Auto* para permitir la recepción consecutiva de los registros 0x12 a 0x15 sin la recepción de su campo de dirección.
3. SiNo Si Addr=0x12 entonces
  - a. Guardar la parte fraccionaria B de la característica definida con dentro de una variable *DataQ16*
4. SiNo Si Addr=0x13 entonces
  - a. Guardar la parte fraccionaria A de la característica definida con dentro de una variable *DataQ16*
5. SiNo Si Addr=0x14 entonces
  - a. Guardar la parte entera B de la característica definida con dentro de una variable *DataQ16*
6. SiNo Si Addr=0x15 entonces
  - a. Guardar la parte entera A de la característica definida con dentro de una variable *DataQ16*
  - b. Convertir *DataQ16* a flotante y guardarlo en su atributo *Chars\_Val[]* de acuerdo al valor que haya adquirido *index*.
  - c. Escribir dato en flotante dentro de la característica que le corresponda dentro del archivo “features.csv”
7. Fin Si

### 5.8.3 Selección secuencial hacia delante flotante

Como se indica en la sección 4.4.7.3 se implementa el algoritmo SFFS de dos formas distintas, una que permite ejecutar un entrenamiento rápido y otra que posibilita un entrenamiento completo. Dichas modalidades se diferencian principalmente en el tiempo de ejecución y la exactitud con la que sea realiza la selección de características, sin embargo, se considera que ambas opciones ofrecen un buen rendimiento para la aplicación en cuestión. En el entrenamiento rápido se emplea la validación simple para llevar a cabo la prueba de los subconjuntos creados durante la ejecución del algoritmo y, por el contrario, en el entrenamiento completo se utiliza la validación cruzada dejando uno fuera. Para implementar ambas tareas se crea un módulo que ejecuta la Selección secuencial hacia delante flotante utilizando ambas validaciones, las cuales se separan en dos funciones distintas denominadas

**SFFS\_wrapper ()** y **SFFS\_wrapper\_LOOCV ()**. Dicho módulo puede ser encontrado dentro de la raíz “Python/Emotion\_Detector\_PY/ML/SFFS.py” del repositorio disponible a través del código QR de la Figura 63.

En la modalidad de entrenamiento rápido se toman como argumento dos conjuntos de vectores de características (entrenamiento y prueba), así como sus respectivas clases. La necesidad de ingresar ambos conjuntos es debido a la ejecución de la SFFS en su modalidad de envoltorio, pues ésta necesita realizar constantemente entrenamientos y validaciones del LDA. Al utilizarse una validación simple, dicho proceso se realiza una única vez por subconjunto creado a lo largo del algoritmo descrito en la sección 4.4.7.3, lo cual implica una rápida ejecución, sacrificando exactitud y precisión en la selección de las características, pues el conjunto de prueba no es del todo representativo de la información que se busca procesar. Cabe resaltar que esto no implica un pobre rendimiento del algoritmo, pues tras las pruebas realizadas se comprobó que éste es eficaz cuando se cuenta con la cantidad de vectores de características suficientes, entendiéndose por “suficientes” a un número de vectores tal que al tomar un 20% de éstos de forma aleatoria, dicha selección cumpla la premisa de compartir las mismas cualidades del conjunto total. A continuación, se muestra el pseudocódigo de la implementación con el objetivo de mostrar con mayor detalle el funcionamiento de esta versión de la SFFS dentro del proyecto.

Tabla 35. Pseudocódigo de la función SFFS con validación simple.

<b>Pseudocódigo</b>
<b>SFFS_wrapper (features_train, features_test, targets_train, targets_test, pLDA)</b>
<ol style="list-style-type: none"> <li>1. Creación de arreglos vacíos para entrenamiento y prueba denominados <i>new_features_train</i> y <i>new_features_test</i>, respectivamente.</li> <li>2. Se inicializa un contador de características <i>k</i>.</li> <li>3. Mientras <math>k &lt; D</math>, donde <i>D</i> es número total de características, entonces               <ol style="list-style-type: none"> <li>a. Se realiza una lista vacía denominada <i>accs</i> que contendrá las exactitudes calculadas.</li> <li>b. Para cada característica sin seleccionar hacer:                   <ol style="list-style-type: none"> <li>i. Se agrega la característica actual a <i>new_features_train</i> y <i>new_features_test</i>.</li> <li>ii. Se realiza el entrenamiento de LDA utilizando <i>new_features_train</i> y <i>targets_train</i>.</li> <li>iii. Se efectúan predicciones por medio de LDA utilizando los parámetros obtenidos durante el entrenamiento. Dichas predicciones se guardan en el arreglo <i>prediction</i>.</li> <li>iv. Se obtiene la exactitud mediante los arreglos <i>prediction</i> y <i>targets_test</i>, posteriormente, se guarda el resultado en <i>acc</i>.</li> <li>v. Se agrega la exactitud obtenida <i>acc</i> dentro de la lista <i>accs</i>.</li> <li>vi. Se desecha la característica agregada en los arreglos <i>new_features_train</i> y <i>new_features_test</i>.</li> </ol> </li> <li>c. Fin Para</li> <li>d. Se guarda la máxima exactitud obtenida en la iteración anterior y se guarda en <i>max_prev</i>.</li> <li>e. Se busca la máxima exactitud obtenida en la iteración actual y se guarda en <i>maxValue</i>.</li> <li>f. Si <math>maxValue &lt; max\_prev</math> entonces                   <ol style="list-style-type: none"> <li>i. Salir del ciclo</li> </ol> </li> <li>g. Fin Si</li> <li>h. Se busca índice de la característica que provocó un aumento en la exactitud.</li> <li>i. Se agrega característica de forma definitiva a los arreglos <i>new_features_train</i> y <i>new_features_test</i>.</li> <li>j. <i>k</i> incrementa en uno.</li> <li>k. Mientras <math>k &gt; 2</math> entonces                   <ol style="list-style-type: none"> <li>i. Se realiza una lista vacía denominada <i>accs</i> que contendrá las exactitudes calculadas.</li> </ol> </li> </ol> </li> </ol>

- ii. Para cada característica seleccionada hacer:
  1. Se elimina la característica actual de los arreglos *new\_features\_train* y *new\_features\_test*.
  2. Se realiza el entrenamiento de LDA utilizando *new\_features\_train* y *targets\_train*.
  3. Se realizan predicciones por medio de LDA utilizando los parámetros obtenidos durante el entrenamiento. Dichas predicciones se guardan en el arreglo *prediction*.
  4. Se obtiene la exactitud mediante los arreglos *prediction* y *targets\_test*, posteriormente, se guarda el resultado en *acc*.
  5. Se agrega la exactitud obtenida *acc* dentro de la lista *accs*.
  6. Se agrega nuevamente la característica eliminada en los arreglos *new\_features\_train* y *new\_features\_test*.
- iii. Fin Para
- iv. Se busca la máxima exactitud obtenida durante en la iteración actual y se guarda en *maxValueRemove*.
- v. Si  $maxValue < max\_prev$  entonces
  1. Salir del ciclo
- vi. Fin Si
- vii. Se busca índice de la característica que al eliminarla provocó un aumento en la exactitud.
- viii. Se elimina característica de forma definitiva de los arreglos *new\_features\_train* y *new\_features\_test*.
- ix. *k* decremента en uno.
  1. Fin Mientras
- 4. Fin Mientras
- 5. Se realiza codificación de los índices de las características seleccionadas en 4 grupos de 8 bits.
- 6. Se retornan los arreglos *new\_features\_train* y *new\_features\_test*, así como los índices codificados en una lista llamada *index\_cod*.

En la modalidad de entrenamiento completo es necesario ingresar un único conjunto de datos que contenga los vectores de características obtenidos sin distinción alguna, así como sus respectivas clases, pues dentro de la misma función se realiza la separación de éstos para ejecutar las pruebas de los subconjuntos con el LDA. Si bien el funcionamiento es el mismo que el estipulado para el entrenamiento rápido, al utilizar la validación cruzada dejando uno fuera se realiza el entrenamiento y validación de los subconjuntos creados siguiendo la metodología estipulada en la sección 3.3.5. Dicho proceso se podría resumir a, teniendo un total de  $N$  vectores en el conjunto de datos, se realizarán  $N$  entrenamientos y validaciones por cada iteración de la SFFS. Esto implica una precisión y exactitud óptima, pues se utiliza todo el conjunto de datos para realizar las evaluaciones, sin embargo, se sacrifica el tiempo de ejecución debido a la cantidad de veces que es efectuado el LDA en cada iteración de la SFFS. Asimismo, al ser un algoritmo que utiliza todos los vectores de características obtenidos, no existe variación alguna en los resultados debido al ordenamiento de los datos, a diferencia del entrenamiento rápido.

A continuación, se muestra el pseudocódigo de la implementación de esta modalidad con el objetivo de mostrar con mayor detalle su funcionamiento dentro del proyecto. Cabe resaltar que las diferencias que tiene este algoritmo y el anterior se ven reflejadas principalmente en la línea 3.b.ii y 3.k.ii.2.

Tabla 36. Pseudocódigo de la función SFFS con validación cruzada dejando uno fuera.

<b>Pseudocódigo</b>	
<b>SFFS_wrapper_LOOCV (features_train, targets_train, pLDA)</b>	
1.	Creación de un arreglo vacío para los datos denominado <i>new_features_train</i> .
2.	Se inicializa un contador de características <i>k</i> .
3.	Mientras $k < D$ , donde <i>D</i> es número total de características, entonces <ol style="list-style-type: none"> <li>a. Se realiza una lista vacía denominada <i>accs</i> que contendrá las exactitudes calculadas.</li> <li>b. Para cada característica sin seleccionar hacer:               <ol style="list-style-type: none"> <li>i. Se agrega la característica actual a <i>new_features_train</i>.</li> <li>ii. Se realiza la validación cruzada dejando uno fuera utilizando <i>new_features_train</i> y <i>targets_train</i>. Para ello se emplea el clasificador LDA por medio de la función <b>LOOCV</b> ( ) indicada en la sección 5.8.5. Como resultado se obtiene la exactitud, la cual se guarda en <i>acc</i>.</li> <li>iii. Se agrega la exactitud obtenida <i>acc</i> dentro de la lista <i>accs</i>.</li> <li>iv. Se desecha la característica agregada en <i>new_features_train</i>.</li> </ol> </li> <li>c. Fin Para</li> <li>d. Se guarda la máxima exactitud obtenida en la iteración anterior y se guarda en <i>max_prev</i>.</li> <li>e. Se busca la máxima exactitud obtenida en la iteración actual y se guarda en <i>maxValue</i>.</li> <li>f. Si <math>maxValue &lt; max\_prev</math> entonces               <ol style="list-style-type: none"> <li>i. Salir del ciclo</li> </ol> </li> <li>g. Fin Si</li> <li>h. Se busca índice de la característica que provocó un aumento en la exactitud.</li> <li>i. Se agrega característica de forma definitiva al arreglo <i>new_features_train</i>.</li> <li>j. <i>k</i> incrementa en uno.</li> </ol>
k.	Mientras $k > 2$ entonces <ol style="list-style-type: none"> <li>i. Se realiza una lista vacía denominada <i>accs</i> que contendrá las exactitudes calculadas.</li> <li>ii. Para cada característica seleccionada hacer:               <ol style="list-style-type: none"> <li>1. Se elimina la característica actual del arreglo <i>new_features_train</i>.</li> <li>2. Se realiza la validación cruzada dejando uno fuera utilizando <i>new_features_train</i> y <i>targets_train</i>. Para ello se emplea el clasificador LDA por medio de la función <b>LOOCV</b> ( ) indicada en la sección 5.8.5. Como resultado se obtiene la exactitud, la cual se guarda en <i>acc</i>.</li> <li>3. Se agrega la exactitud obtenida <i>acc</i> dentro de la lista <i>accs</i>.</li> <li>4. Se agrega nuevamente la característica eliminada en el arreglo <i>new_features_train</i>.</li> </ol> </li> <li>iii. Fin Para</li> <li>iv. Se busca la máxima exactitud obtenida durante en la iteración actual y se guarda en <i>maxValueRemove</i>.</li> <li>v. Si <math>maxValue &lt; max\_prev</math> entonces               <ol style="list-style-type: none"> <li>1. Salir del ciclo</li> </ol> </li> <li>vi. Fin Si</li> <li>vii. Se busca índice de la característica que al eliminarla provocó un aumento en la exactitud.</li> <li>viii. Se elimina la característica de forma definitiva del arreglo <i>new_features_train</i>.</li> <li>ix. <i>k</i> decrementa en uno.</li> </ol>
l.	Fin Mientras
4.	Fin Mientras

5. Se realiza codificación de los índices de las características seleccionadas en 4 grupos de 8 bits.
6. Se retornan el arreglo *new\_features\_train*, así como los índices codificados en una lista llamada *index\_cod*.

#### 5.8.4 Discriminante lineal de Fisher

Con el objetivo de generar la matriz de transformación para reducir las características en el sub-sistema de Clasificación, se genera un módulo que ejecuta el Discriminante lineal de Fisher con las especificaciones de la sección 4.4.7.4. Para ello, se genera un grupo de funciones que cumplen paso a paso el algoritmo estipulado en la sección 3.4.3 en su variante de pLDA, tomando en consideración que, al ser clasificaciones binarias, el cálculo de las matrices de reducción  $W_n$  puede ser simplificado mediante el uso de la ecuación (51). Para ello, el algoritmo detecta de forma automática si únicamente existen dos clases. Dichas funciones son integradas en una función general que permite el entrenamiento del FLD denominada **train** ( ), la cual toma como argumento los vectores de características de entrenamiento con sus respectivas clases, así como un valor que indica si se utilizará o no el pLDA. De igual forma, esta función recibe una serie de parámetros empleados para comprobar la efectividad del algoritmo al momento de su desarrollo, permitiendo así la visualización de los resultados mediante gráficas y desplegando información relevante en la consola, sin embargo, en la implementación final dichas funcionalidades están desactivadas y es necesario modificar el código fuente para habilitarlas.

El módulo creado para ejecutar el FLD puede ser encontrado dentro de la raíz “Python/Emotion\_Detector\_PY/ML/FLD.py” del repositorio disponible a través del código QR de la Figura 63. A continuación, se muestra el pseudocódigo de la implementación del algoritmo con el objetivo de mostrar con mayor detalle su funcionamiento. Cabe resaltar que, al momento de su desarrollo, éste se creó como una generalización del algoritmo capaz de trabajar con dos o más clases, aportándole un grado de versatilidad en caso de utilizarse en futuros trabajos.

Tabla 37. Pseudocódigo de la función FLD.

<b>Pseudocódigo</b>	
<b>train (features_train, targets_train, pLDA, plot, Terminal)</b>	
1.	Se realiza la identificación del número de clases y la separación de los vectores de características por clase.
2.	Se obtienen los vectores de medias por cada clase ( $m_k$ ) y el vector que representa la media total del conjunto de datos ( $m$ ).
3.	Se realiza el cálculo de las matrices de covarianza entre clases ( $S_b$ ) y dentro de las clases ( $S_w$ ).
4.	Si el conjunto solo cuenta con dos clases entonces <ol style="list-style-type: none"> <li>a. Si <math>pLDA=True</math> entonces               <ol style="list-style-type: none"> <li>i. Se obtiene las matrices pseudoinversas de <math>S_b</math> y <math>S_w</math>.</li> <li>ii. Se realiza la operación indicada en (51) y se guarda el resultado en <math>W</math>.</li> </ol> </li> <li>b. SiNo               <ol style="list-style-type: none"> <li>i. Se obtiene la matriz inversa de <math>S_w</math>.</li> <li>ii. Se realiza la operación indicada en (51) y se guarda el resultado en <math>W</math>.</li> </ol> </li> <li>c. Fin Si</li> </ol>
5.	SiNo <ol style="list-style-type: none"> <li>a. Se calculan los vectores y valores propios del resultado de la operación (47).</li> <li>b. Se buscan los <math>K</math> valores propios más grandes, donde <math>K</math> es la nueva dimensión de los vectores de características, y se identifican sus vectores propios correspondientes.</li> <li>c. Se forma la matriz <math>W</math> con los vectores propios seleccionados.</li> </ol>
6.	Fin Si
7.	Si $Terminal=True$ entonces <ol style="list-style-type: none"> <li>a. Imprime los valores propios.</li> </ol>

- b. Imprime los vectores propios.
  - c. Imprime la matriz de reducción.
8. Fin Si
9. Si  $plot=True$  entonces
  - a. Aplica la matriz de reducción al conjunto de entrenamiento.
  - b. Imprime una gráfica 2D para visualizar la proyección que se realiza de las características en las primeras dos dimensiones.
10. Fin Si
11. Se retorna la matriz  $W$  calculada, la cual corresponderá al parámetro de reducción  $W_n$  correspondiente.

### 5.8.5 Análisis discriminante lineal

Con el fin de obtener los parámetros necesarios para realizar la identificación de los estados emocionales en el subsistema de Clasificación, se crea un módulo con una función denominada **train** ( ) que ejecuta paso a paso el algoritmo de entrenamiento para LDA estipulado en la sección 3.5.1. Ésta recibe como argumento los vectores de características del conjunto de entrenamiento junto con sus respectivas clases, además de parámetros que le permiten hacer uso de la matriz pseudoinversa e imprimir en la terminal información relevante del algoritmo, sin embargo, este último se encuentra desactivado en la implementación final y en caso de requerir su habilitación debe realizarse con una modificación del código fuente. Asimismo, se incluye una función llamada **classification** ( ) que permite realizar clasificaciones sobre un conjunto de prueba con los parámetros obtenidos durante el entrenamiento, esto con el objetivo de averiguar la efectividad del algoritmo. Finalmente, existe una función denominada **LOOCV** ( ) que realiza la validación cruzada dejando uno fuera utilizando el LDA.

El módulo creado para ejecutar cada una de las funciones antes descritas puede ser encontrado dentro de la raíz “Python/Emotion\_Detector\_PY/ML/LDA.py” del repositorio disponible a través del código QR de la Figura 63. A continuación, se muestra el pseudocódigo de la implementación con el objetivo de mostrar con mayor detalle su funcionamiento. Cabe resaltar que, al momento de desarrollar el algoritmo, éste se creó como una generalización capaz de trabajar con dos o más clases, aportándole un grado de versatilidad en caso de utilizarse en futuros trabajos.

Tabla 38. Pseudocódigo de la función LDA.

<b>Pseudocódigo</b>	
<b>train (features_train, targets_train, pLDA, Terminal)</b>	
1.	Se realiza la identificación del número de clases y la separación de los vectores de características por clase.
2.	Se hace el cálculo de las probabilidades a priori de cada clase y se guardan en un arreglo denominado $pi_k$ .
3.	Se obtienen los vectores de medias de cada clase y se guardan en un arreglo llamado $m_k$ .
4.	Se calcula la matriz de covarianza, la cual es almacenada en un arreglo denominado $S$ . Cabe resaltar que dicha matriz es la media ponderada de las matrices de covarianza muestrales de las $K$ clases.
5.	Si $Terminal=True$ entonces <ol style="list-style-type: none"> <li>a. Imprime <math>S</math>.</li> <li>b. Imprime <math>m_k</math>.</li> <li>c. Imprime <math>pi_k</math>.</li> </ol>
6.	Fin Si
7.	Se retorna la matriz de covarianza ( $S$ ), los vectores de medias ( $m_k$ ) y las probabilidades a priori ( $pi_k$ ), las cuales corresponderán a los parámetros de clasificación $m_{k,n}$ , $\ln(\pi_k)_n$ y

$S_n$  correspondientes. Cabe resaltar que la probabilidad a priori es obtenida sin la función logaritmo natural implementada, esto se realiza en la función **EMDC** ( ) expresada en la sección 5.8.6.

#### **classification (S, m\_k, pi\_k, features\_entry)**

1. Se crea un arreglo denominado *prediction* para guardar las predicciones realizadas por el algoritmo.
2. Para cada muestra almacenada en *features\_entry* hacer:
  - a. Se crea una lista vacía denominada *delta* para almacenar la probabilidad de pertenencia a cada clase.
  - b. Para cada clase  $k$  existente hacer:
    - i. Realizar el cálculo de probabilidad de pertenencia a la clase  $k$  mediante la ecuación (60) ignorando la componente constante. Posteriormente, agregar el resultado a *delta*.
  - c. Fin Para
  - d. Agregar al arreglo de predicciones *prediction* el índice de la máxima probabilidad detectada dentro de la lista *delta*.
3. Fin Para
4. Retornar el arreglo *prediction*.

#### **LOOCV (features\_train, targets\_train, pLDA)**

1. Se calcula el número de vectores de características dentro de *features\_train* y se almacena en  $N$ .
2. Para cada muestra almacenada en *features\_train* hacer:
  - a. Se elimina la muestra actual del arreglo *features\_train* y su correspondiente clase en *targets\_train*.
  - b. Se agrega la muestra actual en un arreglo temporal llamado *sample\_feature* y su respectiva clase en *sample\_target*.
  - c. Se ejecuta la función **train** ( ) utilizando la variable *pLDA* y los arreglos *features\_train* y *target\_train*. Posteriormente, se almacenan los parámetros obtenidos en  $S$ ,  $m_k$  y  $pi_k$ .
    - i. Se ejecuta la función **classification** ( ) utilizando los parámetros  $S$ ,  $m_k$  y  $pi_k$ , así como el arreglo *sample\_feature*. El resultado de la clasificación es almacenado en la variable *prediction*.
  - d. Se obtiene la exactitud utilizando los arreglos *prediction* y *sample\_target*. Posteriormente, dicha exactitud se acumula en la variable *acc*.
3. Se calcula el promedio de las exactitudes obtenidas mediante la operación  $acc/N$  y se almacena en *acc*.
4. Se retorna el valor de *acc*.

### **5.8.6 Esquema EMDC**

Para realizar una integración de los algoritmos descritos en las secciones 5.8.3, 5.8.4 y 5.8.5 se crea un módulo que comunica directamente el bloque de Aprendizaje Automático con la Interfaz Gráfica de Usuario. Dicho módulo está compuesto de funciones que preparan cada etapa del esquema EMDC para su posterior implementación en el prototipo, es decir, realiza la selección de características utilizando una serie de algoritmos SFFSn tal como se describe en la sección 4.3.6.3, así como el entrenamiento de los reductores dimensionales FLDn y clasificadores LDAn, mismos que se mencionan en la sección 4.3.5 y, posteriormente, se ahonda en su implementación dentro de la sección 5.5. Para realizar dichas acciones, existen tres funciones denominadas **train** ( ), **train\_LOOCV** ( ) y **EMDC** ( ), las cuales pueden ser encontradas en el módulo con raíz “Python/Emotion\_Detector\_PY/ML/EMDC.py” del repositorio disponible a través del código QR de la Figura 63.

La función **train** ( ) es la encargada de realizar el entrenamiento rápido descrito en la sección 4.4.7.3. Ésta debe recibir como argumento las direcciones de dos archivos en formato csv llamados “features.csv” y “Emotions.csv”, los cuales deben alojar los vectores de características extraídos durante la modalidad de entrenamiento y sus respectivas clases, tal como se muestra en la Tabla 39 y Tabla 40.

Tabla 39. Formato de archivo “features.csv”

pNN50	NN50	PPImed	SDNN	RMSSD	SD1/SD2	LF/HF	VLF	PRV_LF	PRV_HF	...
...	...	...	...	...	...	...	...	...	...	

...	SD1	SD2	NumSCR	Am-pSCR	mTL	sdAmp	ctl25	ctl50	ctl75	ctl90	EDA_HF
	...	...	...	...	...	...	...	...	...	...	...

Tabla 40. Formato de archivo “Emotions.csv”

Neutral	Arousal	Valence
...	...	...

En el entrenamiento rápido es necesario efectuar un tratamiento previo de los datos, pues la validación simple necesita que éstos cuenten con una distribución en específico. Este tratamiento se realiza mediante un reordenamiento aleatorio de los vectores de características, así como su separación en un conjunto de entrenamiento (75% de los datos) y uno de prueba (25% de los de datos). Una vez obtenidos dichos conjuntos, es posible adquirir los parámetros para cada etapa del esquema EMDC. Cabe resaltar que se omite el conjunto de validación debido a que no se busca evaluar al clasificador como tal, sino hallar aquellas características que arrojen mejores resultados. Finalmente, dichos valores son codificados en cuatro grupos de 8 bits con el objetivo de transmitirlos mediante el subsistema de Comunicación RF. A continuación, se muestra el pseudocódigo de la implementación del esquema bajo esta modalidad.

Tabla 41. Pseudocódigo de la función EMDC en la modalidad de entrenamiento rápido.

<b>Pseudocódigo</b>	
<b>train (features_csv, targets_csv)</b>	
1.	Se leen los archivos “features.csv” y “Emotions.csv” alojados en las direcciones <i>features_csv</i> y <i>targets_csv</i> .
2.	Se reordenan los datos de forma aleatoria.
3.	Se separan los datos en dos arreglos. El 75% de éstos se aloja en el conjunto de entrenamiento, denotado como <i>features_train</i> y <i>targets_train</i> y el 25% en <i>features_test</i> y <i>targets_test</i> .
Primer nivel del EMDC: Neutralidad	
4.	Se ejecuta la función <b>SFFS.SFFS_wrapper</b> ( ) para hacer la selección de características en la etapa de neutralidad utilizando los arreglos antes descritos. Posteriormente, se guarda el resultado en los arreglos <i>SFS_features_train_1</i> y <i>SFS_features_test_1</i> . De igual forma, se almacenan los índices codificados en la <i>lista_index_cod_1</i> .
5.	Se realiza la unión temporal de los arreglos <i>SFS_features_train_1</i> y <i>SFS_features_test_1</i> y se guarda en <i>SFS_features_temp</i> .
6.	Se unen temporalmente los arreglos <i>targets_train</i> y <i>targets_test</i> y se guardan en <i>targets_temp</i> .

7. Se emplea la función **FLD.train** ( ) utilizando los arreglos temporales *SFS\_features\_temp* y *targets\_temp*. El resultado se guarda en *matrix\_W\_1*.
8. Se realiza la reducción de *SFS\_features\_temp* utilizando la matriz *matrix\_W\_1* y la operación indicada en (50). El resultado se guarda en *new\_features\_train\_1*.
9. Se ejecuta la función **LDA.train** ( ) utilizando *new\_features\_train\_1* y *targets\_temp* enfocándose en el nivel de neutralidad. El resultado se guarda en *S\_1*, *m\_k\_1* y *pi\_k\_1*.
10. Se realiza una validación cruzada dejando uno fuera mediante la función **LDA.LOOCV** ( ) y se imprime la exactitud obtenida.
11. Se descartan los vectores con clase “neutral” de los arreglos *features\_train* y *features\_test*, así como sus correspondientes clases en *targets\_train* y *targets\_test*. El resultado se guarda en los mismos arreglos.

Segundo nivel del EMDC: Excitación

12. Se ejecuta la función **SFFS.SFFS\_wrapper** ( ) para hacer la selección de características en la etapa de excitación utilizando los arreglos antes descritos. Posteriormente, se guarda el resultado en los arreglos *SFS\_features\_train\_2* y *SFS\_features\_test\_2*. De igual forma, se almacenan los índices codificados en la *lista\_index\_cod\_2*.
13. Se realiza la unión temporal de los arreglos *SFS\_features\_train\_2* y *SFS\_features\_test\_2* y se guarda en *SFS\_features\_temp*.
14. Se unen temporalmente los arreglos *targets\_train* y *targets\_test* y se guardan en *targets\_temp*.
15. Se emplea la función **FLD.train** ( ) utilizando los arreglos temporales *SFS\_features\_temp* y *targets\_temp*. El resultado se guarda en *matrix\_W\_2*.
16. Se realiza la reducción de *SFS\_features\_temp* utilizando la matriz *matrix\_W\_2* y la operación indicada en (50). El resultado se guarda en *new\_features\_train\_2*.
17. Se ejecuta la función **LDA.train** ( ) utilizando *new\_features\_train\_2* y *targets\_temp* enfocándose en el nivel de excitación. El resultado se guarda en *S\_2*, *m\_k\_2* y *pi\_k\_2*.
18. Se realiza una validación cruzada dejando uno fuera mediante la función **LDA.LOOCV** ( ) y se imprime la exactitud obtenida.
19. Se descartan los vectores con clase “Excitación baja” de los arreglos *features\_train* y *features\_test*, así como sus correspondientes clases en *targets\_train* y *targets\_test*. El resultado se guarda en los arreglos *features\_trainHA*, *features\_testHA*, *targets\_trainHA* y *targets\_testHA*.

Tercer nivel del EMDC: Valencia (Excitación Alta)

20. Se ejecuta la función **SFFS.SFFS\_wrapper** ( ) para hacer la selección de características en la etapa de valencia utilizando los arreglos antes descritos. Posteriormente, se guarda el resultado en los arreglos *SFS\_features\_train\_3* y *SFS\_features\_test\_3*. De igual forma, se almacenan los índices codificados en la *lista\_index\_cod\_3*.
21. Se realiza la unión temporal de los arreglos *SFS\_features\_train\_3* y *SFS\_features\_test\_3* y se guarda en *SFS\_features\_temp*.
22. Se unen temporalmente los arreglos *targets\_trainHA* y *targets\_testHA* y se guardan en *targets\_temp*.
23. Se emplea la función **FLD.train** ( ) utilizando los arreglos temporales *SFS\_features\_temp* y *targets\_temp*. El resultado se guarda en *matrix\_W\_3*.
24. Se realiza la reducción de *SFS\_features\_temp* utilizando la matriz *matrix\_W\_3* y la operación indicada en (50). El resultado se guarda en *new\_features\_train\_3*.
25. Se ejecuta la función **LDA.train** ( ) utilizando *new\_features\_train\_3* y *targets\_temp* enfocándose en el nivel de valencia. El resultado se guarda en *S\_3*, *m\_k\_3* y *pi\_k\_3*.

26. Se realiza una validación cruzada dejando uno fuera mediante la función **LDA.LOOCV ( )** y se imprime la exactitud obtenida.
27. Se descartan los vectores con clase “Excitación alta” de los arreglos *features\_train* y *features\_test*, así como sus correspondientes clases en *targets\_train* y *targets\_test*. El resultado se guarda en los arreglos *features\_trainLA*, *features\_testLA*, *targets\_trainLA* y *targets\_testLA*.

Tercer nivel del EMDC: Valencia (Excitación Baja)

28. Se ejecuta la función **SFFS.SFFS\_wrapper ( )** para hacer la selección de características en la etapa de valencia utilizando los arreglos antes descritos. Posteriormente, se guarda el resultado en los arreglos *SFS\_features\_train\_4* y *SFS\_features\_test\_4*. De igual forma, se almacenan los índices codificados en la *lista\_index\_cod\_4*.
29. Se realiza la unión temporal de los arreglos *SFS\_features\_train\_4* y *SFS\_features\_test\_4* y se guarda en *SFS\_features\_temp*.
30. Se unen temporalmente los arreglos *targets\_trainLA* y *targets\_testLA* y se guardan en *targets\_temp*.
31. Se emplea la función **FLD.train ( )** utilizando los arreglos temporales *SFS\_features\_temp* y *targets\_temp*. El resultado se guarda en *matrix\_W\_4*.
32. Se realiza la reducción de *SFS\_features\_temp* utilizando la matriz *matrix\_W\_4* y la operación indicada en (50). El resultado se guarda en *new\_features\_train\_4*.
33. Se ejecuta la función **LDA.train ( )** utilizando *new\_features\_train\_4* y *targets\_temp* enfocándose en el nivel de valencia. El resultado se guarda en *S\_4*, *m\_k\_4* y *pi\_k\_4*.
34. Se realiza una validación cruzada dejando uno fuera mediante la función **LDA.LOOCV ( )** y se imprime la exactitud obtenida.
35. Se calcula el logaritmo natural de todos los valores *pi\_k\_x* obtenidos y se almacenan en su misma variable.
36. Se obtienen listas para todos los parámetros generados en cada etapa del esquema EMDC codificados en cuatro grupos de 8 bits. Para ello se hace una transformación de los valores originales en formato de punto flotante de doble precisión (64 bits), a un formato entero de punto fijo Q28 en 32 bits. Los resultados se guardan en *list\_SFFS*, *list\_FLD*, *list\_LDA\_mk*, *list\_LDA\_pik* y *list\_LDA\_S*.
37. Retornar *list\_SFFS*, *list\_FLD*, *list\_LDA\_mk*, *list\_LDA\_pik* y *list\_LDA\_S*.

La función **train\_LOOCV ( )** es la encargada de realizar el entrenamiento completo descrito en la sección 4.4.7.3. Ésta, al igual que la función **train ( )**, recibe como argumento las direcciones de dos archivos en formato csv llamados “features.csv” y “Emotions.csv”. En esta modalidad no es necesario darles un reordenamiento a los datos o separarlos en diversos conjuntos, pues la validación cruzada dejando uno fuera utiliza completamente la información del conjunto original para realizar la selección de características y el entrenamiento de los algoritmos. Finalmente, los parámetros obtenidos pasan por el mismo proceso de codificación mencionado con anterioridad. A continuación, se muestra el pseudocódigo del esquema bajo esta modalidad que, si bien puede parecer muy similar al anterior, existen diferencias clave por el tipo de validación utilizada tal como la ausencia de un tratamiento inicial de los datos y, por consecuencia, la omisión de un conjunto de prueba junto con las operaciones donde era requerido. Dichos cambios se pueden visualizar claramente al comparar lo estipulado en la Tabla 42 con los puntos 2, 5, 6, 12, 13, 19, 20, 26 y 27 de la Tabla 41.

Tabla 42. Pseudocódigo de la función EMDC en la modalidad de entrenamiento completo.

<b>Pseudocódigo</b>	
<b>Train_LOOCV (features_csv, targets_csv)</b>	
<b>train (features_csv, targets_csv)</b>	
1.	Se leen los archivos “features.csv” y “Emotions.csv” alojados en las direcciones <i>features_csv</i> y <i>targets_csv</i> .
2.	Se crean los arreglos <i>features_train</i> y <i>targets_train</i> .
Primer nivel del EMDC: Neutralidad	
3.	Se ejecuta la función <b>SFFS.SFFS_wrapper_LOOCV ( )</b> para hacer la selección de características en la etapa de neutralidad utilizando los arreglos antes descritos. Posteriormente, se guarda el resultado en <i>SFS_features_train_1</i> . De igual forma, se almacenan los índices codificados en la <i>lista_index_cod_1</i> .
4.	Se emplea la función <b>FLD.train ( )</b> utilizando <i>SFS_features_train_1</i> y <i>targets_train</i> . El resultado se guarda en <i>matrix_W_1</i> .
5.	Se realiza la reducción <i>SFS_features_train_1</i> utilizando la matriz <i>matrix_W_1</i> y la operación indicada en (50). El resultado se guarda en <i>new_features_train_1</i> .
6.	Se ejecuta la función <b>LDA.train ( )</b> utilizando <i>new_features_train_1</i> y <i>targets_train</i> enfocándose en el nivel de neutralidad. El resultado se guarda en <i>S_1</i> , <i>m_k_1</i> y <i>pi_k_1</i> .
7.	Se realiza una validación cruzada dejando uno fuera mediante la función <b>LDA.LOOCV ( )</b> y se imprime la exactitud obtenida.
8.	Se descartan los vectores con clase “neutral” del arreglo <i>features_train</i> y <i>targets_train</i> . El resultado se guarda en los mismos arreglos.
Segundo nivel del EMDC: Excitación	
9.	Se ejecuta la función <b>SFFS.SFFS_wrapper_LOOCV ( )</b> para hacer la selección de características en la etapa de excitación utilizando los arreglos antes descritos. Posteriormente, se guarda el resultado en <i>SFS_features_train_2</i> . De igual forma, se almacenan los índices codificados en la <i>lista_index_cod_2</i> .
10.	Se emplea la función <b>FLD.train ( )</b> utilizando <i>SFS_features_train_2</i> y <i>targets_train</i> . El resultado se guarda en <i>matrix_W_2</i> .
11.	Se realiza la reducción <i>SFS_features_train_2</i> utilizando la matriz <i>matrix_W_2</i> y la operación indicada en (50). El resultado se guarda en <i>new_features_train_2</i> .
12.	Se ejecuta la función <b>LDA.train ( )</b> utilizando <i>new_features_train_2</i> y <i>targets_train</i> enfocándose en el nivel de excitación. El resultado se guarda en <i>S_2</i> , <i>m_k_2</i> y <i>pi_k_2</i> .
13.	Se realiza una validación cruzada dejando uno fuera mediante la función <b>LDA.LOOCV ( )</b> y se imprime la exactitud obtenida.
14.	Se descartan los vectores con clase “excitación baja” del arreglo <i>features_train</i> y <i>targets_train</i> . El resultado se guarda en los arreglos <i>features_trainHA</i> , y <i>targets_trainHA</i> .
Tercer nivel del EMDC: Valencia (Excitación Alta)	
15.	Se ejecuta la función <b>SFFS.SFFS_wrapper_LOOCV ( )</b> para hacer la selección de características en la etapa de valencia utilizando los arreglos antes descritos. Posteriormente, se guarda el resultado en <i>SFS_features_train_3</i> . De igual forma, se almacenan los índices codificados en la <i>lista_index_cod_3</i> .
16.	Se emplea la función <b>FLD.train ( )</b> utilizando <i>SFS_features_train_3</i> y <i>targets_trainHA</i> . El resultado se guarda en <i>matrix_W_3</i> .

17. Se realiza la reducción  $SFS\_features\_train\_3$  utilizando la matriz  $matrix\_W\_3$  y la operación indicada en (50). El resultado se guarda en  $new\_features\_train\_3$ .
18. Se ejecuta la función **LDA.train** ( ) utilizando  $new\_features\_train\_3$  y  $targets\_trainHA$  enfocándose en el nivel de valencia. El resultado se guarda en  $S\_3$ ,  $m\_k\_3$  y  $pi\_k\_3$ .
19. Se realiza una validación cruzada dejando uno fuera mediante la función **LDA.LOOCV** ( ) y se imprime la exactitud obtenida.
20. Se descartan los vectores con clase “excitación alta” del arreglo  $features\_train$  y  $targets\_train$ . El resultado se guarda en los arreglos  $features\_trainLA$ , y  $targets\_trainLA$ .

Tercer nivel del EMDC: Valencia (Excitación Baja)

21. Se ejecuta la función **SFFS.SFFS\_wrapper\_LOOCV** ( ) para hacer la selección de características en la etapa de valencia utilizando los arreglos antes descritos. Posteriormente, se guarda el resultado en  $SFS\_features\_train\_4$ . De igual forma, se almacenan los índices codificados en la lista  $index\_cod\_4$ .
22. Se emplea la función **FLD.train** ( ) utilizando  $SFS\_features\_train\_4$  y  $targets\_trainLA$ . El resultado se guarda en  $matrix\_W\_4$ .
23. Se realiza la reducción  $SFS\_features\_train\_4$  utilizando la matriz  $matrix\_W\_4$  y la operación indicada en (50). El resultado se guarda en  $new\_features\_train\_4$ .
24. Se ejecuta la función **LDA.train** ( ) utilizando  $new\_features\_train\_4$  y  $targets\_trainLA$  enfocándose en el nivel de valencia. El resultado se guarda en  $S\_4$ ,  $m\_k\_4$  y  $pi\_k\_4$ .
25. Se realiza una validación cruzada dejando uno fuera mediante la función **LDA.LOOCV** ( ) y se imprime la exactitud obtenida.
26. Se calcula el logaritmo natural de todos los valores  $pi\_k\_x$  obtenidos y se almacenan en su misma variable.
27. Se obtienen listas para todos los parámetros generados en cada etapa del esquema EMDC codificados en cuatro grupos de 8 bits. Para ello se hace una transformación de los valores originales en formato de punto flotante de doble precisión (64 bits), a un formato entero de punto fijo Q28 en 32 bits. Los resultados se guardan en  $list\_SFFS$ ,  $list\_FLD$ ,  $list\_LDA\_mk$ ,  $list\_LDA\_pik$  y  $list\_LDA\_S$ .
28. Retornar  $list\_SFFS$ ,  $list\_FLD$ ,  $list\_LDA\_mk$ ,  $list\_LDA\_pik$  y  $list\_LDA\_S$ .

Como puede observarse, tanto en la Tabla 41 como en la Tabla 42 se realiza un proceso de codificación de los parámetros al finalizar la función, esto se realiza para contar con un formato adecuado que facilite su transmisión al dispositivo mediante el bloque de Comunicación RF. Para ello, se eligió transformar los valores a un formato de punto fijo Q28 en 32 bits, obteniendo así un intervalo dinámico de  $-8 \leq ID \leq 7.999999$  y una resolución de  $3.72529 \times 10^{-9}$ . Dicho  $Q_i$  se elige tras visualizar los valores comunes que arroja el esquema EMDC utilizando características adquiridas con el sistema embebido.

Finalmente, ambas funciones son ensambladas en una sola con el objetivo de permitir la elección de la modalidad (entrenamiento rápido o entrenamiento completo) por medio de los controles señalados en la sección 5.8.1.

Tabla 43. Pseudocódigo de la función EMDC.

<b>Pseudocódigo</b>	
<b>EMDC (features_csv, targets_csv, LOOCV)</b>	
1.	Si LOOCV=True entonces
a.	Ejecutar la función <b>train_LOOCV</b> ( ) enviando <i>features_csv</i> y <i>targets_csv</i> como argumento. El resultado se guarda en <i>list_SFFS</i> , <i>list_FLD</i> , <i>list_LDA_mk</i> , <i>list_LDA_pik</i> y <i>list_LDA_S</i> .
2.	SiNo
a.	Ejecutar la función <b>train</b> ( ) enviando <i>features_csv</i> y <i>targets_csv</i> como argumento. El resultado se guarda en <i>list_SFFS</i> , <i>list_FLD</i> , <i>list_LDA_mk</i> , <i>list_LDA_pik</i> y <i>list_LDA_S</i> .
3.	Retornar <i>list_SFFS</i> , <i>list_FLD</i> , <i>list_LDA_mk</i> , <i>list_LDA_pik</i> y <i>list_LDA_S</i> .

## 6 Resultados y análisis de resultados

El funcionamiento del dispositivo sobre una aplicación en específico queda fuera de los alcances de esta tesis debido a la dificultad que conlleva la búsqueda de sujetos de prueba para el estudio, la elaboración de un procedimiento que garantice el aislamiento de las emociones evocadas por el individuo, así como las complicaciones que involucran la selección y validación de los estimuladores por utilizar, junto con el acondicionamiento de un entorno que nos permita obtener la información requerida sin interferencia de fuentes externas. Es por estos motivos que no se realiza un estudio del sistema en conjunto para la clasificación de emociones, y en su lugar se verifican a continuación únicamente los resultados entregados por cada uno de los subsistemas que conforman al detector de emociones y al sistema en la computadora, comprobando con ello su adecuado funcionamiento de forma individual y, a la vez, evidenciando la posible integración de todo el sistema en conjunto para permitir su uso en otras aplicaciones.

### 6.1 Comunicaciones

Para verificar el funcionamiento adecuado del subsistema de Comunicaciones se analizan por separado los elementos descritos en la Figura 46, es decir, la comunicación del bloque I2C con el sensor biométrico AS7026GG y la comunicación del bloque UART con el módulo HM10.

Dentro del bloque I2C, se analiza la comunicación del procesador digital de señales con el sensor biométrico AS7026GG a través de la trama de datos y reloj del protocolo I2C ilustradas en la Figura 77. En ésta se observa la correcta escritura al registro 0xFE del sensor biométrico, cuya dirección es 0x30, con los tiempos de levantamiento y de caída a 280 ns y 60 ns, respectivamente, acorde a las especificaciones que dicta el protocolo I2C en [73] para una tasa de 400 kbps.

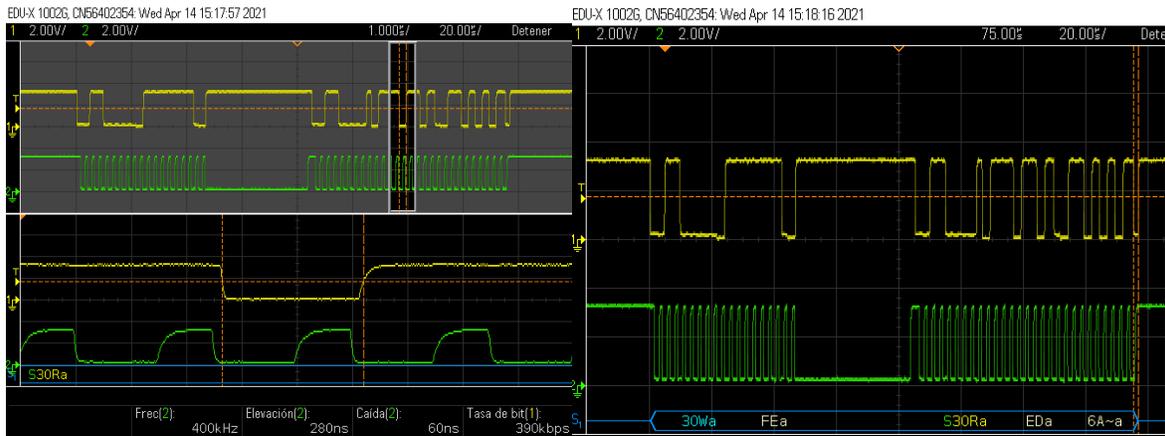


Figura 77. Trama de datos (amarillo) y reloj (verde) del protocolo I2C con el sensor biométrico.

A su vez, la adecuada configuración del sensor se comprueba con los diversos acondicionamientos que sufre la señal adquirida de la PPG mediante las etapas del TIA, el preamplificador, previo al OFE y posterior al OFE en el sensor AS7026GG como se observa en la Figura 78. Siguiendo la secuencia de etapas ilustradas en el diagrama de la Figura 50 y la manera como se configuraron en la sección 4.4.2.2, se comienza la adquisición de la señal obtenida mediante PPG a través del TIA, seguido de un primer acondicionamiento de la señal a través del filtro antialias a 826 Hz y del filtro paso altas a 200 Hz para eliminar fuentes de luz indeseadas, sin embargo, la amplificación x2 de la señal introduce ruido en ésta. Posterior a ello, en la etapa previa a la OFE, se realiza la demodulación PAM de la señal junto con el filtro paso bajas a 10 Hz que conlleva un suavizado de la señal al atenuar las componentes de alta frecuencia, aunque aún presenta las variaciones de la línea base. Finalmente, posterior a la OFE, se aplica el filtro paso altas a 0.33 Hz para reducir en cierta medida las variaciones de la línea base junto con una amplificación x8 de la señal. Cabe mencionar que la línea base que presenta

esta última señal posterior a la OFE se debe a las condiciones del proceso de calibración establecidas

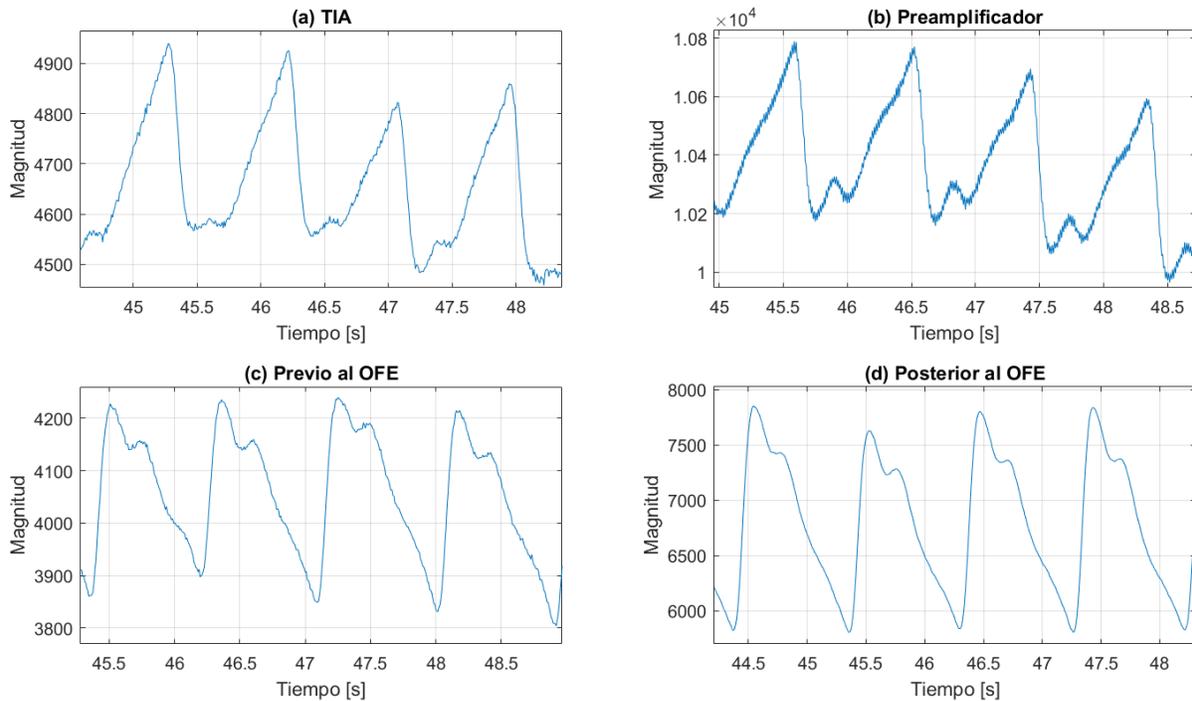


Figura 78. Acondicionamiento de señal obtenida mediante PPG a través del (a) TIA, (b) el preamplificador, (c) previo al OFE y (d) posterior al OFE en el sensor biométrico AS7026GG.

para permitir una correcta lectura de ésta a través del ADC del sensor. Así pues, se observa el cumplimiento de la calibración, descrita en la sección 4.4.2, al adquirirse una señal de pre-ganancia con valores entre 5120 y 12288 y una señal posterior al OFE con amplitudes entre 1200 y 3500.

Por otro lado, se obtuvieron resultados desfavorables para la adquisición de la señal EDA atribuidos a la falta de acondicionamiento durante la etapa de diseño de la PCB del sensor biométrico. Esto debido a que en ésta no se cuenta con una etapa de amplificación, ni un filtro paso altas que elimine la línea base de la señal. Por esta razón puede observarse dentro de la señal EDA de la Figura 79 un comportamiento inesperado correspondiente a un ascenso constante que tiende a adquirir el valor máximo del ADC, que corresponde a los 1.9 V suministrados al circuito, junto con leves variaciones debidas a movimientos corporales. La tendencia anterior se atribuye a la escasa corriente que se suministra al cuerpo a través de los electrodos, lo cual conlleva a que la señal captada tienda a un circuito abierto con el valor de 1.9 V de la alimentación. A su vez, el ruido observado en la señal EDA se atribuye a inductancias parásitas introducidas por la traza y vía de la señal sobre la PCB hasta introducirse por el pin GPIO0 del sensor biométrico.

En cuanto al bloque UART, para su correcta comunicación con el módulo HM10 y el subsistema de configuración, se comprueba la recepción de diversos parámetros por parte del sistema en la computadora. Para ello se escogen algunos vectores requeridos para el subsistema de clasificación, siendo estos el vector de medias y el vector de valores a priori. Para la estimación de los anteriores se utiliza

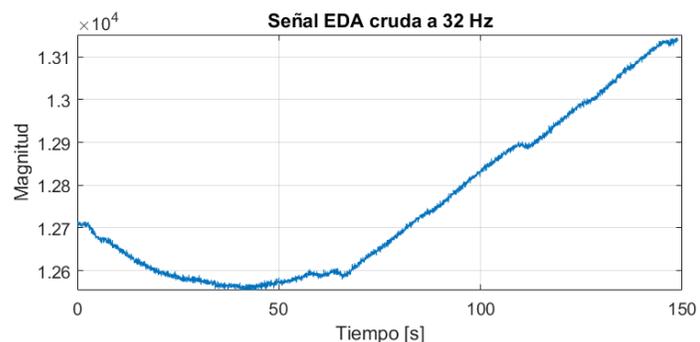


Figura 79. Comportamiento de señal EDA cruda a 32 Hz.

un entrenamiento rápido con los archivos de prueba “features.csv” y “targets.csv” ubicados dentro de la raíz “Python/Emotion\_Detector\_PY/ML” dentro del repositorio disponible a través del código QR de la Figura 63.

Los resultados de los cálculos estimados mediante el sistema en la computadora y aquellos recibidos en el procesador digital de señales se aprecian en la Figura 80. De los datos recibidos puede observarse la similitud con los valores transmitidos con errores inferiores al 0.05%, considerando

Expression	Value
▼ Cov_S	[3.94135714e-06,
[0]	3.94135714e-06
[1]	6.25103712e-06
[2]	5.48101962e-05
[3]	7.17863441e-06
Expression	Value
▼ MeanVect	[0.00714813545,
[0]	0.00714813545
[1]	0.00832680985
[2]	-0.0974590704
[3]	-0.0960213169
[4]	-0.254900187
[5]	-0.249911919
[6]	-0.0839577168
[7]	-0.082966581

```

Console 1/A X
S1=[[3.94205981e-06]]
S2=[[6.2511077e-06]]
S3=[[5.48131914e-05]]
S4=[[7.18216934e-06]]

```

```

Console 1/A X
mk1=[[0.00714814]
[0.00832681]]
mk2=[[ -0.09745907]
[ -0.09602132]]
mk3=[[ -0.25490013]
[ -0.24991213]]
mk4=[[ -0.08395772]
[ -0.08296658]]

```

como dato correcto el estimado en la computadora. Esta discrepancia entre ambos resultados es atribuida a la conversión que sufren los datos antes de su transmisión a través del protocolo BLE ya que, como se menciona en la sección 5.8.6, los datos estimados con una aritmética de punto flotante de doble precisión pasan a tomar una aritmética de punto fijo Q28 a 32 bits, luego son transmitidos, recabados y convertidos nuevamente a punto flotante de precisión simple mediante el procesador digital de señales. Por esta razón, resulta importante la justificación dada en la sección 5.8.6 respecto a la elección de la aritmética de punto fijo Q28 para garantizar la correcta representación de los resultados adquiridos para los parámetros asociados al subsistema de clasificación.

Figura 80. (Izquierda) Datos recibidos por el detector de emociones (Derecha) Datos transmitidos por el sistema en la computadora.

## 6.2 Acondicionamiento de la señal

Utilizando los filtros FIR, especificados en la sección 5.3, se obtienen los siguientes resultados tras introducir un total de 3900 muestras de la señal obtenida a partir de la PGG, como se ilustra en la Figura 81, y 930 muestras de la señal EDA, mostrado en la Figura 83. Cabe resaltar que, como se indicó en la sección antes especificada, al ser filtros FIR de orden 200 existe un retraso de 100 muestras en cada una de las señales filtradas.

### 6.2.1 Filtrado de la señal

Como puede observarse, en el caso de la señal adquirida mediante PPG se realiza un suavizado producto de la eliminación de las componentes de alta frecuencia por encima de los 3.33 Hz, lo cual se visualiza con mayor detalle en la Figura 82. Sin embargo, cabe recordar que el sensor biométrico proporciona un prefiltrado interno comprendido entre 0.33 y 10 Hz. Asimismo, existe una disminución en las variaciones generadas por la línea base, lo cual puede observarse en la Figura 81, que, si bien no resulta en una eliminación completa, disminuye su magnitud en gran medida. Cabe resaltar que durante el diseño del filtro se tuvo especial cuidado en seleccionar una frecuencia de corte inferior que contemple un ritmo cardiaco mínimo de 40 bpm, por lo que incrementar dicha frecuencia para atenuar aún más la distorsión por la línea base podría disminuir el desempeño del dispositivo cuando el usuario presente un ritmo cardiaco bajo.

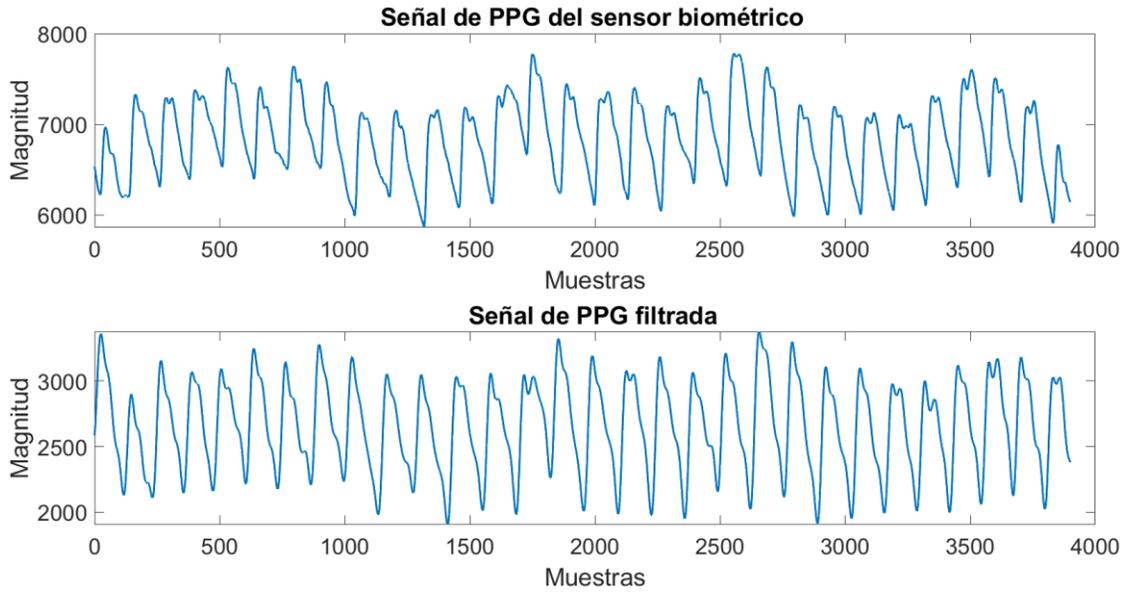


Figura 81. Ejemplo de la señal eléctrica obtenida a partir de PPG con filtrado implementado

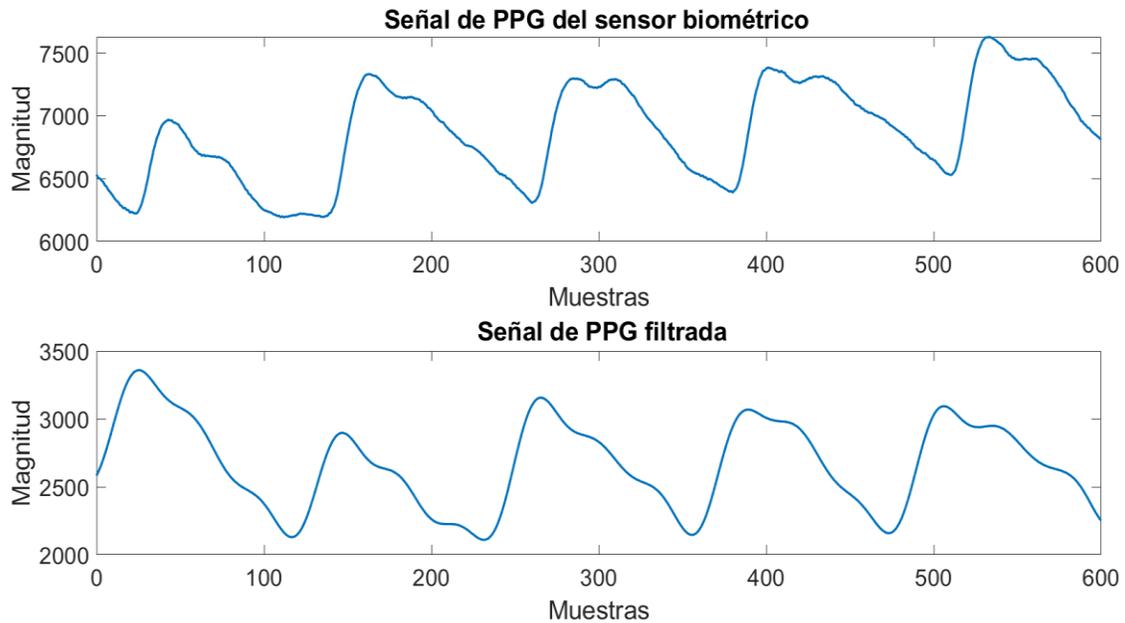


Figura 82. Suavizado en la señal obtenida a partir de PPG

Por otro lado, a raíz de no contar con datos correctos de la señal EDA por parte del sensor biométrico debido a las razones especificadas en la sección 6.1, se opta por utilizar las muestras de una base de datos denominada *Stress Recognition in Automobile Drivers (drivedb)* que puede ser encontrada en [80]. En la Figura 83 es posible observar un suavizado de la señal al excluir el ruido de alta frecuencia situado por encima de los 2.16 Hz. Cabe recordar que no se realiza una eliminación de la línea base debido a que el método utilizado para obtener la componente SCR ya realiza dicha tarea.

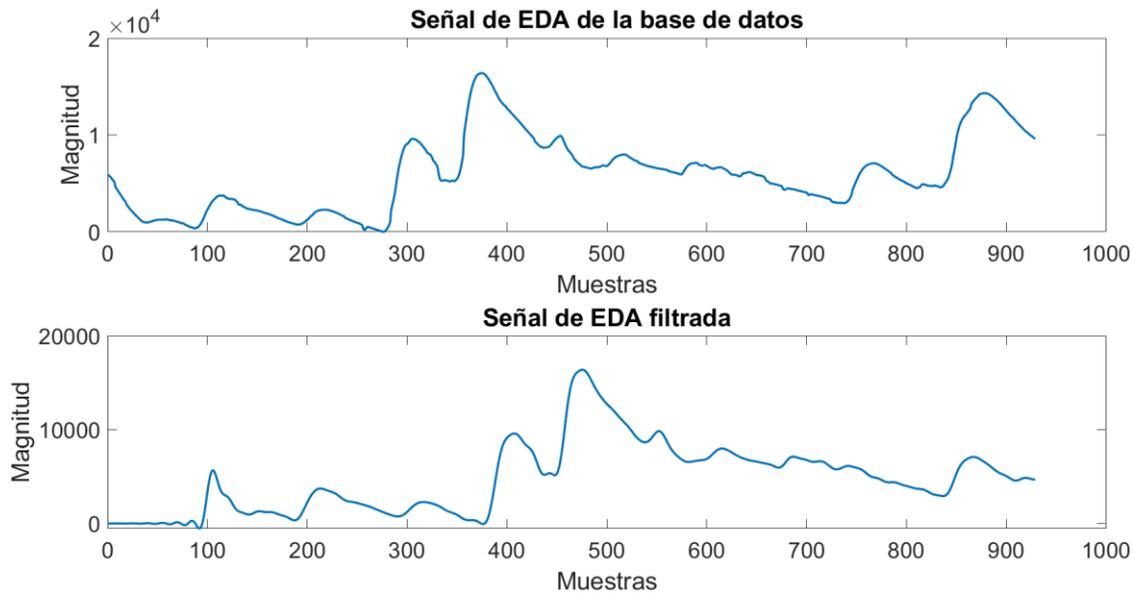


Figura 83. Ejemplo de la señal eléctrica EDA filtrada

### 6.2.2 Relación señal a ruido (SNR)

Tal como se mencionó en la sección 4.2.2, una de las especificaciones objetivo que se busca alcanzar implementando los filtros digitales FIR es una relación señal a ruido (SNR, por sus siglas en inglés) de aproximadamente 50 dB.

Para realizar el análisis se toman en consideración las muestras adquiridas por el sensor biométrico y la base de datos de la sección 6.2.1, así como sus contrapartes filtradas. El proceso para obtener la SNR se realiza mediante una comparación de la potencia de la señal de origen (filtrada y sin filtrar) y la de su ruido correspondiente. En el caso de la señal adquirida mediante PPG se utiliza una metodología especial, ya que se presentaron dificultades al hacer una comparación directa entre la SNR de la señal original y de aquella donde está implementado el filtro paso banda. Los inconvenientes surgen debido a la naturaleza propia de la señal adquirida pues, como se vio en secciones pasadas, ésta se encuentra montada sobre una línea base que puede considerarse ruido de baja frecuencia y gran amplitud, el cual se atenúa tras implementar el filtro. Esto provoca que la señal filtrada tenga un cambio brusco en su potencia respecto a la original, afectando la obtención de la SNR debido a la diferencia de amplitudes. Es por este motivo que para dicha señal se realiza su obtención enfocándose únicamente en el ruido de alta frecuencia, esto con el objetivo de no afectar la potencia de la señal filtrada tras la atenuación de la línea base.

Para realizar el cálculo de las SNR, a cada señal sin filtrar se le aplica un filtro paso altas con frecuencias de corte de 2.16 Hz en el caso de la EDA y 3.58 Hz para la señal obtenida a partir de la PPG, las cuales coinciden con las frecuencias de corte superiores señaladas en la sección 5.3. Dicha acción se realiza con el objetivo de obtener una señal de ruido de alta frecuencia de las señales sin acondicionamiento. Más adelante, se acondicionan las señales sin filtrar por medio de filtros paso bajas utilizando las mismas frecuencias y, finalmente, éstas se pasan por el filtro paso altas mencionado con anterioridad para obtener sus correspondientes señales de ruido tras el filtrado. Finalmente, se utiliza la herramienta `snr(X,Y)` de MATLAB, donde  $X$  es la señal a analizar y  $Y$  el ruido esperado. Dicha función retorna la SNR en decibeles que, en conjunto con las señales antes obtenidas, arroja los siguientes resultados.

$$SNR_{PPG\_crudo} = 38.4945 \text{ dB}$$

$$SNR_{PPG\_filtrado} = 52.2429 \text{ dB}$$

$$SNR_{EDA\_crudo} = 32.9664 \text{ dB}$$

$$SNR_{EDA\_filtrado} = 54.3537 \text{ dB}$$

### 6.3 Extracción de características

La verificación adecuada de este subsistema se realiza mediante el análisis de la señal PRV generada a través de la señal obtenida mediante PPG, con la que se estiman sus características en el dominio del tiempo y mediante el método no lineal de Poincaré, así como su interpolación mediante el método de trazadores cúbicos para poder estimar sus características en el dominio de la frecuencia gracias a la PSD. Asimismo, se analiza la adecuada obtención de las características en el dominio del tiempo y la frecuencia para la señal EDA a través de su señal derivada SCR.

#### 6.3.1 Dominio del tiempo y gráfica de Poincaré de PRV

Con el objetivo de evaluar el desempeño de este bloque, se realiza la estimación de los IPP y características derivadas de los mismos, tanto en el programa matemático MATLAB como en el DSP, empleando el segmento de la señal de PPG ilustrado en la Figura 84.

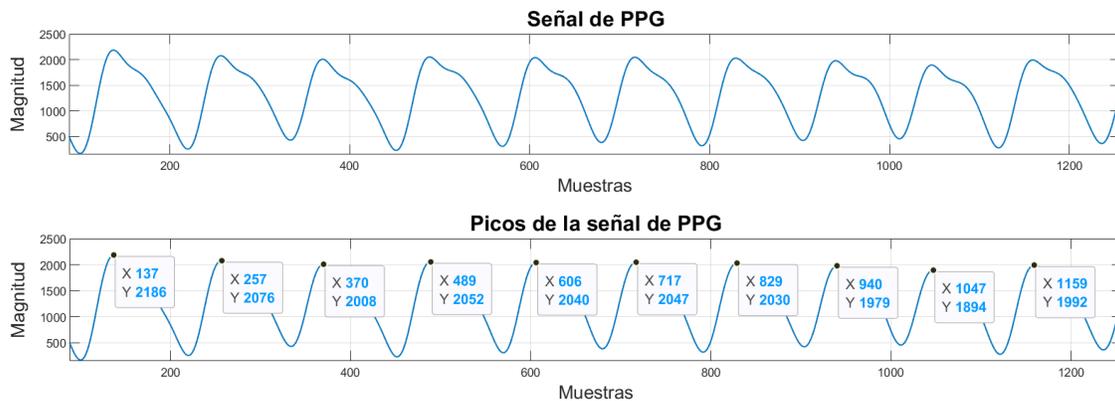


Figura 84. Ejemplo de señal de PPG obtenida a partir del sensor biométrico.

En la Tabla 44 se enlistan los valores calculados mediante MATLAB con una precisión aritmética de punto flotante a 64 bits, los cuales son comparados con los resultados de la Figura 85, obtenidos a partir del DSP con una aritmética de punto flotante de precisión simple, en donde es posible observar que éstos presentan un error menor al 0.01%.

Tabla 44. IPP y características obtenidas a partir de MATLAB.

Intervalos		Características	
1	0.9375	pNN50	0.125
2	0.8828125	NN50	1
3	0.9296875	PPImed	0.887152777777778
4	0.9140625	SDNN	0.033171195840773
5	0.8671875	RMSSD	0.035801372616843
6	0.875	SD1_SD2	1.409787217986912
7	0.8671875	SD1	0.037233989548235
8	0.8359375	SD2	0.026411070460266
9	0.875		

IPP	float[9]	[0.9375,0.8828125,0.92...	Feat_Val	{pNN50=0.125,NN50=1.0,...
[0]	float	0.9375	pNN50	0.125
[1]	float	0.8828125	NN50	1.0
[2]	float	0.9296875	PPImed	0.887152791
[3]	float	0.9140625	SDNN	0.0331716947
[4]	float	0.8671875	RMSSD	0.0358013734
[5]	float	0.875	SD1_SD2	1.4096489
[6]	float	0.8671875	SD1	0.0372303315
[7]	float	0.8359375	SD2	0.0264110658
[8]	float	0.875		

Figura 85. IPP y características obtenidas mediante el MCU.

Para verificar la efectividad del algoritmo ante diversas situaciones, se presentan dos casos que al no ser detectados afectan el desempeño del sistema. Para esto, se ha optado por modificar de dos formas distintas el segmento presentado en la Figura 84 y calcular mediante el DSP los intervalos y características, con el propósito de comparar los resultados y verificar que, a pesar de la presencia de anomalías, el algoritmo arroja los valores correctos. En la gráfica superior de la Figura 86 se ilustra el segmento en el cual se ha omitido un pico sistólico, lo que resulta en un intervalo de longitud incorrecta. Por otro lado, en la gráfica inferior se ha añadido una protuberancia, la cual podría ser interpretada como un pico falso.

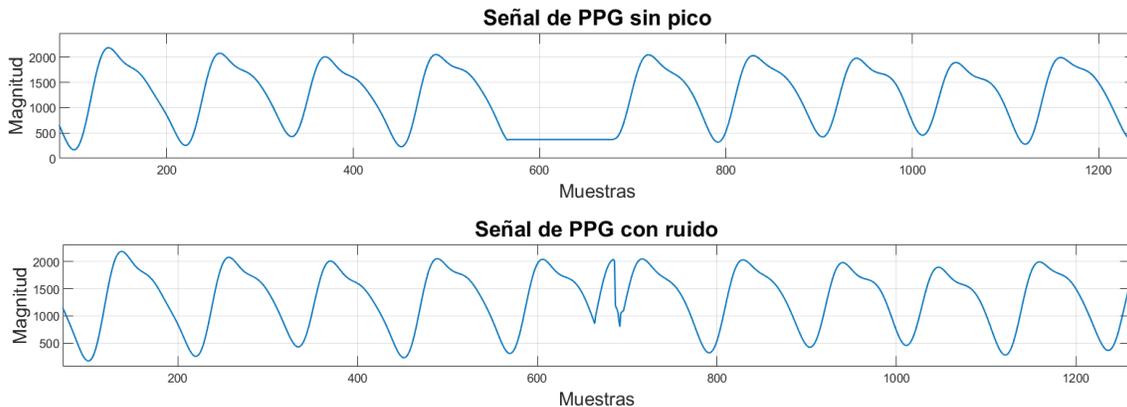


Figura 86. Señales de PPG con anomalías.

a) IPP de la señal de PPG sin pico

IPP	float[9]	[0.9375,0.8828125,0.9296...
[0]	float	0.9375
[1]	float	0.8828125
[2]	float	0.9296875
[3]	float	0.875
[4]	float	0.8671875
[5]	float	0.8359375
[6]	float	0.875
[7]	float	0.0
[8]	float	0.0

b) IPP de la señal de PPG con ruido

IPP	float[9]	[0.9375,0.8828125,0.92...
[0]	float	0.9375
[1]	float	0.8828125
[2]	float	0.9296875
[3]	float	0.9140625
[4]	float	0.8671875
[5]	float	0.875
[6]	float	0.8671875
[7]	float	0.8359375
[8]	float	0.875

Figura 87. IPP de las señales de PPG con anomalías.

Los resultados generados por el DSP se muestran en la Figura 87, donde para ambos casos el algoritmo genera resultados correctos, ya que como puede observarse en la Figura 87.a, se ha detectado la ausencia del pico, por lo que únicamente se obtienen los IPP dentro del intervalo aceptable. Por otro lado, en la Figura 87.b se demuestra que el algoritmo identifica la protuberancia como un pico falso y los intervalos corresponden uno a uno con los estimados a partir de la señal sin anomalías.

### 6.3.2 Interpolación de datos PRV

A partir de la adquisición de 256 s de datos PRV adquiridos mediante el DSP, se realiza su respectiva interpolación por medio del método de trazadores cúbicos, siguiendo el procedimiento descrito en la sección 4.4.4.5, para adquirir la misma señal a 8 Hz como se ilustra en la Figura 88. Con el objetivo de verificar el funcionamiento adecuado de este bloque se exportan los datos PRV obtenidos del DSP al programa matemático MATLAB con el cual se realiza la misma interpolación empleando la función `spline(x,y,xq)` donde `x` y `y` son las abscisas y ordenadas de los datos por interpolar y `xq` un vector con los valores de las nuevas abscisas por interpolar.

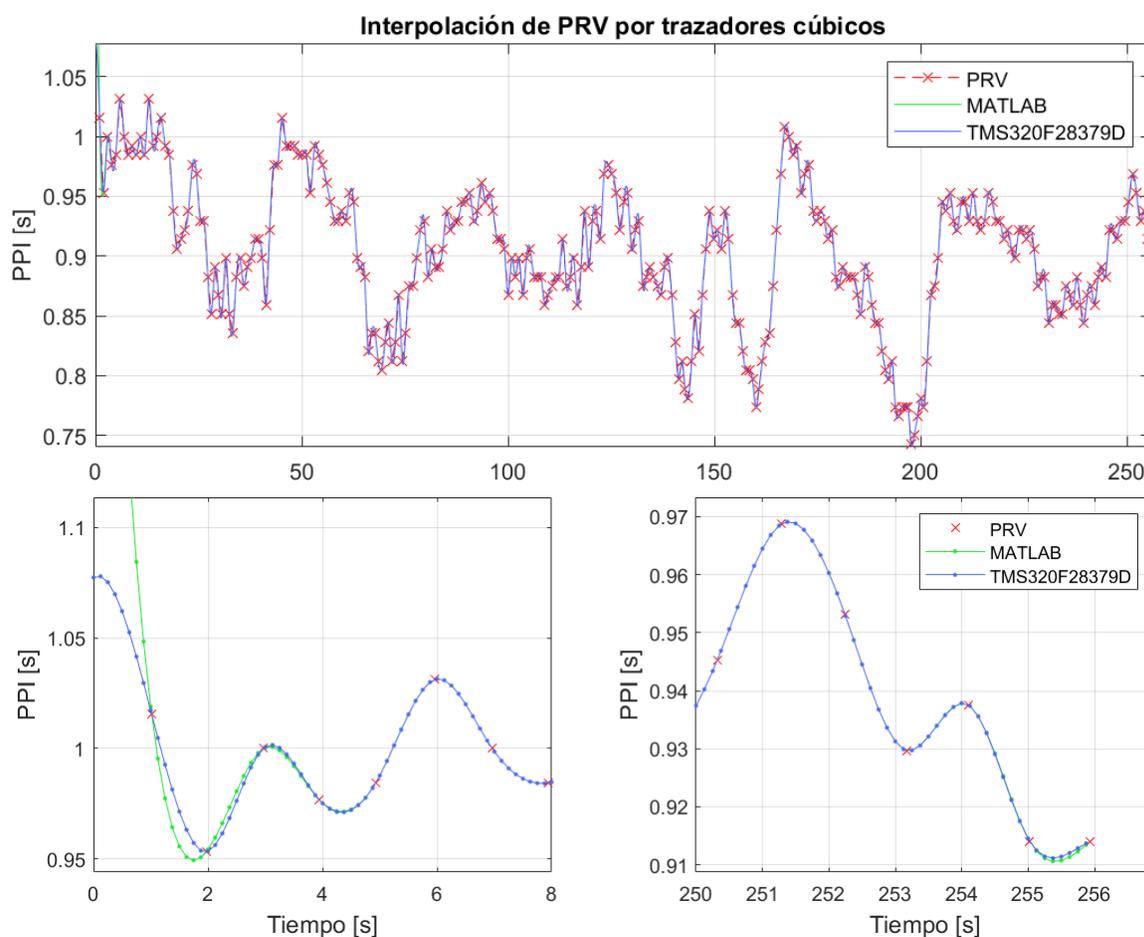


Figura 88. Interpolación de PRV por trazadores cúbicos

De la comparación entre los resultados obtenidos con el DSP y MATLAB, los cuales pueden observarse en la Figura 88, se vislumbra prácticamente la misma interpolación de los datos, salvo la discrepancia que se presenta al inicio y al final de la señal dado que la terminación empleada por la función en MATLAB resulta ser una de condición extrapolada (not a knot), a diferencia de la terminación natural empleada en el programa del DSP. Excluyendo este caso particular, el resto de las interpolaciones presenta un error menor al 0.01% en los resultados a causa de la diferencia de precisión aritmética con la que cuenta MATLAB al manejar operaciones de punto flotante a 64 bits.

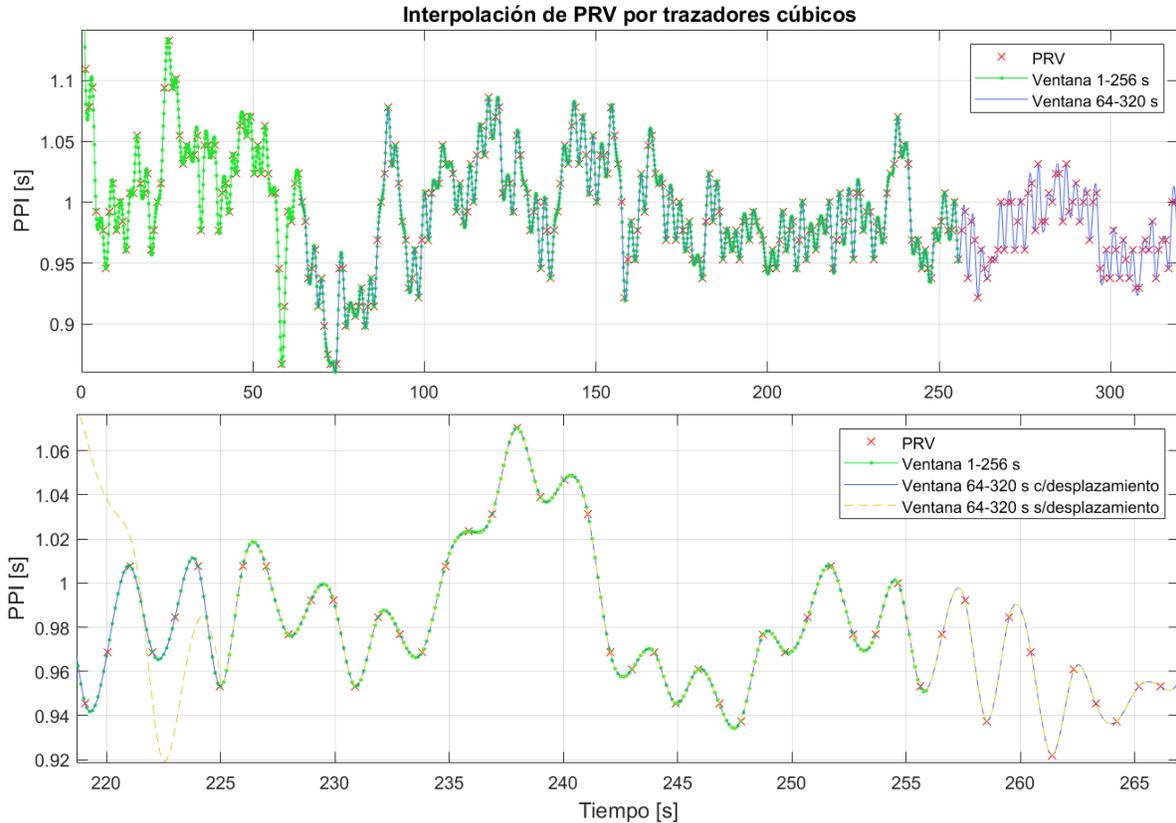


Figura 89. (Arriba) Metodología de desplazamiento de los datos PRV interpolados para conformar dos ventanas de datos de 256 s. (Abajo) Funcionamiento del límite de sustitución hacia atrás en la interpolación de los datos superiores a 225 s.

A su vez, para verificar la metodología del desplazamiento de datos PRV interpolados y el funcionamiento del límite de la sustitución hacia atrás en el algoritmo de trazadores cúbicos, ambos puntos descritos previamente en la sección 4.4.4.5, se extraen 320 s de datos PRV con los que se ejecuta dos veces las estimaciones en frecuencia. A través de la gráfica superior de la Figura 89 se comprueba el desplazamiento de los datos PRV dado que se logran obtener dos ventanas contiguas de 256 s estimadas por el detector de emociones, donde la ventana de 64-320 s se conforma por los mismos datos de los segundos 64-256 junto con los nuevos datos de los segundos 256-320. A su vez, en la gráfica inferior de la Figura 89 se ilustran las gráficas PRV y de ambas ventanas, en compañía de una adicional correspondiente a la misma ventana de 64-320 s pero sin realizarse la metodología de desplazamiento de datos. Esto último se realiza a fin de vislumbrar el instante en el que el algoritmo comienza por interpolar los datos hacia atrás hasta detenerse en el segundo 225 debido a que se cumple con el límite establecido de la sustitución hacia atrás, tras el cual se mantienen las interpolaciones realizadas previamente en la ventana anterior. De esta forma se logra una reducción en el procesamiento del detector al evitar realizar nuevamente la interpolación de los datos en los segundos 64-225.

### 6.3.3 Obtención de SCR

Con el propósito de verificar la correcta extracción de la componente SCR de la señal de EDA, ésta es generada tanto en MATLAB con una precisión aritmética de punto flotante de 64 bits, como en el DSP con una aritmética de punto fijo de 16 bits en Q2, ambas mediante el método descrito en la sección 4.4.4.2. Lo anterior se realiza empleando la señal de EDA filtrada por el subsistema de acondicionamiento de la señal, la cual se muestra en la Figura 83.

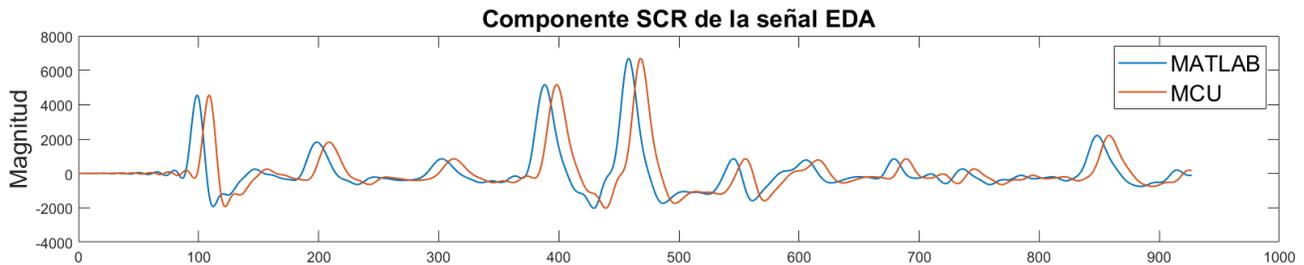


Figura 90. Componente SCR de la señal de EDA obtenida mediante el programa MATLAB (azul) y el MCU (naranja).

A partir de la Figura 90 es posible observar que ambas señales coinciden en amplitud, sin embargo, la señal obtenida mediante el DSP presenta un retraso de 16 muestras generado por la naturaleza de la operación convolución, lo cual corresponde a la mitad de los puntos de la ventana Bartlett.

La señal de MATLAB no presenta dicho retraso debido al tratamiento que este software realiza sobre la misma. Además, en ambos segmentos se ha removido exitosamente la componente SCL, lo cual se comprueba al observar la oscilación de la señal respecto a cero, y con ello, la ocurrencia de los ER-SCR. Lo anterior coincide con la señal ilustrada en la Figura 56 de la sección 4.4.4.2, con lo que se comprueba el buen desempeño del bloque que realiza la obtención de la componente SCR.

### 6.3.4 Dominio del tiempo de SCR

Con el objetivo de evaluar el desempeño del bloque correspondiente al dominio del tiempo de EDA, se realiza la detección de los SCR y estimación de las características derivadas de los mismos, tanto en MATLAB como en el DSP, empleando las componentes SCR ilustradas en la Figura 90.

En la Tabla 45 se enlistan los valores calculados mediante MATLAB con una precisión aritmética de punto flotante a 64 bits, los cuales son comparados con los resultados de la Figura 91, obtenidos a partir del DSP con una aritmética de punto flotante de precisión simple, en donde es posible observar que éstos presentan un error menor al 0.01%.

Feat_Val	struct... {NumSCR=8.0,AmpSCR=2403.0,mTL=...	Amplitude	float[8] [781.25,844.5,847.25,850.25,1814.0...]
NumSCR	float 8.0	[0]	float 781.25
AmpSCR	float 2403.0	[1]	float 844.5
mTL	float 0.580645144	[2]	float 847.25
sdAmp	float 2280.59814	[3]	float 850.25
ctl25	float 845.875	[4]	float 1814.0
ctl50	float 1332.125	[5]	float 2213.5
ctl75	float 3694.0	[6]	float 5174.5
ctl90	float 6698.75	[7]	float 6698.75

Figura 91. Amplitudes y características obtenidas mediante el MCU.

Tabla 45. Amplitudes y características obtenidas a partir de MATLAB.

Amplitudes		Características	
1	781.3684210526314	NumSCR	8
2	844.7368421052631	AmpSCR	2403.25
3	847.4736842105262	mTL	0.580645161290323
4	850.5263157894735	sdAmp	2280.644011294282
5	1814.210526315790	ctl25	846.1052631578947
6	2213.789473684210	ctl50	1332.368421052632
7	5174.842105263157	ctl75	3694.315789473683
8	6699.052631578948	ctl90	6699.052631578948

### 6.3.5 Densidad espectral de potencia

De forma similar al procedimiento realizado en la sección anterior, para verificar el funcionamiento de la estimación de la PSD se exportan los datos interpolados PRV por trazadores cúbicos en el DSP al software MATLAB para realizar la misma estimación del periodograma a través de la función `periodogram(x,window,nfft,fs)` donde `x` es la señal por analizar, `window` el ventaneo utilizado (Hanning), `nfft` el número de puntos para realizar la FFT y `fs` la frecuencia de muestreo de la señal. Al comparar los resultados de la Figura 92 obtenidos con ambos elementos, se identifica la estimación de las componentes de la PSD con errores por debajo del 0.02% por las mismas razones de precisión mencionadas para la interpolación.

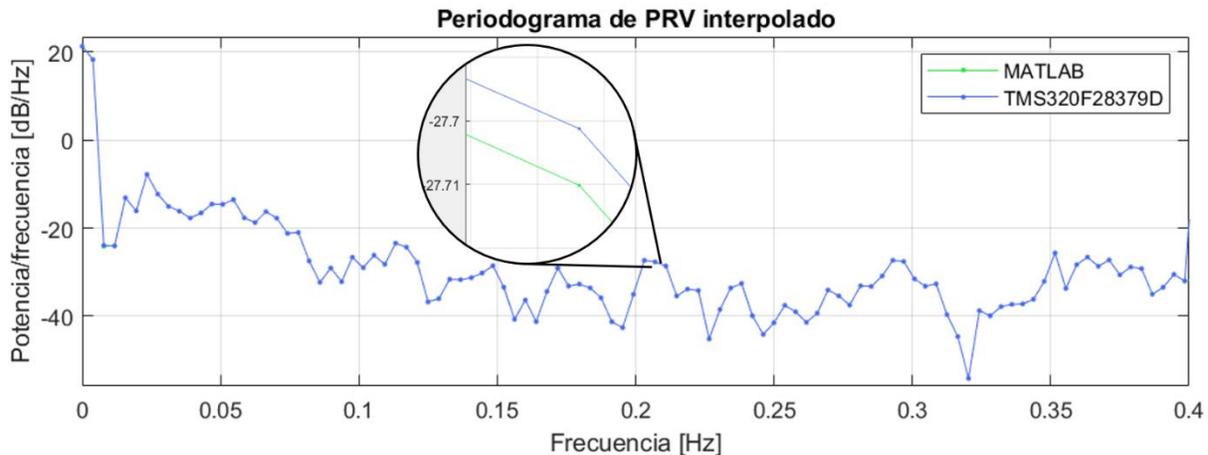


Figura 92. PSD de la señal PRV interpolada mediante el método del periodograma.

## 6.4 Clasificación

Con el objetivo de verificar el buen funcionamiento del subsistema de clasificación, implementado en la sección 5.5, se realiza una prueba utilizando los parámetros generados en el ejemplo de la sección 6.5.6, los cuales son adquiridos por el sistema en la computadora tras un entrenamiento bajo el esquema EMDC empleando un entrenamiento rápido. Cabe resaltar que la prueba elaborada en este apartado únicamente verifica paso a paso el correcto desarrollo del algoritmo pues, como se especifica en la sección 4.2.4, para evaluar el desempeño del sistema en cuanto a la clasificación de emociones es necesario adquirir muestras adjudicadas a cada emoción tras realizarse un estudio en un ambiente controlado, con sujetos de prueba y una aplicación en específico, tomando en consideración la implicación especificada en la sección 3.7.3. A continuación, se muestran los parámetros ingresados al subsistema, así como los resultados en sus etapas intermedias y final.

En primer lugar, se realiza una selección de características en cada etapa del esquema EMDC de acuerdo con la codificación establecida en la sección 4.4.1 para los registros “Caract PRV 1”, “Caract PRV 2”, “Caract EDA 1” y “Caract EDA 2”. Cabe resaltar que las selecciones para las señales fisiológica se agrupan en un vector de ocho elementos cuyos primeros cuatro son respecto a la señal PRV para cada una de las etapas del esquema EMDC, y los últimos cuatro son respectivos a la señal EDA nuevamente para cada una de las etapas del esquema. Cabe mencionar que cada elemento une en un solo valor de 16 bits las selecciones realizadas sobre las características a realizar para una determinada señal fisiológica y etapa correspondiente.

Asig_car	unsigned int[8]	[2302,1664,1608,2256,254...]
[0]	unsigned int	0x08FE (Hex)
[1]	unsigned int	0x0680 (Hex)
[2]	unsigned int	0x0648 (Hex)
[3]	unsigned int	0x08D0 (Hex)
[4]	unsigned int	0x00FE (Hex)
[5]	unsigned int	0x0004 (Hex)
[6]	unsigned int	0x0020 (Hex)
[7]	unsigned int	0x0106 (Hex)

Figura 93. Selecciones realizadas por la SFFS

Asimismo, los parámetros respectivos a los algoritmos FLD y LDA son ordenados de acuerdo con la etapa del esquema EMDC que le corresponda, tal como se observa en la Figura 94 y la Figura 95, de modo que el valor [0] y [1] pertenecen a la primera etapa, [2] y [3] a la segunda y así sucesivamente. La excepción a esta regla son las matrices de reducción y de covarianza, las cuales son un único vector por etapa.

FLD_W	float[4][21]	[[0.0182356127,-1.09588918e-0...
[0]	float[21]	[0.0182356127,-1.09588918e-0...
[1]	float[21]	[0.359337389,-0.000518818735,...]
[2]	float[21]	[0.0535527915,0.000988460029...
[3]	float[21]	[0.0527725443,-0.00363595854,...]

Figura 94. Matrices de reducción obtenidas por el FLD

Cov_S	float[4]	[1.51295103e-06,5.18391835e-...
[0]	float	1.51295103e-06
[1]	float	5.18391835e-06
[2]	float	6.78382639e-05
[3]	float	9.22779327e-06

ApriVect	float[8]	[-0.261364758,-1.46967602,-0.5...
[0]	float	-0.261364758
[1]	float	-1.46967602
[2]	float	-0.507943809
[3]	float	-0.920629144
[4]	float	-0.693147182
[5]	float	-0.693147182
[6]	float	-0.412685364
[7]	float	-1.08432639

MeanVect	float[8]	[0.0371466018,0.0375989787,0...
[0]	float	0.0371466018
[1]	float	0.0375989787
[2]	float	0.068839103
[3]	float	0.070031397
[4]	float	0.0474749692
[5]	float	0.0536482483
[6]	float	0.000857300009
[7]	float	0.00213074009

Figura 95. Parámetros de LDA

Una vez establecidos los parámetros del subsistema, se introduce un vector de características con los valores indicados en la Figura 96 y, tras ejecutar el algoritmo, se obtiene una clasificación por cada etapa del esquema EMDC, tal como se observa en la Figura 97. La primera clasificación arroja un 0 en “prediction\_1”, indicando que la muestra pertenece a la clase “no neutro”; 0 en “prediction\_2”, denotando una excitación negativa y, finalmente, 1 en “prediction\_4”, señalando que tiene valencia positiva. El resultado final se ilustra en la Figura 98, lo que indica una clasificación dentro de la clase de Excitación baja – Valencia positiva (EB-VP), de acuerdo con la codificación mostrada en la

sección 4.4.1 dentro del registro “Emoción”. Asimismo, durante la prueba del algoritmo se hace una medición del tiempo de ejecución, arrojando un resultado de 2.3025 ms al clasificar el vector.

prediction_1	unsigned int	0
prediction_2	unsigned int	0
prediction_3	unsigned int	0
prediction_4	unsigned int	1

Figura 97. Clasificación realizada en cada etapa del esquema EMDC

prediction	int	2
------------	-----	---

Figura 98. Resultado obtenido mediante el subsistema de Clasificación

vect_feat	float[21]	[0.238866389,59.0,1.02249241,0...
[0]	float	0.238866389
[1]	float	59.0
[2]	float	1.02249241
[3]	float	0.0814883634
[4]	float	0.0461926311
[5]	float	3.37243795
[6]	float	0.700834155
[7]	float	-11.7330437
[8]	float	-22.4973888
[9]	float	-32.100872
[10]	float	0.110361852
[11]	float	0.0327246487
[12]	float	10.0327244
[13]	float	8.0
[14]	float	7195.625
[15]	float	0.532258093
[16]	float	5794.0376
[17]	float	16653.0
[18]	float	12297.5
[19]	float	5245.0
[20]	float	2059.0

Figura 96. Vector de características ingresado

## 6.5 Sistema en la computadora

En esta sección se realiza la prueba de cada uno de los subsistemas desarrollados en la computadora con el objetivo de verificar su adecuado funcionamiento individual y en conjunto, posibilitando así una correcta integración del mismo con el sistema embebido para el desarrollo del presente proyecto.

### 6.5.1 Comunicación RF

La verificación del funcionamiento de este subsistema se realiza a través de la extracción de 960 s de características extraídas por el detector de emociones, de forma que se reciben 15 conjuntos de características equivalentes a 15 ventanas de 64 s, junto con su respectiva recepción y escritura en el archivo “features.csv” de acuerdo con la estructura descrita en la Tabla 39.

De la comparación entre el último conjunto de características extraídas del DSP y el último resultado escrito al archivo “features.csv” en la Figura 99, se corrobora la correcta recepción de las características sobre el archivo en la computadora con errores por debajo del 0.02%. Este error se debe nuevamente a la conversión que sufren los datos antes de su transmisión a través del protocolo BLE tal como se mencionó en la sección 6.1, a diferencia de que las características en el mapa de registros de la Tabla 13 manejan una aritmética de punto fijo Q16.

De esta forma, la escritura de características fisiológicas a un archivo externo permite al sistema almacenar los datos en la computadora a manera de una base de datos, lo cual evita el consumo de largos espacios de memoria en el procesador digital de señales.

	(pNN50=0.6446
pNN50	0.64467001
NN50	127.0
PPImed	0.95059979
SDNN	0.146515722
RMSSD	0.142968386
SD1_SD2	1.78454876
PRV_LF_HF	0.892418981
PRV_VLF	-3.9884727
PRV_LF	-18.3161011
PRV_HF	-20.5241051
SD1	0.180860817
SD2	0.101348206
NumSCR	0.0
AmpSCR	0.0
mTL	0.0
sdAmp	0.0
ctl25	0.0
ctl50	0.0
ctl75	0.0

	A	B	C	D	E	F	G	H	I	PRV_LF	PRV_VLF
1	pNN50	NN50	PPImed	SDNN	RMSSD	SD1/SD2	LF/HF	VLF			
2	0.4903717	102	1.18222046	0.13156128	0.10214233	2.37205505	0.909729	-9.6305695	-22.928406	-25.	
3	0.5148468	104	1.20542908	0.12774658	0.12878418	1.43717957	0.89941406	-9.540741	-22.898056	-2!	
4	0.48570251	102	1.14424133	0.2144165	0.20245361	1.38160706	0.75022888	-7.9790192	-17.386963	-2!	
5	0.42451477	90	1.10118103	0.24215698	0.24261475	1.18136597	0.73956299	-7.8166504	-16.886093	-22.	
6	0.3273468	73	1.06922913	0.23384094	0.23757935	1.09736633	0.74919128	-8.44487	-17.39679	-23.	
7	0.28694153	66	1.03382874	0.25585938	0.26371765	1.06959534	0.74742126	-9.261322	-16.929581	-22.	
8	0.22648621	53	1.02105713	0.24412537	0.24435425	1.1708374	0.74555969	-9.2300568	-17.047379	-22.	
9	0.2510376	60	1.00912476	0.21582031	0.21154785	1.20994568	0.74557495	-9.2300873	-17.04747	-22.	
10	0.2666626	64	0.99739075	0.2147522	0.21377563	1.19995117	0.74095154	-8.9890442	-16.994843	-22.	
11	0.2696991	65	1.0022583	0.20930481	0.19953918	1.33309937	0.7361908	-9.0582733	-16.828491	-22.	
12	0.28083801	66	1.01750183	0.22784424	0.21772766	1.32255554	0.73847961	-9.0155029	-16.886047	-2!	
13	0.26086426	60	1.03370667	0.25804138	0.25193787	1.27848816	0.74099731	-9.7302094	-16.972809	-22.	
14	0.27391052	63	1.02989197	0.2663269	0.26359558	1.24235535	0.75106812	-9.0162201	-17.110428	-22.	
15	0.3125	70	1.0433197	0.25782776	0.25965881	1.18426514	0.74227905	-11.167007	-16.889084	-22.	
16	0.64466858	127	0.95059204	0.14651489	0.14295959	1.7845459	0.89241028	-3.9884644	-18.316101	-20.	

Figura 99. (Izquierda) Últimas características extraídas de las señales fisiológicas PPG y SCR en el MCU. (Derecha) Base de datos de algunas de las características fisiológicas en el archivo features.csv.

### 6.5.2 Selección secuencial hacia delante flotante

Para verificar el buen funcionamiento del subsistema SFFS en las modalidades especificadas en la sección 4.4.7.3, se opta por realizar la prueba de éste utilizando uno de los conjuntos de datos que provee la biblioteca scikit-learn llamado *Breast cancer wisconsin (diagnostic) dataset* en cuyo interior hay 569 muestras, donde cada una de ellas está conformada por vectores con 30 características numéricas y sus respectivas clases. El conjunto es el resultado de un estudio para predecir el cáncer de mama por medio de características calculadas a partir de una imagen digitalizada, la cual se obtiene por la aspiración con aguja FNA de una masa mamaria. El problema plantea una clasificación binaria, siendo las clases contempladas aquellas que indiquen si la masa es maligna, con un total de 212 muestras, o benigna, con 357 muestras. Más información sobre el conjunto de datos puede ser encontrada en [81].

A continuación, se realiza una prueba de la selección de características efectuada por las funciones **SFFS\_wrapper()** y **SFFS\_wrapper\_LOOCV()** especificadas en la sección 5.8.3. Para ello, se lleva a cabo un entrenamiento y prueba del algoritmo LDA por medio de las funciones **LDA.train()** y **LDA.classification()** explicadas en la sección 5.8.5. Dicho procedimiento se realiza antes y después de la ejecución de la SFFS con el objetivo de visualizar su impacto en la exactitud de las clasificaciones después de la reducción de los vectores.

En el caso de la función **SFFS\_wrapper()** los datos se dividen en tres partes para efectuar las tareas anteriormente mencionadas, siendo el 60% de las muestras para entrenamiento, 20% para validación y 20% para prueba. Los primeros dos conjuntos son utilizados para ejecutar la SFFS y el entrenamiento del LDA, siendo el tercer conjunto únicamente para la prueba de este último. El motivo de dicha decisión yace en buscar una medida de rendimiento generalizada del LDA pues, tal como se explica en la sección 3.3.5, si se prueba el algoritmo con alguno de los conjuntos que emplea la SFFS para seleccionar las características, el resultado será sesgado pues el clasificador indirectamente estará ajustado a éstos ya que el selector lo utiliza de forma interna para realizar valoraciones.

En **SFFS\_wrapper\_LOOCV()** los datos se dividen únicamente en dos partes, siendo el 80% de muestras para entrenamiento y el 20% restante para prueba. Esto es porque dicha función no necesita de un conjunto de validación para evaluar el LDA internamente ya que emplea el mismo conjunto de entrenamiento a través de un proceso iterativo.

```
In [1]: runfile('C:/Users/Samuel/Desktop/github/ML/SFS/test.py', wdir='C:/Users/Samuel/Desktop/github/ML/SFS')

Dimensión actual de los vectores de características: 30
Exactitud con conjunto de prueba: 93.85964912280701 %
Tiempo de ejecución LDA: 45.769691467285156 ms

Tiempo de ejecución SFFS con validación simple: 10747.17378616333 ms
Nueva dimensión de los vectores de características: 27
Exactitud con conjunto de prueba: 91.22807017543859 %
Tiempo de ejecución LDA: 9.002685546875 ms

Tiempo de ejecución SFFS con validación cruzada dejando uno fuera:
1608558.8743686676 ms
Nueva dimensión de los vectores de características: 13
Exactitud con conjunto de prueba: 93.85964912280701 %
Tiempo de ejecución LDA: 8.592367172241211 ms
```

Figura 100. Prueba del algoritmo SFFS utilizando validación simple y validación cruzada dejando uno fuera

Como puede observarse en la Figura 100, la función **SFFS\_wrapper** ( ) realiza una reducción dimensional de 30 a 19 características aumentando ligeramente la exactitud, siendo de 95.61% antes de la selección y 97.37% posterior a ella. Por otro lado, la función **SFFS\_wrapper\_LOOCV** ( ) ejecuta una reducción de 30 a 11 características obteniendo una exactitud cercana, siendo de 96.49%. Si bien este último algoritmo tuvo un desempeño mayor que su contraparte en cuanto al número de características desechadas, es necesario resaltar las diferencias en tiempo de ejecución de ambos, siendo de 10.25 s para el algoritmo que emplea la validación simple y 1441.37 s para aquel que utiliza la validación cruzada dejando uno fuera. Con estos resultados queda probada la eficacia de los algoritmos pues, a pesar de las diferencias, ambos reducen los vectores de características del conjunto original manteniendo relativamente la misma exactitud en el clasificador.

### 6.5.3 Discriminante lineal de Fisher

Con el objetivo de verificar el funcionamiento del subsistema FLD especificado en la sección 5.8.4, se opta por realizar la prueba de éste utilizando el conjunto de datos descrito la sección 6.5.2. Para ello el total de muestras se divide en dos partes, siendo el 80% para entrenamiento y 20% para prueba.

En primer lugar, se utiliza la función **FLD.train** ( ) para obtener la matriz de reducción dimensional utilizando el conjunto de entrenamiento. Dado que los datos están compuestos de vectores pertenecientes a dos clases, la reducción debe ser a una dimensión por lo motivos expuestos en la sección 3.4.3. Posteriormente, se realiza la multiplicación de la matriz de reducción con los vectores del conjunto de prueba a fin de adquirir un espacio de características reducido. Finalmente, con el nuevo espacio se hace una evaluación de la efectividad del algoritmo LDA por medio de las funciones **LDA.train** ( ) y **LDA.classification** ( ) explicadas en la sección 5.8.5. Este último procedimiento se realiza antes y después de la ejecución del FLD con el objetivo de visualizar su impacto en la exactitud de las clasificaciones después de la reducción de los vectores.

```
In [1]: runfile('C:/Users/Samuel/Desktop/github/ML/FLD/test.py', wdir='C:/Users/Samuel/Desktop/github/ML/FLD')

Dimensión actual de los vectores de características: 30
Exactitud con conjunto de prueba: 93.85964912280701 %
Tiempo de ejecución LDA: 64.01634216308594 ms

Tiempo de ejecución de FLD: 15.003681182861328 ms
Nueva dimensión de los vectores de características: 1
Exactitud con conjunto de prueba: 93.85964912280701 %
Tiempo de ejecución LDA: 15.000581741333008 ms
```

Figura 101. Resultados de la reducción dimensional por medio de FLD

Tal como se observa en la Figura 101, el algoritmo FLD reduce los vectores de características de dimensión 30 a 1 manteniendo la misma exactitud en las clasificaciones del LDA, siendo de 95.61% antes y después de la implementación del reductor, demostrando así la efectividad del algoritmo para comprimir el espacio de características sin alterar la información que contiene. Una forma sencilla de visualizar el efecto del FLD es graficando el espacio de características original y el espacio resultante, lo cual se plasma en la Figura 102, donde se muestran las primeras dos dimensiones del espacio original y, posteriormente, la reducción a una dimensión efectuada.

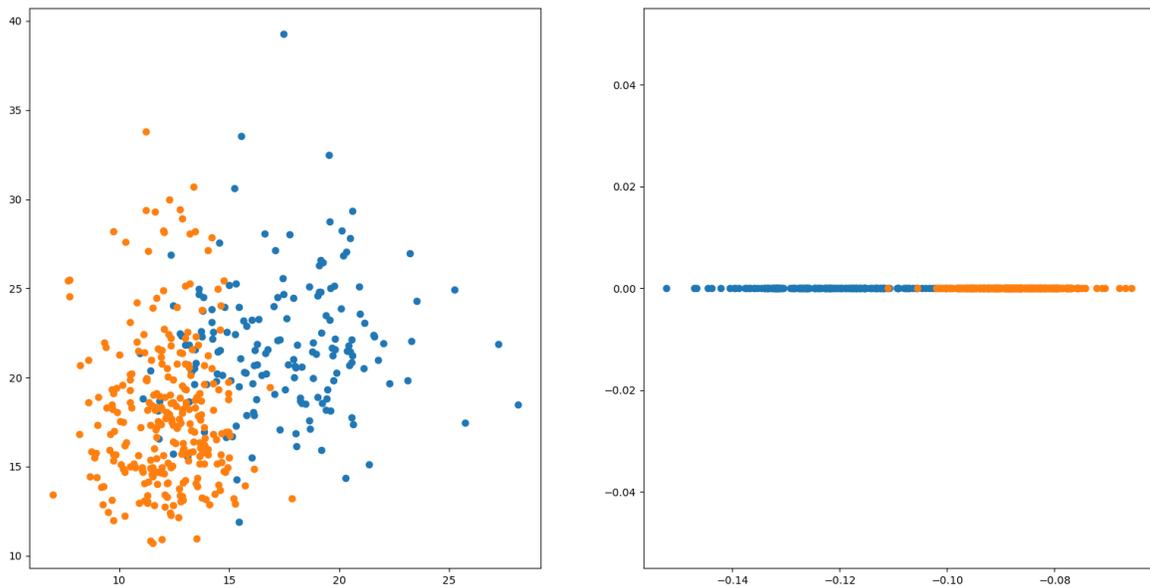


Figura 102. (Izquierda) Primeras dos dimensiones del espacio de características original / (Derecha) Espacio de características reducido por el FLD

#### 6.5.4 Algoritmo híbrido SFFS-FLD

Una vez mostrado el funcionamiento de los algoritmos SFFS y FLD por separado, es necesario mostrar su efectividad en conjunto pues de esta forma está implementado dentro del esquema EMDC. Para ello, se utiliza la misma metodología de la sección 6.5.2, sin embargo, en esta ocasión la salida de los algoritmos SFFS (en ambas modalidades) sirven como entrada de un FLD para generar una matriz de reducción que permita comprimir el espacio de características seleccionadas.

```
In [1]: runfile('C:/Users/Samuel/Desktop/github/ML/SFFS_FLD/test.py',
wdir='C:/Users/Samuel/Desktop/github/ML/SFFS_FLD')

Dimensión actual de los vectores de características: 30

Exactitud con conjunto de prueba: 95.1048951048951 %

Tiempo de ejecución LDA: 937.2317790985107 ms

Tiempo de ejecución SFFS con validación simple: 11419.883966445923 ms

Nueva dimensión de los vectores de características (SFFS): 27

Tiempo de ejecución de FLD: 13.014793395996094 ms

Nueva dimensión de los vectores de características (SFFS-FLD): 1

Exactitud con conjunto de prueba: 96.5034965034965 %

Tiempo de ejecución LDA: 7.985115051269531 ms

Tiempo de ejecución SFFS con validación cruzada dejando uno fuera:
1948585.0608348846 ms

Nueva dimensión de los vectores de características (SFFS): 13

Tiempo de ejecución de FLD: 13.876676559448242 ms

Nueva dimensión de los vectores de características (SFFS-FLD): 1

Exactitud con conjunto de prueba: 96.5034965034965 %

Tiempo de ejecución LDA: 6.997585296630859 ms
```

Figura 103. Ejecución de algoritmo híbrido SFFS-FLD

Como se muestra en la Figura 103, los algoritmos SFFS y FLD funcionan de forma óptima en conjunto, pues mientras SFFS deshecha aquellas características con menor impacto en el clasificador, maximizando su desempeño, FLD se encarga de comprimir el nuevo conjunto de características generado, obteniendo así las ventajas de ambos algoritmos de forma simultánea, pasando de un conjunto de datos con vectores de 30 elementos y una exactitud del 95.61%, a uno con vectores unidimensionales que arrojan una exactitud del 97.37% en el caso de la SFFS con validación simple y 96.49% utilizando la validación cruzada dejando uno fuera. Si bien parece que en esta ocasión el uso de la validación simple es preferible tanto en tiempo como en rendimiento, es necesario mencionar que los resultados pueden variar al implementarse sobre el detector de emociones o empleándolo sobre conjuntos de datos más pequeños.

### 6.5.5 Análisis discriminante lineal

Para verificar el buen funcionamiento del subsistema LDA especificado en la sección 5.8.5, se opta por realizar el entrenamiento y prueba de dicho algoritmo empleando uno de los conjuntos de datos que provee la biblioteca scikit-learn llamado *Iris plants dataset*, el cual contiene 150 muestras con 4 características numéricas por cada una. Asimismo, en su interior existen 3 clases distintas repartidas de manera uniforme, siendo cada clase un tipo de planta de iris. Más información sobre el conjunto de datos puede ser encontrada en [81]. La razón por la que se elige utilizar un conjunto distinto al empleado en la sección 6.5.2 es por meros fines visuales, ya que el número de características es relativamente menor, facilitando la visualización de los parámetros obtenidos durante el entrenamiento. El conjunto de datos se divide en dos partes para efectuar la validación del algoritmo, siendo el 80% de las muestras para entrenamiento y 20% para prueba.

Tal como se plasma en la Figura 104, al entrenar el algoritmo LDA se obtienen los parámetros indicados en la sección 3.5.1, los cuales conforman las funciones lineales que predicen las clases durante la etapa de prueba, presentando en ella una exactitud del 96.67%. Cabe resaltar que el resultado obtenido verifica que el algoritmo no presenta un sobreajuste a los datos de entrenamiento (véase la sección 3.3.4), confirmando el buen funcionamiento del mismo.

```
In [1]: runfile('C:/Users/Samuel/Desktop/github/ML/LDA/test.py',
          wdir='C:/Users/Samuel/Desktop/github/ML/LDA')

S:
[[0.26545644 0.0961443  0.17338308 0.0374443 ]
 [0.0961443  0.12079918 0.06020962 0.03898039]
 [0.17338308 0.06020962 0.18975453 0.044517  ]
 [0.0374443  0.03898039 0.044517  0.04513082]]

m_k:
[[5.01621622 3.43243243 1.46756757 0.25945946]
 [5.95       2.78409091 4.24090909 1.32272727]
 [6.58717949 2.95897436 5.57948718 2.02820513]]

pi_k:
[[0.30833333]
 [0.36666667]
 [0.325      ]]

Exactitud con conjunto de prueba: 96.6666666666667 %

Tiempo de ejecución LDA: 4.00090217590332 ms
```

Figura 104. Entrenamiento y prueba de algoritmo LDA

### 6.5.6 Esquema EMDC

Para realizar la comprobación del algoritmo desarrollado en la sección 5.8.6 correspondiente al esquema EMDC, se introducen una serie de vectores de características generados dentro del sistema embebido, empleando para ello datos del sensor biométrico y la base de datos especificada en la sección 6.2. El objetivo es realizar un entrenamiento en cada etapa del esquema y obtener sus parámetros correspondientes. Asimismo, cabe recalcar que únicamente se busca comprobar paso a paso el correcto desarrollo del algoritmo pues, como se especifica en la sección 4.2.4, para evaluar el desempeño del sistema es necesario adquirir muestras adjudicadas a emociones empleando una metodología en específico. Para esta prueba las clases fueron asignadas de forma aleatoria con el único

objetivo de hacer funcionar al subsistema. A continuación, se muestran los resultados del esquema después de su ejecución en las modalidades de entrenamiento rápido y completo.

```

Execution time of SFFS_1 with 11 dimensions: 2369.600772857666 ms
Execution time of FLD_1 with 1 dimensions: 8.016109466552734 ms
*****LDA_1*****
Training time: 2.0132064819335938 ms
Accuracy with LOOCV: 77.0 %

Execution time of SFFS_2 with 4 dimensions: 648.8504409790039 ms
Execution time of FLD_2 with 1 dimensions: 3.813505172729492 ms
*****LDA_2*****
Training time: 4.001379013061523 ms
Accuracy with LOOCV: 61.038961038961034 %

Execution time of SFFS_3 with 5 dimensions: 345.7341194152832 ms
Execution time of FLD_3 with 1 dimensions: 0.0 ms
*****LDA_3*****
Training time: 8.01229476928711 ms
Accuracy with LOOCV: 63.04347826086957 %

Execution time of SFFS_4 with 7 dimensions: 630.465030670166 ms
Execution time of FLD_4 with 1 dimensions: 8.030891418457031 ms
*****LDA_4*****
Training time: 0.0 ms
Accuracy with LOOCV: 66.18705035971223 %

```

Figura 106. Resultados del esquema EMDC empleando el entrenamiento rápido

```

Execution time of SFFS_1 with 16 dimensions: 667525.4034996033 ms
Execution time of FLD_1 with 1 dimensions: 6.001472473144531 ms
*****LDA_1*****
Training time: 5.000829696655273 ms
Accuracy with LOOCV: 77.0 %

Execution time of SFFS_2 with 5 dimensions: 116646.38710021973 ms
Execution time of FLD_2 with 1 dimensions: 3.997325897216797 ms
*****LDA_2*****
Training time: 2.998828887939453 ms
Accuracy with LOOCV: 67.09956709956711 %

Execution time of SFFS_3 with 5 dimensions: 20758.387327194214 ms
Execution time of FLD_3 with 1 dimensions: 2.000093460083008 ms
*****LDA_3*****
Training time: 1.0001659393310547 ms
Accuracy with LOOCV: 65.21739130434783 %

Execution time of SFFS_4 with 7 dimensions: 62157.28235244751 ms
Execution time of FLD_4 with 1 dimensions: 3.0012130737304688 ms
*****LDA_4*****
Training time: 2.155303955078125 ms
Accuracy with LOOCV: 66.18705035971223 %

```

Figura 105. Resultados del esquema EMDC empleando el entrenamiento completo

Al observar la Figura 106 y Figura 105 es evidente que las exactitudes mostradas señalan un pobre rendimiento del subsistema, sin embargo, esto es debido a que las clases fueron asignadas de forma aleatoria, por lo que no se deben considerar a dichos resultados como una medida de su desempeño. Por otro lado, se demuestra que el algoritmo desarrollado entrena y evalúa cada etapa con normalidad, garantizando el correcto funcionamiento del esquema EMDC.

## 7 Conclusiones y trabajo a futuro

Se ha identificado que numerosos trabajos en el área de la computación afectiva han sido desarrollados con base en computadoras con procesadores de alto nivel, sin embargo, esto conlleva una complicación para la elaboración de dispositivos portátiles. Por ende, en esta tesis se propuso el desarrollo de un detector de emociones utilizando señales fisiológicas procesadas en un procesador digital de señales, siguiendo las características denotadas en la Tabla 46 y que, a su vez, puede compararse con los trabajos en el Estado del Arte dentro de la Tabla 2. Éste último entabla una comunicación Bluetooth con un medio externo el cual realiza el entrenamiento de la etapa de Aprendizaje Automático encargada de la clasificación de las emociones. Sin embargo, a pesar de no poderse llevar a cabo un estudio completo que permitiera comprobar el desempeño en conjunto del detector de emociones, fue posible, mediante señales adquiridas por el sistema o empleando bases de datos, verificar el funcionamiento adecuado de los diversos subsistemas que lo componen al analizar el resultado de sus salidas.

Tabla 46. Características particulares del detector de emociones de esta tesis.

Señal	Características extraídas	Clasificación	Descripción
PRV, SCR	<b>PRV:</b> HF, LF, VLF, LF/HF, SD2, SD1, SD1/SD2, RMSSD, SDNN, PPImed, NN50, pNN50 <b>SCR:</b> cti90, cti75, cti50, cti25, sdAmpl, mTL, AmpSCR, NumSCR, HF	<b>Clasificador:</b> LDA <b>Selector:</b> SFFS <b>Reductor:</b> FLD <b>Validación:</b> Validación cruzada dejando uno fuera	5 regiones (EA-VN, EA-VP, EN-VN, EN-VP y Neutro) del MD-VE.

Para la comunicación entre el procesador digital de señales y la computadora se identificó una correcta conexión y transmisión de los datos, a expensas de las pérdidas en precisión de algunas variables que sufren de la conversión en su tipo de aritmética, al presentarse errores por debajo del 3% con respecto a aquellos obtenidos originalmente en punto flotante de doble precisión. Se logra la adquisición de la señal obtenida mediante la PPG con una SNR de 38.49 dB, no obstante, la señal EDA no obtuvo el comportamiento esperado a raíz de un mal acondicionamiento en la electrónica analógica, por lo que la evaluación del desempeño de los subsistemas correspondientes a esta señal es realizada a través de bases de datos. El acondicionamiento de las señales fisiológicas EDA y PPG fue adecuadamente realizado, teniendo un SNR de 52.24 dB y 54.35 dB, respectivamente. El procesamiento digital de las señales y la extracción de sus características en tiempo, frecuencia y no lineales a través del DSP no sobrepasaron el 0.02% de error con respecto a aquellas estimadas en MATLAB. Sin embargo, a partir de las características en el dominio de la frecuencia, se identificó la limitante de requerir una ventana de por lo menos 256 s para la señal obtenida mediante la PPG, lo cual se contrapone con el análisis de la ventana de 64 s para la señal EDA. Para ello, se propuso una metodología que permite resolver esta problemática mediante el desplazamiento de los datos PRV para un análisis de ambas señales cada 64 s.

En cuanto al subsistema de clasificación de características desarrollado dentro del DSP, se comprobó que éste obtiene resultados coherentes respecto a las simulaciones realizadas en MATLAB, constatando así un correcto funcionamiento para realizar su tarea en cuestión. Asimismo, empleando los algoritmos que tienen como objetivo efectuar el entrenamiento de dicha etapa dentro del sistema en la computadora, se validó un correcto funcionamiento individual al obtener exactitudes superiores al 90% utilizando bases de datos especializadas, verificando así la correcta selección de características por parte del algoritmo SFFS, reducción dimensional con el FLD, clasificación empleando el LDA y, finalmente, su funcionamiento en conjunto dentro del esquema EMDC.

A pesar de no haberse empleado características adjudicadas a emociones para comprobar el rendimiento de la etapa de Aprendizaje Automático como tal, se considera que el sistema desarrollado en este proyecto es una buena propuesta para la detección de emociones debido al rendimiento demostrado del esquema EMDC en el artículo [8], que si bien emplea una metodología similar, éste clasifica las muestras obteniendo las distancias euclidianas respecto a los centroides de cada clase, a diferencia del presente proyecto que emplea LDA, el cual es un algoritmo más robusto que puede

mejorar el rendimiento del sistema. Asimismo, este trabajo trae consigo un valor agregado debido a su implementación en un dispositivo portátil, en contraste con el resto de los sistemas del Estado del Arte, lo cual fue posible dada la operabilidad en conjunto con el dispositivo externo de mayor capacidad (computadora) permitiendo la reducción del volumen de trabajo por procesar dentro del sistema embebido (DSP). Esto queda evidenciado al observar el tiempo de ejecución del entrenamiento de los algoritmos en el sistema de la computadora, siendo de aproximadamente 4.02 s para el entrenamiento rápido y 867.11 s para el entrenamiento completo, a diferencia de la simple implementación del esquema EMDC en el procesador digital de señales con un total de 2.3025 ms. Además, esto permite al dispositivo externo fungir como base de datos de todas las emociones y características extraídas del sistema embebido.

En conclusión, los resultados de la presente tesis comprueban la posibilidad de integrar un sistema embebido que incorpore las etapas fundamentales de un detector de emociones, cuya implementación es adecuada tanto para aplicaciones de duración superior a los 64 s como para dispositivos vestibles de bajo costo. De esta forma se da lugar al desarrollo de productos que permiten interactuar con las emociones, mejorando así, la calidad de vida del ser humano.

Cabe mencionar que fuera de las observaciones realizadas a lo largo de la presente tesis, tras el desarrollo del detector de emociones se identificaron determinados rubros cuya modificación podría mejorar su funcionamiento. Entre ellas podemos mencionar las siguientes:

- La omisión de las características en frecuencia podría permitir ampliar el uso del dispositivo a aplicaciones de duración inferior a los 64 s, sin embargo, es posible que disminuya el rendimiento del clasificador, lo cual requiere evaluarse con antelación.
- Debido a la naturaleza del algoritmo de generación de la PRV en su diferenciación finita hacia atrás, puede resultar innecesario un valor de SNR tan elevado para la señal obtenida mediante la PPG, por lo que sería posible reducir el grado de los filtros FIR u omitir la etapa de acondicionamiento digital de ésta.
- El mecanismo de desplazamiento de ventanas aplicado sobre los datos de la PRV podría aplicarse también sobre los datos de la señal SCR, lo cual permitiría ampliar el uso del dispositivo a aplicaciones de duración inferior a los 64 s, aún si se mantienen las estimaciones en frecuencia. No obstante, la longitud del desplazamiento debe seleccionarse con cuidado a fin de permitir la suficiente varianza de las ventanas de datos para así garantizar resultados útiles.
- Dado que el uso de los electrodos para la señal EDA compromete la ergonomía del dispositivo, en su lugar podría verse viable el manejo de la señal de respiración, ya que existen artículos que parten de la señal PPG para su adquisición debido a que un fenómeno de modulación entre ambas, no obstante, esto conllevaría un mayor grado de procesamiento digital de la señal.
- En caso de realizar el punto anterior, existe la posibilidad de remplazar el sensor biométrico por otro enfocado primordialmente a la señal obtenida mediante PPG a fin de reducir el costo del sistema embebido.
- Posibilidad de omitir el procedimiento de interpolación por trazadores cúbicos de la señal PRV, para remplazarse por una estimación de la PSD a partir del método de Lomb-Scargle que contempla el análisis de muestras no constantes en tiempo.
- Para permitir el funcionamiento de esta clase sobre aplicaciones con duración inferior a los 64 s y considerando las estimaciones en frecuencia que pueden resultar bastante útiles para la clasificación de emociones, puede buscarse otro procesador digital de señales que cuente con un segundo núcleo para alcanzar el paralelismo de los procesos realizados, o bien un esquema en concurrencia a través de un sistema operativo en tiempo real.

- Posibilidad de omitir el uso de la TMU a cambio de un mayor tiempo de procesamiento para la estimación de las funciones trigonométricas seno y coseno empleadas dentro de los análisis en frecuencia. Esto da lugar a que pueda incrementarse el número de dispositivos en los que puede replicarse el sistema embebido.
- Existe la posibilidad de reducir el tamaño y costo del sistema embebido si se selecciona un sistema en un chip que integre el protocolo BLE para la comunicación inalámbrica con el dispositivo externo. O bien, se integre todo el sistema dentro de una sola PCB, dejando de lado la utilización de la tarjeta de desarrollo y en su lugar se utilice meramente el procesador digital de señales que la tarjeta utiliza.
- Posibilidad de ahorrar espacio en memoria omitiendo características que involucran cantidades porcentuales, tal como los cuantiles y el tiempo de levantamiento. Esto es debido a que ambos requieren de un valor máximo como referencia, por lo que es necesario el almacenamiento de muestras para su uso posterior.
- A lo largo del Estado del Arte no fue común el uso del algoritmo hermano de la SFS llamado Selección secuencial hacia atrás (SBS, por sus siglas en inglés) así como su variante flotante. Éste plantea iniciar con el conjunto completo de características e ir desechando las mismas a fin de buscar el mejor rendimiento del clasificador, siendo básicamente el proceso inverso de SFS. Es recomendable evaluar sí este algoritmo puede proporcionar un mejor rendimiento que aquel utilizado en este proyecto, por lo que para realizar su diseño se recomienda consultar [59], tomando en consideración si es posible mejorar su desempeño tal como se hizo con SFFS en la sección 4.4.7.3.
- En caso de que el sistema muestre un pobre rendimiento en la clasificación de emociones, se sugiere cambiar el algoritmo LDA por su versión cuadrática QDA, cuya única diferencia consiste en considerar a las matrices de covarianza para cada clase diferentes entre sí, lo cual se describe con mayor detalle en la sección 3.5.1.
- Si bien el dispositivo no fue probado para una aplicación en específico, en caso de utilizarse en futuros trabajos se recomienda tomar en consideración el uso de los estimuladores con mejor rendimiento indicados en la sección 2.

## 8 Referencias

- [1] P. Rosalind, *Affective Computing*, MIT Press, 2000.
- [2] C. Walter, «The James-Lange Theory of Emotions: A Critical Examination and an Alternative Theory,» *The American Journal of Psychology*, vol. 39, n° 1/4, p. 106–124, 1927.
- [3] R. Picard, E. Vyzas y J. Healey, «Toward machine emotional intelligence: analysis of affective physiological state,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, n° 10, pp. 1175-1191, 2001.
- [4] J. Wagner, J. Kim y E. André, «From Physiological Signals to Emotions: Implementing and Comparing Selected Methods for Feature Extraction and Classification,» de *International Conference on Multimedia and Expo*, Amsterdam, 2005.
- [5] C. Lee, S. Yoo, Y. Park, N. Kim, K. Jeong y B. Lee, «Using Neural Network to Recognize Human Emotions from Heart Rate Variability and Skin Resistance,» de *Engineering in Medicine and Biology 27th Annual Conference*, Shanghai, 2005.
- [6] O. Villon y C. Lisetti, «A User-Modeling Approach to Build User's Psycho-Physiological Maps,» de *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, Hatfield, 2006.
- [7] O. Villon y C. Lisetti, «A User Model of Psycho-physiological Measure of Emotion,» de *International Conference on User Modeling, Adaptation, and Personalization*, Berlin, 2007.
- [8] J. Kim y E. André, «Emotion recognition based on physiological changes in music listening,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, n° 12, pp. 2067-2083, Diciembre 2008.
- [9] M. Solymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt y I. Patras, «DEAP: A Database for Emotion Analysis ;Using Physiological Signals,» *IEEE Transactions on Affective Computing*, vol. 3, n° 1, pp. 18-31, Enero-Marzo 2012.
- [10] F. Agrafioti, D. Hatzinakos y A. Anderson, «ECG Pattern Analysis for Emotion Detection,» *IEEE Transactions on Affective Computing*, vol. 3, n° 1, pp. 102-115, Enero-Marzo 2012.
- [11] G. Valenza, L. Citi, A. Lanata, E. Pasquale y R. Barbieri, «A nonlinear heartbeat dynamics model approach for personalized emotion recognition,» de *5th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Osaka, 2013.
- [12] E.-H. Jang, B.-J. Park, S.-H. Kim, M.-A. Chung, M.-S. Park y J.-H. Sohn, «Emotion Classification based on Bio-Signals emotion recognition using machine learning algorithms,» de *2014 International Conference on Information Science, Electronics and Electrical Engineering*, Daejeon, 2014.
- [13] H. W. Guo, Y. S. Huang, J. C. Chien y J. S. Shieh, «Short-term analysis of heart rate variability for emotion recognition via wearable ECG device,» de *2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Okinawa, 2015.
- [14] M. Nardelli, G. Valenza, A. Greco, A. Lanata y E. P. Scilingo, «Recognizing Emotions Induced by Affective Sounds through Heart Rate Variability,» *IEEE Transactions on Affective Computing*, vol. 6, n° 4, p. 10, 2015.
- [15] M. Zhao, F. Adib y D. Katabi, «Emotion Recognition Using Wireless Signals,» *Commun. ACM*, vol. 61, n° 9, pp. 91-100, 3-7 Octubre 2016.

- [16] M. Matsubara, O. Augereau, C. L. Sanches y K. Kise, «Emotional arousal estimation while reading comics based,» de *MANPU '16*, Cancún, 2016.
- [17] B. Henia y Z. Lachiri, «Emotion sensing from physiological signals using three defined areas in arousal-valence model,» de *2017 International Conference on Control, Automation and Diagnosis (ICCAD)*, Hammamet, 2017.
- [18] B.-J. Park, C. Yoon, E.-H. Jang y D.-H. Kim, «Physiological signals and recognition of negative emotions,» de *2017 International Conference of Information and Communication Technology Convergence (ICTC)*, Daejeon, 2017.
- [19] P. Schmidt, A. Reiss, R. Dürichen, C. Marberger y K. V. Laerhoven, «Introducing WESAD, a Multimodal Dataset for Wearable,» de *International Conference on Multimodal Interaction (ICMI '18)*, Boulder, 2018.
- [20] R. Harper y J. Southern, «End-To-End Prediction of Emotion From Heartbeat Data Collected by a Consumer Fitness Tracker,» de *International Conference on Affective Computing and Intelligent Interaction (ACII)*, Cambridge, 2019.
- [21] R. W. Levenson, «The autonomic nervous system and emotion,» *Emotion review*, vol. 6, n° 2, pp. 100-112, 2014.
- [22] P. Bota, C. Wang, A. Fred y H. Placido, «A Review, Current Challenges, and Future Possibilities on Emotion Recognition Using Machine Learning and Physiological Signals,» *IEEE Access*, vol. 7, pp. 140990-141020, 2019.
- [23] J. Pan y W. J. Tompkins, «A real-time QRS detection algorithm,» *IEEE Trans. Biomed. Eng.*, vol. 32, n° 3, pp. 230-236, 1985.
- [24] P. Hamilton, «Open source ECG analysis,» de *Computers in Cardiology*, Memphis, 2002.
- [25] I. I. Christov, «Real time electrocardiogram QRS detection using combined adaptive threshold,» *Biomed. Eng. Online*, vol. 3, n° 1, p. 28, 2004.
- [26] W. A. H. Engelse y C. Zeelenberg, «A single scan algorithm for QRS-detection and feature extraction,» *Computers in Cardiology*, vol. 6, pp. 37-42, 1979.
- [27] W. Zong, T. Heldt, G. B. Moody y R. G. Mark, «An open-source algorithm to detect onset of arterial blood pressure pulses,» de *Computers in Cardiology, 2003*, Thessaloniki Chalkidiki, 2003.
- [28] F. Canento, A. Lourenço, H. Silva y A. L. N. Fred, «Review and comparison of real time electrocardiogram segmentation algorithms for biometric applications,» de *Proc. 6th Int. Conf. Health Inform.*, 2012.
- [29] L. Lindberg y P. Öberg, «Photoplethysmography,» *Medical and Biological Engineering and Computing*, vol. 29, pp. 48-54, 1991.
- [30] S. K. Langmore, G. Y. Lui, G. Naik, P. P. Breen, B. Jalaludin y G. D. Gargiulo, «A comparison of reflective photoplethysmography for detection of heart rate, blood oxygen saturation, and respiration rate at various anatomical locations,» *Sensors*, vol. 19, n° 8, 2019.
- [31] Y. Maeda, M. Sekine, T. Tamura, A. Moriya, T. Suzuki y K. Kameyama, «Comparison of reflected green light and infrared photoplethysmography,» de *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vancouver, 2008.
- [32] A. Choi y H. Shin, «Photoplethysmography sampling frequency: pilot assessment of how low can we go to analyze pulse rate variability,» *Institute of Physics and Engineering in Medicine*, vol. 38, n° 3, p. 586–600, 2017.

- [33] D. Fujita y A. Susuki, «Evaluation of the Possible Use of PPG Waveform Features Measured at Low Sampling Rate,» *IEEE Access*, vol. 7, pp. 58361-58367, 2019.
- [34] M. Aboy, J. McNames, T. Thong, D. Tsunami, M. S. Ellenby y B. Goldstein, «An automatic beat detection algorithm for pressure signals,» *IEEE Transactions on Biomedical Engineering*, vol. 52, n° 10, pp. 1662-1670, 2005.
- [35] B. N. Li, M. C. Dong y M. I. Vai, «On an automatic delineator for arterial blood pressure waveforms,» *Biomedical Signal Processing and Control*, vol. 5, n° 1, pp. 76-81, 2010.
- [36] S. Vadrevu y M. S. Manikandan, «Real-Time Quality-Aware PPG Waveform Delineation and Parameter Extraction for Effective Unsupervised and IoT Health Monitoring Systems,» *IEEE Sensors Journal*, vol. 19, n° 17, pp. 7613-7623, 2019.
- [37] A. Verma, S. Cabrera, A. Mayorga y H. Nazeran, «A Robust Algorithm for Derivation of Heart Rate Variability Spectra from ECG and PPG Signals,» de *2013 29th Southern Biomedical Engineering Conference*, 2013.
- [38] B. Ricardo Ferro, A. Ramírez Aguilera y R. Fernández De La Vara Prieto, «Automated detection of the onset and systolic peak in the pulse wave using Hilbert transform,» *Biomedical Signal Processing and Control*, vol. 20, pp. 78-84, 2015.
- [39] E. J. A. Prada, «The mountaineer's method for peak detection in photoplethysmographic signals,» *Revista Facultad de Ingeniería, Universidad de Antioquia*, n° 90, pp. 42-50, 2019.
- [40] G. Lu, F. Yang, J. A. Taylor y J. F. Stein, «A comparison of photoplethysmography and ECG recording to analyse heart rate variability in healthy subjects,» *Journal of Medical Engineering & Technology*, vol. 33, n° 8, pp. 634-641, 2009.
- [41] A. Verma, S. Cabrera, A. Mayorga y H. Nazeran, «A Robust Algorithm for Derivation of Heart Rate Variability Spectra from ECG and PPG Signals,» de *2013 29th Southern Biomedical Engineering Conference*, Miami, 2013.
- [42] N. Pinheiro, R. Couceiro, J. Henriques, J. Muehlsteff, I. Quintal, L. Gonçalves y P. Carvalho, «Can PPG be used for HRV analysis?,» de *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, 2016.
- [43] J. Cacioppo, L. G. Tassinari y G. G. Berntson, *The handbook of psychophysiology*, New York: Cambridge University Press, 2007.
- [44] J. A. Healey y R. Picard, «Detecting stress during real-world driving tasks using physiological sensors,» *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, n° 2, pp. 156-166, 2005.
- [45] A. Gautam, N. Simoes-Capela, G. Schiavone, A. Acharyya, W. de Raedt y C. Van Hoof, «A Data Driven Empirical Iterative Algorithm for GSR Signal Pre-Processing,» de *2018 26th European Signal Processing Conference (EUSIPCO)*, Roma, 2018.
- [46] L. Escobar, *Diseño de Filtros Digitales*, CDMX: Facultad de Ingeniería UNAM, 2006.
- [47] J. Cooley y J. Tukey, «An Algorithm for the Machine Calculation of Complex Fourier Series,» *Mathematics of Computation*, vol. 19, n° 90, pp. 297-301, 1965.
- [48] S. Kay y L. M. Marple, «Spectrum analysis—A modern perspective,» *Proceedings of the IEEE*, vol. 69, n° 11, pp. 1380-1419, 1981.
- [49] T. Söderstrom y P. Stoica, *System Identification*, USA: Cambridge University Press, 1989.
- [50] J. Proakis y D. Manolakis, «Digital Signal Processing. Principles, algorithms and applications,» USA, Prentice Hall International, 1996, p. 919.

- [51] O. Solomon, «PSD computations using Welch's method,» Albuquerque, NM, 1991.
- [52] R. Avram, G. Tison, K. Aschbacher, P. Kuhar, E. Vittinghoff, M. Butzner, R. Runge, N. Wu, M. Pletcher, G. Marcus y J. Olgin, «Real-world heart rate norms in the Health eHeart study,» *NPJ Digital Medicine*, vol. 2, n° 58, pp. 1-10, 2019.
- [53] J. Pereira-Rodríguez, L. Boada-Morales, T. Jaimes-Martin, J. Melo-Ascanio, D. Niño-Serrato y G. Rincón-González, «Predictive equations for maximum heart rate. Myth or reality,» *Revista Mexicana de Cardiología*, vol. 27, n° 4, pp. 156-165, 2016.
- [54] R. U. Acharya, P. K. Joseph, N. Kannathal, C. M. Lim y J. S. Suri, «Heart rate variability: a review,» *Medical and Biological Engineering and Computing*, vol. 44, n° 12, pp. 1031-1051, 2006.
- [55] B. CardioPoint, *Poincare graph: complete ecg record in one sight*, 2014.
- [56] C. Gerald y P. Wheatley, *Applied Numerical Analysis*, Estados Unidos: Pearson Education, 2004.
- [57] J. Hoffman, *Numerical Methods for Engineers and Scientists*, Estados Unidos: Marcel Dekker, 2001.
- [58] K. H. Kim, S. W. Bang y S. R. Kim, «Emotion recognition system using short-term monitoring of physiological signals,» *Med. Biol. Eng. Comput.*, vol. 42, n° 3, pp. 419-427, 2004.
- [59] S. Raschka, «MLXTEND,» 2014. [En línea]. Available: [http://rasbt.github.io/mlxtend/user\\_guide/feature\\_selection/SequentialFeatureSelector/](http://rasbt.github.io/mlxtend/user_guide/feature_selection/SequentialFeatureSelector/). [Último acceso: 9 Mayo 2021].
- [60] T. Silva, «Thalles' blog,» 03 Enero 2019. [En línea]. Available: <https://sthalles.github.io/fisher-linear-discriminant/>. [Último acceso: 25 Julio 2021].
- [61] Y. Xiaozhou, «towards data science,» 9 Mayo 2020. [En línea]. Available: <https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b>. [Último acceso: 3 Septiembre 2020].
- [62] T. Hastie, R. Tibshirani y J. Friedman, *The Elements of Statistical Learning*, New York: Springer, 2001.
- [63] J. Amat Rodrigo, «Ciencia de Datos, Estadística, Machine Learning y Programación,» Septiembre 2016. [En línea]. Available: [https://www.cienciadedatos.net/documentos/28\\_linear\\_discriminant\\_analysis\\_lda\\_y\\_quadric\\_discriminant\\_analysis\\_qda#An%C3%A1lisis\\_Discriminante\\_Cuadr%C3%A1tico](https://www.cienciadedatos.net/documentos/28_linear_discriminant_analysis_lda_y_quadric_discriminant_analysis_qda#An%C3%A1lisis_Discriminante_Cuadr%C3%A1tico). [Último acceso: 23 Abril 2021].
- [64] R. A. Calvo y S. D'Mello, «Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications,» *IEEE Transactions on Affective Computing*, vol. I, n° 1, pp. 18-37, 2010.
- [65] C. Darwin, *The expression of the Emotions in Man and Animals*, Nueva York: D. Appleton and Company, 1897.
- [66] M. Arnold, *Emotion and Personality Vol. 1 & 2*, Nueva York: Columbia University Press, 1960.
- [67] J. Averill, *A Constructivist View of Emotion*, Academic Press, 1980, pp. 305-339.
- [68] P. Salovey, «Introduction: Emotion and Social Processes,» de *Handbook of Affective Science*, Oxford University Press, 2003.

- [69] O. Villon y C. Lisetti, «Toward Recognizing Individual's Subjective Emotion from Physiological Signals in Practical Application,» de *Twentieth IEEE International Symposium on Computer-Based Medical Systems (CBMS'07)*, Maribor, 2007.
- [70] M. Bradley y P. Lang, «Measuring emotion: The self-assessment manikin and the semantic differential,» *Journal of Behavior Therapy and Experimental Psychiatry*, vol. 25, n° 1, pp. 49-59, 1994.
- [71] K. Ulrich y S. Eppinger, *Diseño y desarrollo de productos: enfoque multidisciplinario*, McGrawHill, 2013.
- [72] F. Elsamnah, A. Bilgaiyan, M. Affiq, C.-H. Shim, H. Ishidai y R. Hattori, «Comparative Design Study for Power Reduction in Organic Optoelectronic Pulse Meter Sensor,» *MDPI Biosensors*, vol. 9, n° 48, pp. 1-13, 2019.
- [73] N. Semiconductors, *I2C-bus specification and user manual*, 2014.
- [74] AMS, *AS702x Galvanic Skin Resistance*, 2019.
- [75] AMS, *AS7026GG Datasheet*, 2019.
- [76] A. Anusha, J. Jose, S. Preejith, J. Jayaraj y S. Mohanasankar, «Physiological signal based work stress detection using unobtrusive sensors,» *Biomedical Physics & Engineering Express*, vol. 4, n° 6, pp. 1-18, 2018.
- [77] D.-G. Jang, M. Hahn, J.-K. Jang, U. Farooq y S.-H. Park, «A comparison of interpolation techniques for RR interval fitting in AR spectrum estimation,» de *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Hsinchu, 2012.
- [78] G. Clifford, *Signal Processing Methods*, University of Oxford, 2002.
- [79] T. Brittain, «How to Send Data between PC and Arduino using Bluetooth LE,» 11 Julio 2020. [En línea]. Available: <https://ladvien.com/python-serial-terminal-with-arduino-and-bleak/>. [Último acceso: 2020 Noviembre 8].
- [80] National Institute of General Medical Sciences (NIGMS) & National Institute of Biomedical Imaging and Bioengineering (NIBIB), «PhysioBank ATM,» [En línea]. Available: <https://archive.physionet.org/cgi-bin/atm/ATM>. [Último acceso: 2021 04 09].
- [81] scikit-learn developers, «scikit learn,» 2007. [En línea]. Available: [https://scikit-learn.org/stable/datasets/toy\\_dataset.html](https://scikit-learn.org/stable/datasets/toy_dataset.html). [Último acceso: 08 Abril 2021].
- [82] J. Russell, «Core affect and the psychological construction of emotion,» *Psychological Review*, vol. 110, n° 1, p. 145–172, 2003.
- [84] S. Hanly, «Endaq Blogs,» [En línea]. Available: <https://blog.endaq.com/why-the-power-spectral-density-psd-is-the-gold-standard-of-vibration-analysis>. [Último acceso: 30 Junio 2020].
- [85] R. Kari, K. Ismo, E. Paulo y V. Vesa, «Frequency warped Burg's method for AR-modeling,» de *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*, NY, USA, 2003.
- [86] D. F. Weber, «SpiSOP,» 2013. [En línea]. Available: <https://www.spisop.org/documentation/>. [Último acceso: 01 07 2020].
- [87] A. Smola y S. V. N. Vishwanathan, *Introduction to Machine Learning*, Cambridge, United Kingdom: Cambridge University Press, 2008.
- [88] C. M. Bishop, *Pattern Recognition and Machine Learning*, Singapore: Springer, 2006.

- [89] J. Brownlee, «Machine Learning Mastery,» 27 Diciembre 2019. [En línea]. Available: <https://machinelearningmastery.com/how-to-develop-an-intuition-skewed-class-distributions/>. [Último acceso: 28 Julio 2020].
- [90] J. Brownlee, «Machine Learning Mastery,» 8 Abril 2020. [En línea]. Available: <https://aidevelopmenthub.com/4-types-of-classification-tasks-in-machine-learning/>. [Último acceso: 28 Julio 2020].
- [91] A. Budhiraja, «Medium,» 19 Julio 2017. [En línea]. Available: <https://medium.com/@amarbudhiraja/ml-101-linear-regression-tutorial-1e40e29f1934>. [Último acceso: 28 Julio 2020].
- [92] S. Priy, «GeeksforGeeks,» [En línea]. Available: <https://www.geeksforgeeks.org/clustering-in-machine-learning/>. [Último acceso: 28 Julio 2020].
- [93] Raman, «GeeksforGeeks,» [En línea]. Available: <https://www.geeksforgeeks.org/ml-linear-discriminant-analysis/>. [Último acceso: 28 Julio 2020].
- [94] A. Amini y D. Rus, «Reconsider,» 9 Mayo 2018. [En línea]. Available: <https://reconsider.news/2018/05/09/ai-researchers-allege-machine-learning-alchemy/>. [Último acceso: 28 Julio 2020].
- [95] A. Mahajan, «Makshay,» 25 Marzo 2020. [En línea]. Available: <https://makshay.com/neural-network-basics-gradient-descent>. [Último acceso: 28 Julio 2020].
- [96] «suniljangirblog,» [En línea]. Available: <https://suniljangirblog.wordpress.com/2018/12/13/variants-of-gradient-descent/>. [Último acceso: 28 Julio 2020].
- [97] GeeksforGeeks, «GeeksforGeeks,» 18 Mayo 2020. [En línea]. Available: <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>. [Último acceso: 01 Septiembre 2020].
- [98] P. J. Bota, C. Wang, A. L. N. Fred y H. P. D. Silva, «A review, current challenges, and future possibilities on emotion recognition using machine learning and physiological signals,» *IEEE Access*, vol. 7, pp. 140990 - 141020, 2019.
- [99] R. P. Jaakko Malmivuo, *Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields*, New York: Oxford University Press, 1995.
- [100] I. Bisio, A. Delfino, F. Lavagetto y M. Marchese, «Opportunistic detection methods for emotion-aware smartphone,» 2013, pp. 53-85.
- [101] I. Pico, «Psicología online, neurociencia y ayuda psicológica,» 6 Junio 2016. [En línea]. Available: <https://psicopico.com/la-rueda-las-emociones-robert-plutchik/>. [Último acceso: 2 Junio 2020].
- [102] D. Fujita y A. Suzuki, «Evaluation of the Possible Use of PPG Waveform Features Measured at Low Sampling Rate,» *IEEE Access*, vol. 7, pp. 58361-58367, 2019.
- [103] A. Choi y H. Shin, «Photoplethysmography sampling frequency: pilot assessment of how low can we go to analyze pulse rate variability with reliability?,» *Physiological Measurement*, vol. 38, n° 3, p. 586–600, 2017.
- [104] A. Dupre, S. Vincent y P. Iaizzo, «Basic ECG Theory, Recordings, and Interpretation,» de *Handbook of Cardiac Anatomy, Physiology, and Devices*, Totowa, Humana Press, 2005, pp. 191-201.

- [105] Task Force of The European Society of Cardiology and The North American Society of Pacing and Electrophysiology, «Heart rate variability: standards of measurement, physiological interpretation, and clinical use,» *European Heart Journal*, vol. 17, pp. 354-381, 1996.
- [106] AMS, *AS7030B Datasheet*, 2020.
- [107] T. Instruments, «C2000 real-time control MCUs,» [En línea]. Available: <https://www.ti.com/microcontrollers-mcus-processors/microcontrollers/c2000-real-time-control-mcus/overview.html>. [Último acceso: 28 04 2021].
- [108] «Wikipedia,» [En línea]. Available: [https://en.wikipedia.org/wiki/Spectral\\_leakage](https://en.wikipedia.org/wiki/Spectral_leakage). [Último acceso: 12 Junio 2021].

## 9 Índice alfabético

<b>A</b>	
Agrupamiento .....	39
Algoritmo Levinson-Durbin .....	28
Análisis discriminante lineal .....	51
Análisis Discriminante Lineal pseudoinverso .....	50
Aprendizaje Automático .....	36
Aprendizaje no supervisado .....	39
Aprendizaje supervisado .....	38
<b>C</b>	
Características .....	18, 36
Círculo circunflejo de Russel .....	58
Clase .....	38, 51
Clasificación .....	38
Clasificación dicotómica multinivel específica a emociones .....	II, 55
Clasificaciones binarias .....	38
Coefficientes de predicción hacia adelante .....	27
Coefficientes de predicción hacia atrás .....	27
Coefficientes de reflexión .....	28
Coefficientes del modelo AR .....	26
Componente espectral .....	20
Computación afectiva .....	1
Conjunto de entrenamiento .....	42
Conjunto de prueba .....	42
Conjunto de validación .....	42
<b>Cuantiles</b> .....	19
<b>D</b>	
Densidad espectral de energía .....	25
Densidad espectral de potencia .....	25
Método de Burg .....	29
Método de Welch .....	30
Métodos no paramétricos .....	26
Métodos paramétricos .....	26
Densidad Espectral de potencia modelo Autoregresivo .....	26
Derrame espectral .....	20
Desviación estándar .....	19
Detección de anomalías .....	40
Dimensionalidad .....	36
Discriminante Lineal de Fisher .....	47
<b>E</b>	
Emoción	
acercamiento cognitivo a las emociones .....	57
emociones como construcciones sociales .....	57
emociones como expresiones .....	56
emociones como personificaciones .....	56
Emoción .....	56
Emoción	
modelo discreto .....	57
Emoción	
modelo dimensional .....	58
Emoción	
clasificador dependiente .....	58
Emoción	
clasificador independiente .....	59
Emoción	
estimuladores .....	59
Entrenamiento .....	36, 41, 51
Envoltorios .....	44
Error de predicción hacia adelante .....	27
Error de predicción hacia atrás .....	28
<i>exactitud</i> .....	41
<b>F</b>	
Función de costo .....	36
<b>L</b>	
Lattice .....	28
<b>M</b>	
Maniquí de Auto Valoración .....	60
Media aritmética .....	18
Método de máxima entropía .....	Véase modelo Autoregresivo
<b>O</b>	
Onset .....	Véase Tiempo de levantamiento
<b>P</b>	
Percentil .....	19
Periodograma .....	25
Promedio .....	18
Proyección Fisher .....	47
<b>R</b>	
Raíz de la media cuadrática .....	19
Reducción dimensional .....	40, 43
Regresión .....	38
<b>S</b>	
Selección de características .....	40, 44
Selección secuencial hacia adelante .....	44

Selección secuencial hacia delante flotante.....	46
Sistema multimodal .....	1
Sobreajuste .....	40
Subajuste.....	40

## T

Teorema de Parseval.....	25
Tiempo de levantamiento .....	18
Transformada de Fourier de tiempo discreto .....	20
Transformada discreta de Fourier	
resolución.....	22
Transformada rápida de Fourier .....	22
decimación en tiempo.....	24
factor twiddle.....	22

operación mariposa.....	24
radix.....	22

## V

Validación cruzada de k iteraciones.....	42
Validación cruzada de k iteraciones con barajeo .....	42
Validación cruzada dejando uno afuera .....	43
Validación simple .....	42
Varianza.....	19

## Z

Zero padding .....	22
--------------------	----