

**UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA**

**FACULTAD DE INGENIERÍA**

**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

---



**BIOSENSADO PARA LA DETECCIÓN DE EMOCIONES:  
CLASIFICACIÓN DE EVENTOS PARA DOS TECNOLOGÍAS**

POR:

**FLORENTINO ALVÍDREZ LOZANO**

**TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE**

**MAESTRO EN INGENIERÍA EN COMPUTACIÓN**



Biosensado para la detección de emociones: clasificación de eventos para dos tecnologías. Tesis presentada por Florentino Alvídrez Lozano como requisito parcial para obtener el grado de Master en ingeniería en computación, que ha sido aprobada y aceptada por:

---

**M.I. Javier González Cantú**  
Director de la Facultad de Ingeniería

---

**Dr. Fernando Rafael Astorga Bustillos**  
Secretario de Investigación y Posgrado

---

**M.S.I. Karina Rocío Requena Yáñez**  
Coordinadora Académica

---

**Dr. Fernando Martínez Reyes**  
Director de Tesis

Fecha 6/3/2019

Comité:

**Dr. Luis Carlos González Gurrola**

**Dr. Raymundo Cornejo García**

**Dr. Juan Gabriel González Serna**

**Dr. Fernando Martínez Reyes**

© Derechos Reservados  
**Florentino Alvídrez Lozano**  
Rincón del picacho #9312  
Colonia Rincones del picacho.

Junio 2019

## Dedicatoria

*A mis padres, por ser los pilares de valores que a mi hermano y a mí nos hacen seguir siempre adelante, educándonos con el ejemplo, brindándonos su cariño, sabiduría y su apoyo incondicional. Este logro también es de ustedes, por su esfuerzo y dedicación, les agradezco muchísimo por hacer todo esto posible.*

## Agradecimientos

Este trabajo fue gracias a un gran número de personas que formaron parte importante en el camino hacia la realización de este proyecto. A mis padres, mi hermano y también a toda mi familia, que siempre han estado brindándome su apoyo, a mi querida novia, compañeros y amigos que me ayudaron a seguir adelante y sobre todo que soportaron las limitaciones que la carga de trabajo exigía.

A mi director de tesis, por su confianza, consejos, dedicación y por enseñarme desde un inicio el gusto por la investigación. De igual manera mis profesores, personas admirables, las cuales dedicaron su tiempo y esfuerzo en transmitirnos de la mejor manera sus conocimientos y que se empeñaron en hacer que diéramos lo mejor de nosotros mismos.

A los doctores y personal del Instituto de Investigación e Innovación en Bioingeniería, por darme la oportunidad de trabajar con ellos, tener experiencias muy gratas y por los conocimientos que adquirí durante mi estadía en España.

Finalmente, gracias a la UACH y a Conacyt, por brindar el apoyo económico para poder realizar mis estudios de posgrado y por la calidad de los mismos.

## Resumen

Las emociones son una parte fundamental en nuestra forma de pensar y de nuestro comportamiento. Ellas mismas accionan estados fisiológicos, los cuales pueden ser medidos y clasificados para identificar el tipo de emoción que una persona experimenta. En este trabajo se utilizaron imágenes del set de datos IAPS (International Affective Picture System) para evocar diferentes tipos de emociones a 18 participantes. Dos tipos de tecnologías de sensado (e-Health y Bitalino) fueron utilizadas en el estudio, con el objetivo de utilizar señales fisiológicas como la frecuencia cardíaca y la actividad dérmica de la piel para identificar y clasificar Valencia alta y baja, así como Arousal alto y bajo. Los clasificadores implementados fueron: Bosques aleatorios (Random-Forest), Maquinas de vectores de soporte (SVM), redes neuronales (MLP), k-vecinos más cercanos (KNN) y Arboles de decisión (Decision-Tree). Dentro de los cuales se lograron identificar con una precisión de 51.06% (con Decision-Tree) en Bitalino y un 54.81% (con KNN) en e-Health la Valencia alta y baja, mientras que para detectar Arousal, se logró un 70.67% (con Random-Forest) en Bitalino y un 55.85% (con KNN) en e-Health.

## Índice de Contenido

Dedicatoria .....	3
Agradecimientos.....	4
Resumen .....	5
Índice de Contenido .....	6
Índice de Tablas .....	8
Índice de Figuras .....	9
Capítulo 1 Introducción .....	12
1.1. Antecedentes .....	13
1.2. Hipótesis o pregunta de investigación.....	16
1.3. Objetivos .....	16
1.3.1. General.....	16
1.3.2. Específicos.....	16
Capítulo 2 Fundamentación Teórica.....	17
2.1.1. Actividad Electro dérmica (EDA) .....	17
2.1.2. ECG .....	18
2.1.3. Señales fisiológicas en el dispositivo e-Health Sensor Shield V2.0. [17].....	19
2.1.4. Señales fisiológicas en el dispositivo BITalino (r)evolution Plugged Kit BLE [2]19	
2.1.5. Configuración del estudio.....	19
2.1.6. Información sociodemográfica .....	19
2.1.7. Cuestionario Self-Assessment Manikin (SAM) .....	20
2.1.8. Perceptrón multicapa (MLP por sus siglas en inglés) .....	20
2.1.9. Máquinas de vectores de soporte.....	21
2.1.10. Árboles de decisión .....	22
2.1.11. Bosques aleatorios (Random Forest) .....	22
2.1.12. K vecinos más cercanos (KNN por sus siglas en inglés) .....	23
2.1.13. Mutual_info_classif .....	24
Capítulo 3 Metodología .....	25
3.1. Configuración del estudio .....	25
3.2. Interface del usuario .....	27
3.2.1. Selección de imágenes IAPS .....	28
3.3. Tecnologías de sensado.....	33

3.3.1. Colocación de electrodos.....	33
3.3.2. Recolección de datos .....	35
3.3.3. Visualización de datos .....	36
3.4. Etapa de pre-procesamiento .....	37
3.4.1. Ajustar frecuencia de muestreo y formato.....	37
3.4.2. Pre procesamiento ECG.....	39
3.4.3. Pre procesamiento EDA .....	45
3.5. Creación de set de datos .....	54
3.5.1. Normalización y estandarización.....	56
3.5.2. Combinar archivos.....	57
3.5.3. Preparación del Dataset .....	58
3.6. Clasificación.....	59
3.6.1. Proceso de clasificación.....	60
Capítulo 4 Resultados .....	63
Capítulo 5 Discusión de resultados.....	65
Capítulo 6 Conclusiones y trabajo futuro .....	67
Referencias .....	69
Glosario .....	73
Apéndices.....	74
Apéndice A.....	74
Guardar documento .....	74
Controlar diapositivas.....	74
Obtener botón activo .....	75
Mostrar cuestionario SAM .....	77
Apéndice B (Código Arduino para e-Health) .....	77
Código para Clasificar las Señales de E-Health .....	78
Código para clasificar señales Bitalino .....	81
Apéndice C - Downsample .....	84
Apéndice D - Determinar PPM en Matlab .....	84
Código en Matlab para determinar el promedio de pulsaciones por ventana (evento) ....	84
Apéndice E.....	85
Diferentes clasificadores.....	85

Apéndice F - Clasificación Alto, medio y Bajo .....	87
---	----

## Índice de Tablas

Tabla 2-1: Cuestionario sociodemográfico .....	20
Tabla 3-1: Extracto de la tabla IAPS con clasificaciones tomado de <i>International Affective Picture System: All subjects</i> [22] .....	28
Tabla 3-2: Tabla de grupos para la clasificación de IAPS (ejemplo) [Val=Valencia;Ar=Arousal;Dom=Dominancia] .....	29
Tabla 3-3: Selección de imágenes IAPS - VALENCIA .....	29
Tabla 3-4: Ejemplo de tabla generada con valores SAM .....	32
Tabla 3-5: Extracto del registro de datos generado para e-Health .....	35
Tabla 3-6: Extracto del registro de datos generado para Bitalino.....	35
Tabla 3-7: Ejemplo de tabla generada al reducir la frecuencia de muestreo de la señal E-Health.....	38
Tabla 3-8: Ejemplo de tabla generada incluyendo la hora de cada registro en Bitalino.....	39
Tabla 3-9: Ejemplo de tabla de set de datos por participante .....	55
Tabla 3-10: Ejemplo de señal ECG normalizada.....	56
Tabla 3-11: Ejemplo de datos normalizados EDA .....	57
Tabla 3-12: Ejemplo de set de datos (Bitalino_Dataset.csv) .....	57
Tabla 3-13: Ejemplo de formato del dataset final a ser utilizado en la etapa de clasificación (Bitalino_Dataset_Valencia.csv) .....	59
Tabla 3-14: Trabajos relacionados respecto a clasificación de emociones.....	59
Tabla 3-15: Características seleccionadas por dataset .....	62
Tabla 4-1: Resultados de clasificación (cross validation).....	63
Tabla 6-1: Glosario .....	73
Tabla 0-1: Ejemplo para comparación entre sensores .....	87
Tabla 0-2: Tabla comparativa SVM con diferentes kernels .....	87

## Índice de Figuras

Figura 2-1: Funciones del sistema simpático y parasimpático .....	17
Figura 2-2: Comportamiento normal del corazón. [4] .....	18
Figura 2-3: Ejemplo de Self-Assessment Manikin utilizado para medir IAPS [7].....	20
Figura 2-4: Ejemplo de una red neuronal totalmente conectada. [18].....	21
Figura 2-5: Ejemplo de una Máquina de Vectores de Soporte (SVM) [19] .....	21
Figura 2-6: Ejemplo de Árboles de decisión [20] .....	22
Figura 2-7: Ejemplo de Random Forest [20] .....	23
Figura 2-8: Ejemplo de KNN [20] .....	23
Figura 3-1: Esquema general de metodología .....	25
Figura 3-2: Sala de experimentación .....	26
Figura 3-3: Diagrama a bloques de las actividades desarrolladas para evaluar la detección de estados afectivos mediante tecnologías de biosensado.....	27
Figura 3-4: Selección de imágenes IAPS para la creación de Set_A y Set_B.....	30
Figura 3-5: Esquema de uso general de la herramienta de auto-evaluación SAM .....	31
Figura 3-6: Ciclo de eventos que se desarrollan en la interface del usuario.....	31
Figura 3-7: La colocación de los electrodos ECG para la tecnología e-Health (izquierda) y para la tecnología Bitalino (derecha). Para esta última, observe la ubicación del electrodo (+). .....	34
Figura 3-8: Colocación de los electrodos EDA en el dedo índice y medio. [17] .....	34
Figura 3-9: Visualización de señales ECG registradas con la tecnología e-Health (izquierda) y con la tecnología Bitalino (derecha) .....	36
Figura 3-10: Señal EDA recolectada por e-Health (izquierda) y por Bitalino (derecha) .....	37
Figura 3-11: Diagrama de flujo para calcular los pulsos por minuto (PPM) de la frecuencia cardíaca de cada ventana.....	40
Figura 3-12: Importando datos de ECG.....	41
Figura 3-13: Eliminación de tendencia en señal ECG. E-Health (Izquierda), Bitalino (Derecha).....	42

Figura 3-14: Señales ECG de salida para un filtro pasabajas. Izquierda e-Health, derecha Bitalino. En cada imagen en la parte superior se encuentra la señal original y el parte inferior la señal filtrada. ....	42
Figura 3-15: Fase compleja de frecuencia cardíaca [27] .....	43
Figura 3-16: Identificación de eventos R en la señal ECG registrada con e-Health (izquierda) y Bitalino (derecha). El evento R se marca con un triángulo invertido sobre la señal filtrada. ....	44
Figura 3-17: Identificación de intervalos R para la línea base, e-health (izquierda) y bitalino (derecha) .....	45
Figura 3-18: Conductancia eléctrica de la piel solo está definida por la velocidad de cambio de la señal en respuesta a estímulos. Sin embargo, en la respuesta galvánica de la piel existen otros componentes como la conductancia tónica y fásica. [29].....	46
Figura 3-19: Pre-procesamiento de la señal EDA.....	46
Figura 3-20: Ejemplo de cómo se tomaron ventanas de eventos para importar a Ledalab .....	48
Figura 3-21: Ejemplo de Archivo SC_Marker_participante(n) .....	48
Figura 3-22: Pre-procesamiento de la señal EDA.....	49
Figura 3-23: Importación de datos EDA a Ledalab .....	50
Figura 3-24: Downsampling a 10 Hz.....	50
Figura 3-25: Ejemplo importando datos EDA .....	51
Figura 3-26: Señal inicial en el ADC, con un error de 22.013. ....	52
Figura 3-27: Tercera optimización en el ADC, con un error de 0.399 .....	52
Figura 3-28: Después del análisis ADC - ejemplo.....	53
Figura 3-29: Exportando resultados EDA.....	54
Figura 3-30: Actividades desarrolladas para preparar el set de datos a utilizar en la etapa de clasificación. ....	54
Figura 3-31: Creación de archivo de set de datos desde Tabla 3-4 .....	55
Figura 3-32: Ejemplo de separar el conjunto de datos completo de Bitalino en valencia y Arousal .....	58
Figura 3-33: Diagrama de flujo para la clasificación.....	61
Figura 3-34: ejemplo de selección de características.....	61

Figura 4-1: Grafica comparativa entre sensores – Valencia .....	64
Figura 4-2: Grafica comparativa entre sensores - Arousal .....	64
Figura 0-1: Ventana de Unity generada con el código anterior. ....	76
Figura 0-2: Ventana de Unity donde se controlan los parámetros de la interface (código anterior).....	77
Figura 0-3: Comparación ECG con valores alto, medio y bajo .....	88
Figura 0-4: Comparación GSR para valores alto, medio y bajo .....	88

# Capítulo 1

## Introducción

La medición de emociones es en base a la percepción de las personas, y son difíciles de medir sin que la misma persona nos diga directamente o sea el resultado de una autoevaluación. Algunas investigaciones han reportado el uso de señales fisiológicas para la detección de estados afectivos. En estos trabajos, se denota el uso de sensores de electroencefalografía (EEG), electromiografía (EMG), actividad dérmica de la piel (EDA), frecuencia cardíaca, temperatura corporal, frecuencia de la respiración, entre otros, en los cuales se hacen clasificaciones con los sensores en forma individual o en conjunto, para realizar la detección de estados afectivos. En la historia de la psicofisiología, EDA es uno de los sensores más utilizados en investigación [1]. En este estudio se hizo uso de los sensores ECG y EDA para realizar una comparación entre dos sensores de bajo costo (e-Health y Bitalino) con respecto a su clasificación en la detección de emociones. A lo largo del documento se puede apreciar la comparación de los resultados de cada proceso entre ambos sensores. Esto, para comparar los resultados obtenidos por estas dos tecnologías con los mismos parámetros, aplicando los mismos procedimientos de filtrado, procesamiento y clasificación. Los contextos de aplicación para este trabajo son muy variados, puesto que podría ayudar a los proyectos en los que la percepción de las personas sea importante. Algunos de los usos de los trabajos reportados son por ejemplo en marketing [2], en donde se realizó un estudio de toma de decisiones y actitud en general en el turismo, haciendo uso de realidad virtual. Otros trabajos usan contextos psicológicos, de relajación, médicos (en quimioterapias), educativos ([3], [4], [5], [6]), entre otros.

En este trabajo de tesis, se estudia la clasificación de estados afectivos utilizando como medio de inducción de estos estados el *International Affective Picture System*, IAPS, dos sistemas de recolección de datos fisiológicos (frecuencia cardíaca y respuesta galvánica de la piel) y la aplicación de técnicas de clasificación. Específicamente, se busca comparar el nivel de clasificación que puede realizarse sobre los componentes afectivos de Arousal y Valencia a partir de los datos recolectados con dos tecnologías de biosensado: e-Health y Bitalino.

## 1.1. Antecedentes

La forma en que percibimos nuestro entorno varía entre cada persona, y si se quisieran comparar las emociones que experimentan varios individuos, sería necesario de utilizar una medida estándar para su comparación. En la psicología, se han establecido ciertos parámetros para que las emociones pudieran ser catalogadas en valencia, activación y control (o dominancia). Estas emociones son medidas como positivas o negativas como, por ejemplo: la alegría o interés hace que la valencia sea positiva, mientras que la valencia negativa se manifiesta con miedo, enojo o disgusto [7]. Debido a que las emociones también presentan estados fisiológicos, los esfuerzos de investigación realizados han sido detectar estos parámetros fisiológicos de las personas y utilizarlos en terapias psicológicas relacionadas con el comportamiento y las emociones.

En [8] se realizó un estudio en donde se clasificaron estados afectivos como valencia y arousal, esto por medio de señales EDA durante la presentación de estímulos sonoros (obtenidos de la base de datos IADS [9]). 25 voluntarios entre 25 y 35 años fueron sometidos cada uno a 8 sesiones (una de descanso al inicio de 5 minutos y tres de Arousal, donde se fueron intercalando por sesiones neutras). Para tratar la señal se utilizó la normalización  $Z$ -Score, luego se extrajeron las características y un análisis estadístico. El clasificador utilizado fue el KNN utilizando leave-one-out cross-validation. Los resultados obtenidos de la matriz de confusión fueron de un 77.33% en la detección de Arousal y un 84% para valencia utilizando las señales EDA.

En [10] se muestra el uso de ECG y SC para detectar estados emocionales como disgusto, felicidad, tristeza y estados neutros. 10 sujetos (3 mujeres y 7 hombres) visualizaron y escucharon 5 videos (entre 110 y 430 segundos) intercalados con periodos de reposo con música (118 segundos), mientras se registraban las señales mencionados anteriormente. Para la clasificación utilizaron un discriminante lineal, la cual obtuvo una precisión de 64% al detectar los estados emocionales.

En [11], utilizaron sensores EMG, GSR y EEG de “bajo costo”, en donde se recolectaron y procesan datos para identificar estados afectivos de valencia y arousal. 19 personas en un rango de edad entre los 20 y 40 años observaron y escucharon videos musicales por 2 horas, divididas en 4 sesiones de 30 minutos, representando los primeros 30 minutos como la línea base. Los datos recolectados en el estudio se filtraron y clasificaron

utilizando algoritmos como SVM, árboles de decisión (J48) y Naive Bayes. El clasificador SVM fue el algoritmo que obtuvo el mejor porcentaje de clasificación (F1-score de 56.3%). Los autores distinguen que el sensor GSR permitió clasificar de mejor forma el estado de arousal (F1-score de 63%) mientras que los datos EEG permitieron clasificar mejor el estado de valencia (F1-score de 56.3%).

En [2] se hace uso de las tecnologías EDA y ECG para la investigación de marketing en el turismo en Realidad Virtual (RV). Esto tomando en cuenta las emociones y estudiando la toma de decisiones. El estudio comprendía a 101 participantes divididos en dos grupos, experimental (visualizando estímulos en RV) y de control (visualizando estímulos en una computadora). El número de participante fue 46 y 55 respectivamente. A cada uno se les tomó una línea base, después se les mostró un video promocional en 360 grados, y al final se les pidió llenar un cuestionario en línea. Se obtuvo un incremento significativo en las señales fisiológicas en el grupo que visualizó los videos en realidad virtual.

El estudio reportado en [12], es una réplica del estudio “International Affective Picture System (IAPS): Affective Ratings of Pictures and Instruction Manual (Rep. No. A-8)”. En donde, utilizando imágenes IAPS, fueron catalogadas en 4 dimensiones las emociones en el self-report (alegre, satisfecho, enojado y aburrido). Utilizaron también la escala SAM de self-report. Muestran una metodología que se puede apreciar en otros estudios similares para la presentación de imágenes IAPS y SAM. Esto es, 58 sesiones con la siguiente configuración: una imagen que muestra el texto “Prepárese para la siguiente diapositiva” (5 segundos), seguido de una Imagen IAPS al azar (6 segundos), self-report SAM con una escala de 9 puntos (15 segundos). La población comprendía a 6 participantes, de los cuales solo obtuvieron 3 sets de datos útiles debido a problemas técnicos. Aún con esto observaron que cuando se mostraron imágenes de enojo o de alegría a los participantes, se generaron picos en los sensores de EDA, al igual que en el estudio que se estaba replicando.

El estudio [13], fue realizado con 90 imágenes IAPS agrupadas en 3 bloques de 30, en donde cada bloque fue mostrado aleatoriamente durante la presentación del estímulo con la siguiente configuración: Una imagen de espera con una cruz de fijación (10-15 segundos), seguido de una imagen IAPS (6 segundos) y la validación del self-report SAM. La configuración del estudio es parecida a otras investigaciones relacionadas, en donde se utiliza el sistema IAPS y el self-report SAM. La muestra fue de 39 participantes, de los cuales

fueron excluidos 8 debido a que generaron menos de un 5% en relación-evento de conductancia de la piel en las imágenes IAPS. Para descartarlos, se basaron en un estudio que dice que hay una proporción sustancial de adultos sanos considerados “estables” que muestran respuesta Electrodermica, por lo que existe un porcentaje que no muestra cambios considerables en la respuesta galvánica de la piel.

Como resultado del estudio, se encontró que las personas con Salience Network Connectivity más fuerte, reportaron experiencias más intensas de Excitación subjetiva (Arousal subjetivo) mientras veían imágenes que se conoce que evocan experiencias de alto Arousal positivo y negativo. Estos resultados están enfocados a otro tema neurológico, pero en cuestión de SCR, en los individuos se observó que a mayor actividad SCR también se reportó un Arousal más intenso.

Como se muestra en [14], las características de un entorno pueden ser adaptadas basándose en un sistema de biosensado (parámetros fisiológicos) en tiempo real. Para esto, se pueden tomar en cuenta sensores como la respuesta galvánica de la piel, la frecuencia cardíaca, temperatura, electromiografía, entre otros y hacer que las señales sean los parámetros de entrada al sistema. Con esto, es posible que el usuario tenga retroalimentación de su estado fisiológico mostrando esos datos en el ambiente o cambiando ciertos aspectos del entorno según las necesidades del estudio.

En cuanto ambientes adaptativos realizados, se puede encontrar el sistema EMMA [15], el en cual se presenta un escenario de un parque en donde diferentes técnicas como la narración, las imágenes presentadas y fragmentos de películas son usados para la inducción de estados de ánimo, mientras que presentan diferencias de colores, texturas y cambios en la ambientación. Esto es muy importante, ya que, es una herramienta que puede ser utilizada para diferentes propósitos, como comprender algunos fenómenos que tengan que ver con la psicopatología o problemas psicológicos en donde la adaptación del entorno es requerida.

Con el fin de realizar una identificación de emociones por medio de señales fisiológicas, en este estudio se pretende identificar y clasificar las señales adquiridas de los participantes al ser expuestos a cierto tipo de estímulos visuales.

## 1.2. Hipótesis o pregunta de investigación.

Las emociones inducidas con las imágenes del set de datos IAPS pueden ser medidas y clasificadas al mismo nivel utilizando datos recolectados por dos tipos de tecnologías de biosensado (E-Health y Bitalino).

## 1.3. Objetivos

### 1.3.1. General

Este estudio tiene como objetivo principal la identificación y clasificación de datos fisiológicos, ritmo cardíaco y actividad dérmica de la piel, en relación con estados afectivos, inducidos mediante el IAPS, *International Affective Picture System*.

### 1.3.2. Específicos

1. Seleccionar las tecnologías de biosensado y evaluar su desempeño con un sujeto de prueba
2. Seleccionar el sub-conjunto de imágenes IAPS e integrarlo con la interface gráfica, y pruebas preliminares
3. Reclutar participantes
4. Recolectar y filtrar señales fisiológicas
5. Clasificar estados afectivos

## Capítulo 2

### Fundamentación Teórica

En este apartado se muestran los conceptos teóricos necesarios para la comprensión y delimitación del tema.

#### 2.1.1. Actividad Electrodermica (EDA)

Se dice que la actividad dérmica de la piel refleja la activación del sistema simpático. El cuál, es el que genera una reacción automática conocida como “lucha o huida”. Al momento de esta reacción, se presentan cambios físicos en el cuerpo como: incremento en la frecuencia cardiaca y la presión arterial, cambios electrodermicos y de respiración, por mencionar algunos. [1]

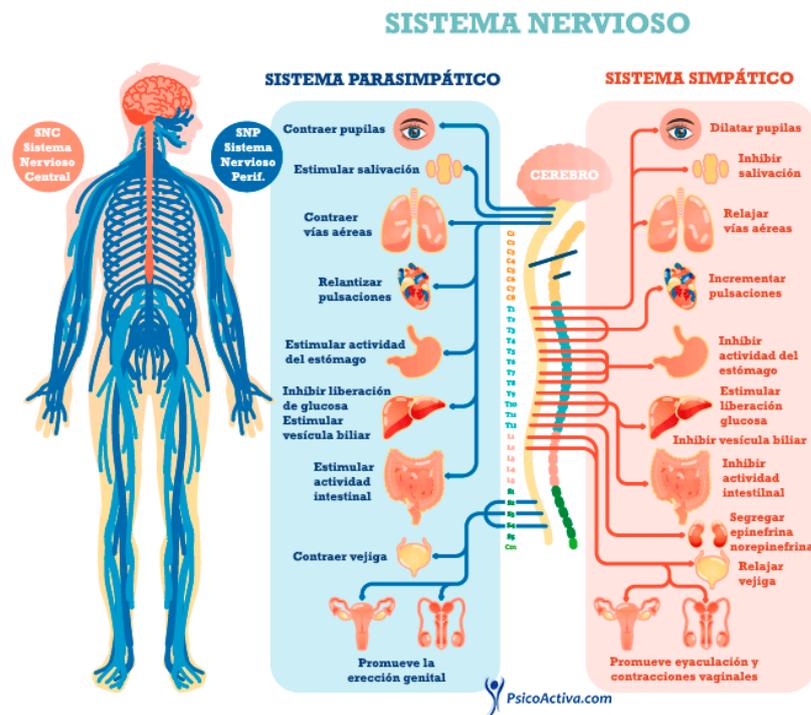


FIGURA 2-1: FUNCIONES DEL SISTEMA SIMPÁTICO Y PARASIMPÁTICO <sup>1</sup>

<sup>1</sup> <https://www.psicoadictiva.com/blog/wp-content/uploads/2016/11/sistema-simpatico-parasimpatico.png>

Una de las principales funciones de las glándulas sudoríparas es la regulación de la temperatura, y de esta función se hacen cargo la mayoría de las glándulas eccrinas. Estas glándulas también se tienen en las palmas de las manos y en la planta de los pies, aunque se piensa que la función que se realizan en estas partes del cuerpo es diferente y que podría ser para mejorar el agarre. Es por eso que se cree que medir la actividad dérmica en esas partes del cuerpo, es más significativa psicológicamente. [1]

### 2.1.2. ECG

El corazón tiene como función suministrar sangre oxigenada a todo el cuerpo. Modula la cantidad de sangre expulsada por los ventrículos para suministrar suficiente sangre para los órganos a medida que se va necesitando. [4]

Un electrocardiograma (ECG) registra los diferentes potenciales eléctricos generados por el corazón al pasar por el cuerpo, siendo una expresión de su distensión o agitación y no de su contracción. El proceso normal del corazón se muestra en la Figura 2-2. [4]

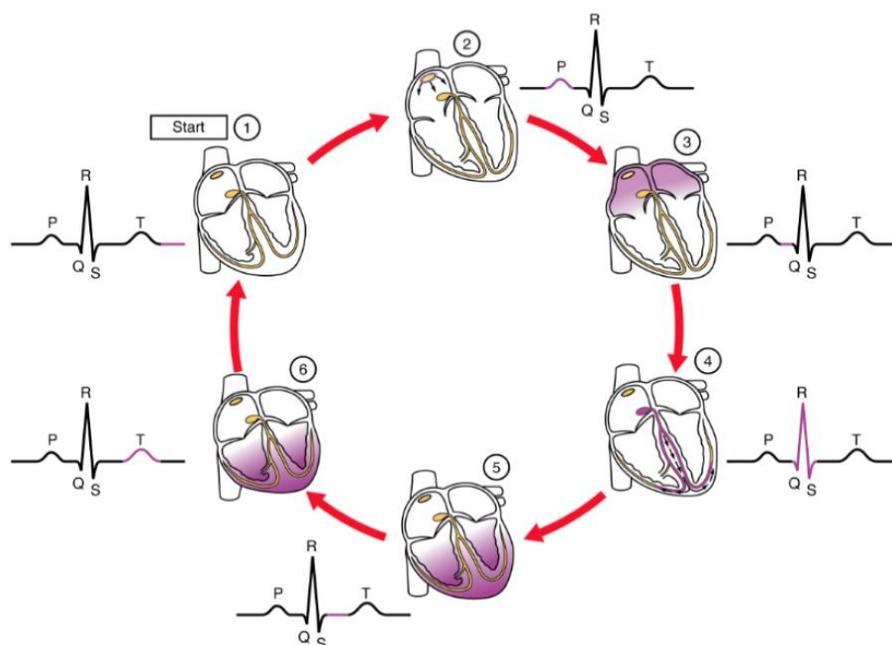


FIGURA 2-2: COMPORTAMIENTO NORMAL DEL CORAZÓN. [4]

Las diferentes fases mostradas en la imagen anterior corresponden al comportamiento normal del corazón, las pequeñas señales a los costados de cada fase, son la ejemplificación de la señal generada en voltaje, de esta manera se pueden contar las pulsaciones por minuto identificando la fase 4 (QRS) durante un periodo de tiempo. Como se menciona en [16], la

frecuencia cardiaca está relacionada con las condiciones emocionales y fisiológicas. Es por eso por lo que se tomará esta frecuencia para detectar el incremento o decremento de la frecuencia cardiaca influenciada por la inducción de emociones.

#### 2.1.3. Señales fisiológicas en el dispositivo e-Health Sensor Shield V2.0. [17]

- Sensor de ritmo cardiaco y oxigenación en la sangre
- Frecuencia respiratoria
- Sensor de temperatura corporal
- GSR (Respuesta galvánica de la piel)
- ECG (Frecuencia cardíaca del corazón)
- Oxigenación en la sangre

#### 2.1.4. Señales fisiológicas en el dispositivo BITalino (r)evolution Plugged Kit BLE [2]

- Electromiografía (EMG)
- Electrocardiografía (ECG)
- Actividad Electro dérmica (EDA)
- electroencefalograma (EEG)
- Acelerómetro (ACC)

#### 2.1.5. Configuración del estudio

##### 2.1.5.1. Criterios de inclusión:

- a) Hombres y mujeres mayores de 18 años.
- b) Aceptar su inclusión voluntaria en el estudio, previa información y comprensión de las características de éste, al firmar el correspondiente consentimiento informado.

##### 2.1.5.2. Criterio de exclusión:

- a) Cualquier motivo que incapacite a la persona para la correcta lectura y comprensión de los cuestionarios de evaluación.

#### 2.1.6. Información sociodemográfica

La información requerida para el presente estudio fue solo la mostrada en la siguiente tabla:

**TABLA 2-1: CUESTIONARIO SOCIODEMOGRÁFICO**

**1. Sexo**

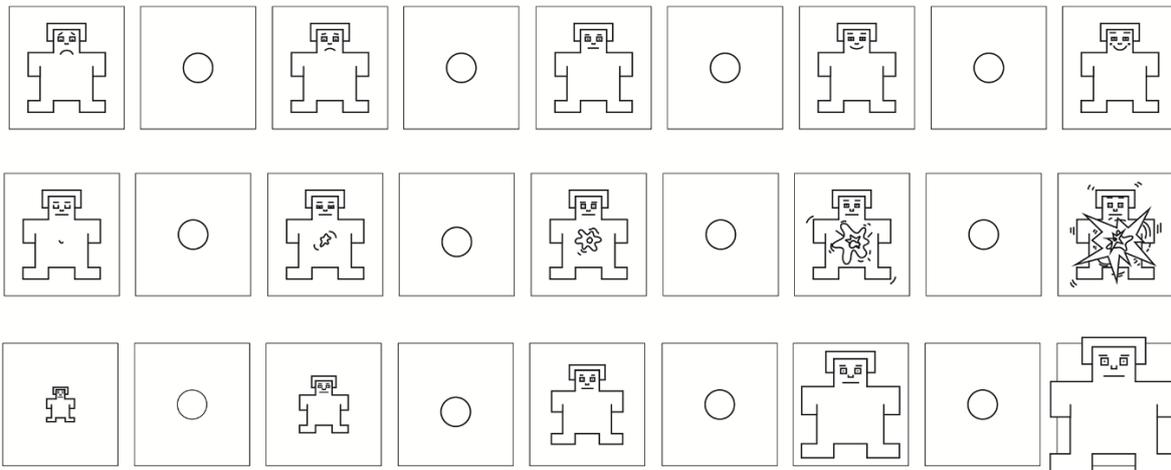
Hombre

Mujer

**2. Edad** \_\_\_\_\_

**2.1.7. Cuestionario Self-Assessment Manikin (SAM)**

El cuestionario SAM es utilizado para medir los valores de valencia, excitación (arousal) y dominancia (o control) de una persona. Esto es basándose en una escala de clasificación de 9 puntos por cada emoción. Se puede elegir entre una de las 5 figuras mostradas (Ver Figura 2-3), o alguno de los rectángulos entre las figuras (Por lo que en total son 9 puntos). [7]



**FIGURA 2-3: EJEMPLO DE SELF-ASSESSMENT MANIKIN UTILIZADO PARA MEDIR IAPS [7]**

**2.1.8. Perceptrón multicapa (MLP por sus siglas en inglés)**

Como se describe en [18], un perceptrón multicapa es una red neuronal artificial formada por múltiples capas, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón (también llamado perceptrón simple). En la figura 3 se muestra un ejemplo de una red neuronal artificial. Esta red consta de 3 capas, las tres neuronas de la izquierda forman la capa de entrada. Esta capa representa

los valores de entrada, las capas ocultas hacen el procesamiento y las neuronas de salida dan el resultado de la clasificación.

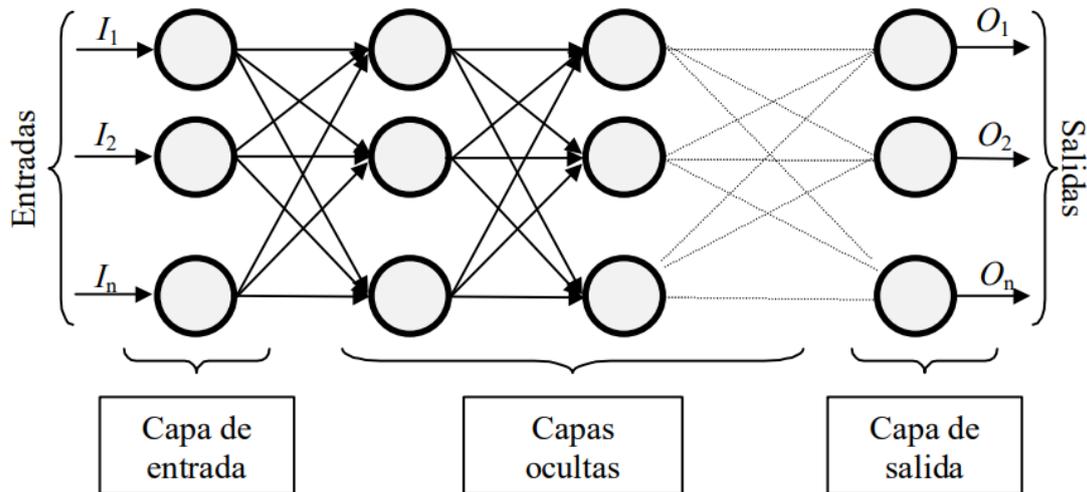


FIGURA 2-4: EJEMPLO DE UNA RED NEURONAL TOTALMENTE CONECTADA. [18]

El valor resultante en la capa de salida puede ser de dos formas diferentes. Esto quiere decir que la función de salida puede devolver el mismo número de valores de entrada (también se le llama función identidad) o un valor basándose en un umbral. La configuración binaria es que, si el valor resultante sobrepasa un umbral, se devuelve 1, de lo contrario se devuelve un 0. [18]

### 2.1.9. Máquinas de vectores de soporte

Las máquinas de vectores de soporte [19], en inglés llamadas Support Vector Machine (SVM), son un conjunto de algoritmos de aprendizaje supervisado y son utilizados para problemas de clasificación o regresión. Una SVM construye un hiperplano o un conjunto de ellos para poder diferenciar entre las clases, un ejemplo de ello se muestra en la Figura 2-5.

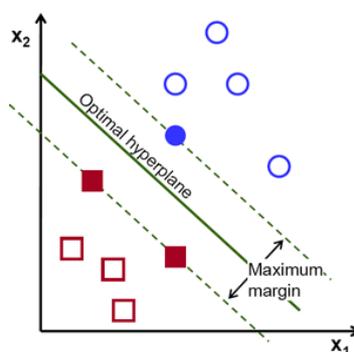


FIGURA 2-5: EJEMPLO DE UNA MÁQUINA DE VECTORES DE SOPORTE (SVM) [19]

### 2.1.10. Árboles de decisión

Un árbol de decisión es en sí un mapa en donde se colocan los posibles resultados de una serie de decisiones que están relacionadas. Este, se basa en condiciones, comienza con un nodo principal, del que derivan las distintas reglas, al cumplirse o no, se generan nuevas condiciones, como si fueran ramas se extienden con las distintas probabilidades hasta llegar a los resultados. Al terminar este proceso se puede apreciar (como en la Figura 2-6) que el resultado luce de forma parecida a un árbol. [20]

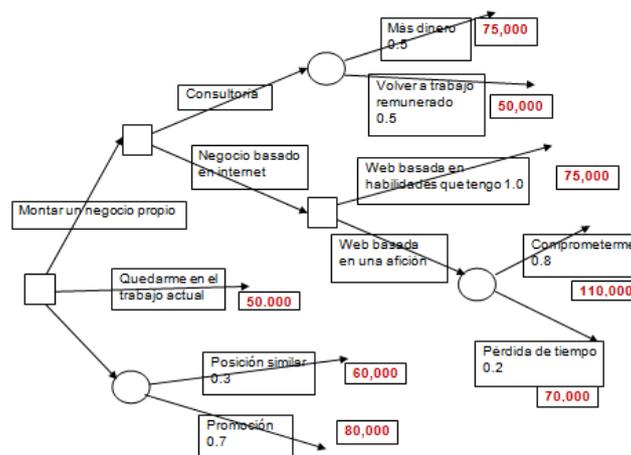


FIGURA 2-6: EJEMPLO DE ÁRBOLES DE DECISIÓN [20]

### 2.1.11. Bosques aleatorios (Random Forest)

Basándose en el algoritmo anterior, un bosque aleatorio es un meta estimador que realiza múltiples árboles de decisión con submuestras del set de datos proporcionado y por medio de un promedio de los árboles se obtiene el resultado de clasificación. En el ejemplo de la Figura 2-7, se muestra que la mayoría de los árboles de decisión generó como resultado la clase “B”, así, la clasificación final es la clase que tiene más votos. [20]

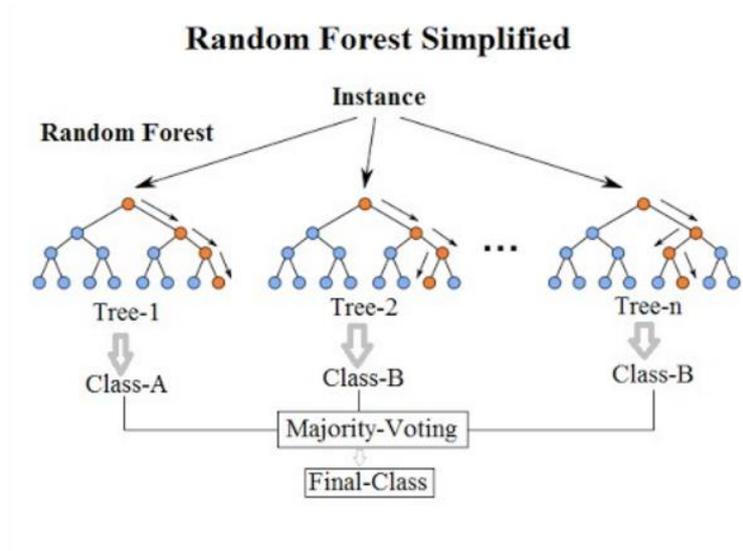


FIGURA 2-7: EJEMPLO DE RANDOM FOREST [20]

### 2.1.12. K vecinos más cercanos (KNN por sus siglas en inglés)

K vecinos más cercanos funciona encontrando un número predefinido de muestras de entrenamiento más cerca a la distancia de un nuevo punto y así, predecir el grupo al que pertenece en base a esas referencias.  $K$  (número de muestras) puede ser asignado por el usuario o puede cambiar dependiendo de la densidad de puntos. Entre ellos, la distancia utilizada puede ser cualquier medida, aunque la más utilizada es la distancia euclidiana. En la Figura 2-8 se puede observar que en este caso, un nuevo punto es asignado comparándose con 3 vecinos (puntos) más cercanos, al tener más muestras referentes a la clase 2, esta es la clase que se le asignará. [20]

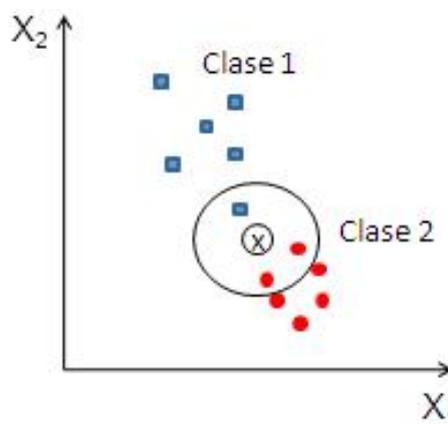


FIGURA 2-8: EJEMPLO DE KNN [20]

### 2.1.13. Mutual\_info\_classif

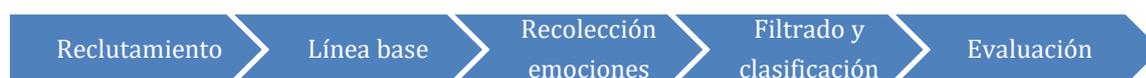
*mutual\_info\_classif*, es un algoritmo comprendido en el scikit-learn dentro de la selección de características. La información mutua, se refiere a que “entre dos variables aleatorias es un valor no negativo, que mide la dependencia entre las variables. Es igual a cero si y solo si dos variables aleatorias son independientes, y los valores más altos significan una mayor dependencia” [21]. La función de información mutua utiliza el algoritmo KNN para calcular la incertidumbre (entropía) de los k vecinos más cercanos.

## Capítulo 3

### Metodología

Partimos describiendo el escenario de biosensado de interés: Biosensado para la detección de emociones: clasificación de eventos para dos tecnologías. Específicamente, se utilizan dos tecnologías disponibles en el mercado, e-Health y Bitalino, para recolectar datos de frecuencia cardiaca y datos de la respuesta galvánica en la piel. Los datos recolectados a través de estos sensores son producto de la respuesta humana a la inducción de emociones que se realiza mediante una interface digital.

En esta sección se presenta la metodología utilizada para realizar el estudio a través del cual se realiza la recolección de bioseñales, el procesamiento de los datos, y la clasificación de los estados afectivos. Sin embargo, antes de ofrecer información acerca de la implementación técnica de cada una de estas fases, se describe la configuración del estudio, lo cual incluye el reclutamiento de participantes. Un esquema general que indica cada una de las fases del estudio se muestra en la Figura 3-1.



**FIGURA 3-1: ESQUEMA GENERAL DE METODOLOGÍA**

#### 3.1. Configuración del estudio

El estudio sobre la detección de estados afectivos se realizó en el Instituto de Investigación e Innovación en Bioingeniería (I3B) de la Universidad Politécnica de Valencia, institución en la cual tuve la oportunidad de realizar una estancia durante el periodo de junio a diciembre de 2018.

Para la fase de reclutamiento, se utilizó una base de datos de participantes previos a proyectos dentro del I3B. Se realizó entonces una convocatoria abierta y se mantuvo disponible durante los meses de septiembre a octubre de 2018. La invitación se envió directamente al correo electrónico de las 200 personas registradas en la base de datos. Del llamado a participar, 21 personas respondieron a la invitación. A cada persona que aceptó participar en el estudio se le asignó un horario para realizar la recolección de los datos. Cada

sesión de participación tenía una duración de 1 hora y consistía de las siguientes actividades: recepción, consentimiento informado, recolección de datos para la línea base y recolección de datos para estados afectivos.

En el día asignado, se ofreció a cada participante información por escrito en donde se detallaba el objetivo del estudio, el uso de tecnología y la disposición que se debía guardar para los sensores en el cuerpo de la persona, el tiempo requerido para la recolección, así como el uso que se haría de los datos recolectados. Posterior a esto, el participante firmó el documento de consentimiento informado. Posteriormente, fue conducido al área en la cual estaba instalado el equipo y donde se procedió al desarrollo del resto del estudio. Dentro de la sala, se contaba sobre un escritorio un equipo de cómputo (CPU, monitor, teclado y ratón) en el cual se desplegaría la información digital para inducir emociones. También se disponía de una silla de oficina en la cual se sentaba a la persona para proceder a la colocación de los sensores (ver Figura 3-2).

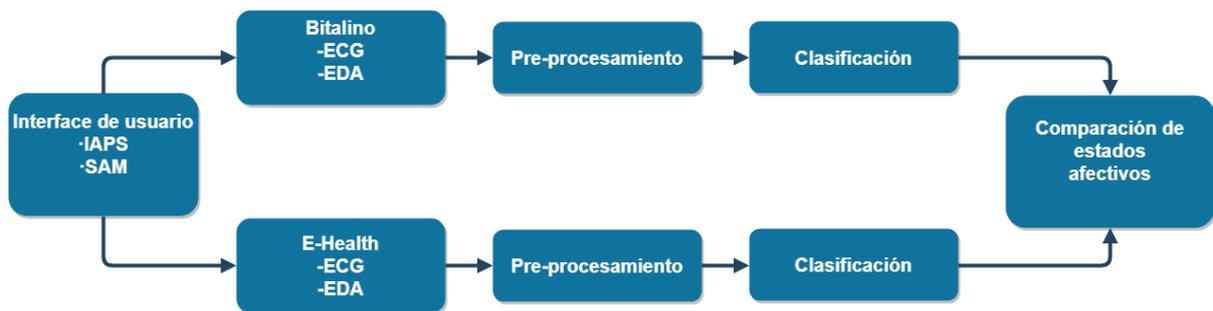


**FIGURA 3-2: SALA DE EXPERIMENTACIÓN**

Colocados los sensores, se procedía con la fase de recolección de datos para establecer la línea base. En esta actividad se solicitó al participante permanecer lo más posible en una postura relajada y con los ojos cerrados. El tiempo de recolección de datos fue de 3 minutos. Concluida esta etapa, se procedía con la recolección de datos asociados a

estados afectivos, etapa que tiene una duración aproximada de 45 minutos. Para esta etapa fue importante dar información al usuario del desarrollo de la misma. Por ejemplo, se debe describir la interface visual que se le presenta y el medio de interacción disponible con la misma, y la secuencia de eventos que ocurrirán durante el tiempo que dure esta etapa de recolección de datos. El participante se encontró en todo momento acompañado por un investigador, de manera que, si el participante experimentara alguna sensación de incomodidad o malestar, los dispositivos se retirarían inmediatamente. Al finalizar esta etapa de recolección de datos se procedió a remover los sensores y agradecer la participación de la persona.

En total, 21 participantes se presentaron para el estudio, de las cuales 3 fueron excluidos por pérdida de datos durante la recolección. Por lo que finalmente se consideraron a 18 personas, 15 del sexo femenino y 3 del sexo masculino, y con un rango de edad entre los 19 y 54 años. Esta muestra se dividió en dos grupos con el objetivo de contar con un grupo para el cual se utilizará la tecnología E-Health y otro grupo con la tecnología Bitalino. A grandes rasgos, la Figura 3-3 muestra las actividades llevadas a cabo en el estudio, las cuales son descritas más adelante.



**FIGURA 3-3: DIAGRAMA A BLOQUES DE LAS ACTIVIDADES DESARROLLADAS PARA EVALUAR LA DETECCIÓN DE ESTADOS AFECTIVOS MEDIANTE TECNOLOGÍAS DE BIOSENSADO.**

### 3.2. Interface del usuario

La interface del usuario se utiliza como el medio para la presentación del contenido digital orientado a inducir emociones. Las siguientes secciones muestran más a detalle el contenido que esta contiene.

### 3.2.1. Selección de imágenes IAPS

El sistema internacional de imágenes afectivas [22] (IAPS), se toma de base para el desarrollo de la interface del usuario. El IAPS cuenta con 1196 fotografías de animales, escenas de la naturaleza, caras humanas, escenas eróticas, comida, acontecimientos deportivos, cuerpos mutilados, escenas con violencia, etc. Este, se usa como técnica de evaluación para identificar fobias y trastornos de ansiedad e inducción de estados afectivos. Cada estímulo (imagen) está valorado en tres dimensiones (Valencia, Arousal y Dominancia) por el cuestionario SAM (que está comprendido por valores del 1-9), lo cual permite identificar el elemento afectivo asociado a cada imagen, en ejemplo de ello se puede ver en la Tabla 3-1.

**TABLA 3-1: EXTRACTO DE LA TABLA IAPS CON CLASIFICACIONES TOMADO DE INTERNATIONAL AFFECTIVE PICTURE SYSTEM: ALL SUBJECTS [22]**

<b>Description</b>	<b>Slide No.</b>	<b>Valence Mean(SD)</b>	<b>Arousal Mean(SD)</b>	<b>Dominance Mean(SD)</b>
Snake	1030	4.30(2.35)	5.46(2.43)	4.56(2.43)
Snake	1040	3.99(2.24)	6.25(2.13)	3.92(2.13)
Snake	1050	3.46(2.15)	6.87(1.68)	3.08(1.93)
Snake	1051	3.80(1.75)	5.95(1.98)	3.80(2.21)
Roaches	1275	3.30(1.65)	4.81(2.22)	5.11(2.23)
Rat	1280	3.66(1.75)	4.93(2.01)	5.05(2.20)
PitBull	1300	3.55(1.78)	6.79(1.84)	3.49(2.10)

Del conjunto de imágenes IAPS disponibles se crearon 27 grupos con las combinaciones de valores altos, medios y bajos, así como se muestra en la Tabla 3-2, en donde cada imagen fue insertada al grupo que correspondía. El criterio para catalogarlos fue: si valor < 4, se considera estado bajo, si su valor está entre 4 y 6, e inclusive, se toma como estado medio (o neutral), y si su valor > 6, se toma como un estado alto.

**TABLA 3-2: TABLA DE GRUPOS PARA LA CLASIFICACIÓN DE IAPS (EJEMPLO)**  
**[VAL=VALENCIA;AR=AROUSAL;DOM=DOMINANCIA]**

<b>Grupo 1</b> (Val_Alto Ar_Alto Dom_Alto)	<b>Grupo 2</b> (Val_Alto Ar_Alto Dom_Medio)	<b>Grupo 3</b> (Val_Alto Ar_Alto Dom_Bajo)	<b>Grupo 4</b> (Val_Alto Ar_Medio Dom_Alto)	<b>Grupo 5</b> (Val_Alto Ar_Medio Dom_Medio)	...	<b>Grupo 26</b> (Val_Bajo Ar_Bajo Dom_Medio)	<b>Grupo 27</b> (Val_Bajo Ar_Bajo Dom_BAjo)
5010		1333	4652	3001	...	2053	2301
		1603	4656	3060	...	3017	2456
		1900	4687	3071	...	3550	

Una vez hecho esto, se seleccionaron 4 imágenes para cada categoría, es decir, de cada grupo de la Tabla 3-2, se crearon 3 tablas, una para Valencia, otra para Arousal y una última para Dominancia, un ejemplo de ello es la Tabla 3-3 para Valencia.

**TABLA 3-3: SELECCIÓN DE IMÁGENES IAPS - VALENCIA**

<b>Categoría/Grupo</b>	<b>Grupo 1</b>	<b>Grupo 2</b>	<b>Grupo 3</b>	<b>Grupo 4</b>	...
<b>Bajo</b>	5010		1333	4652	...
			1603	4656	
			1900	4687	
			2240	4690	
<b>Medio</b>			1604	4607	...
			2035	4680	
			2360	8034	
			2598	8470	
<b>Alto</b>			5760		...
			8190		

A partir de tener las tres tablas como la Tabla 3-3 (para Valencia, Arousal y Dominancia), se separaron en dos conjuntos de imágenes (denominados Set\_A y Set\_B). Recuerde que tenemos dos grupos de participantes y que en cada grupo se utilizará o la tecnología e-Health o la tecnología Bitalino. Para el conjunto A, se tomaron los valores de la primera y segunda fila de cada grupo. Para el Set\_B, se tomaron las filas 3 y 4. En caso que solo haya valores en la fila 1 y 2 de cada grupo, se dividieron en dos y se incluirán en el Set\_A y Set\_B, cuando hay un solo valor, se toma en ambos sets. Tal es el caso de la Tabla

3-3, que en el Grupo 1 solo hay una imagen, aquí la imagen “5010” se incluyó en el Set\_A y Set\_B por igual. Un ejemplo de esto, se puede apreciar en la Figura 3-4.

Categoría/Grupo	Grupo 1	Grupo 2	Grupo 3	Grupo 4	...
<b>Bajo</b>	<u>5010</u>		1333	4652	...
			1603	4656	
			1900	4687	
			2240	4690	
<b>Medio</b>			1604	4607	...
			2035	4680	
			2360	8034	
			2598	8470	
<b>Alto</b>			<u>5760</u>		...
			<u>8190</u>		



IAPS Set_A	IAPS Set_B
<u>5010</u>	<u>5010</u>
1333	1900
1603	2240
1604	2360
2035	2598
<u>5760</u>	<u>8190</u>
4652	4687
4656	4690
4607	8034
...	...

**FIGURA 3-4: SELECCIÓN DE IMÁGENES IAPS PARA LA CREACIÓN DE SET\_A Y SET\_B**

El resultado del procedimiento anterior concluyó en dos conjuntos de datos con 148 imágenes IAPS cada uno. Con esto, se logra evocar la misma cantidad de emoción por imagen con cada set, pero con imágenes diferentes. Entonces, para cada sensor se intercalaron ambos sets de datos.

Las IAPS seleccionadas se integraron en la interfaz de usuario la cual se implementó en Unity. A esta interface también se integró el cuestionario de autoevaluación SAM (Self-Assessment Mannequin). Hay que recordar que el SAM le permite al participante calificar cuál fue la percepción de su experiencia al exponerse a cada IAPS. El objetivo de usar SAM es poder identificar la cantidad de Valencia, Arousal y Dominancia que tuvo el participante al recibir un estímulo (en este caso visual), un ejemplo del cuestionario SAM utilizado se muestra en la Figura 3-5, La primera fila corresponde a la medición de Valencia, la segunda a Arousal y la tercera a Dominancia.

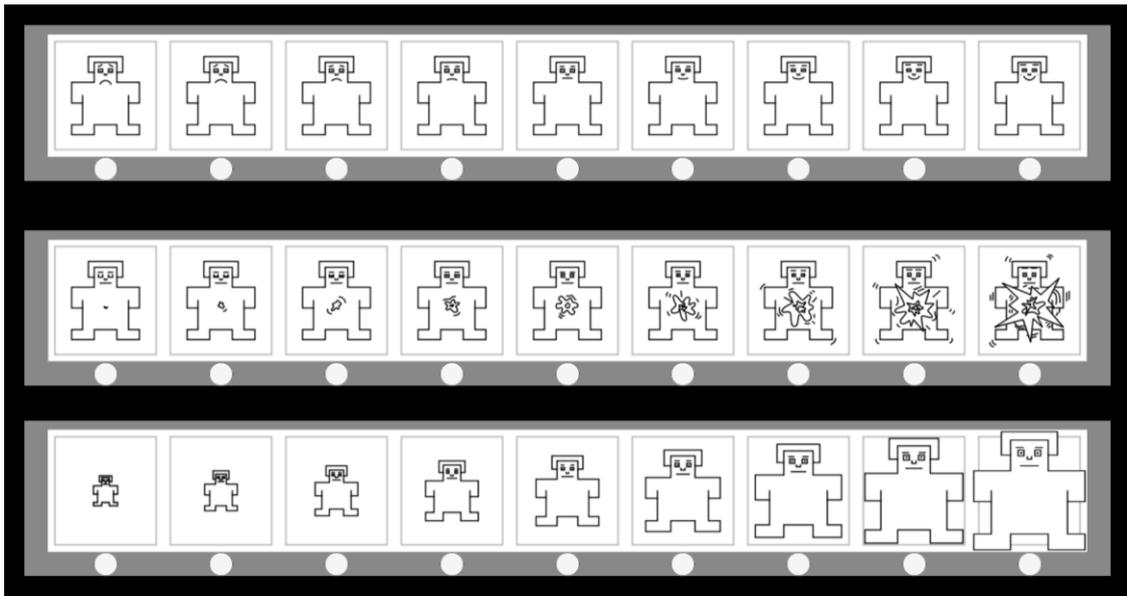


Figura 3-5: Esquema de uso general de la herramienta de auto-evaluación SAM

La técnica IAPS-SAM sugiere que entre transición de estímulos existiera un estado neutral o de espera, por lo cual en la interface se inserta una imagen en blanco. Es decir, a lo largo de la sesión de recolección de datos, se despliega una imagen en blanco, una imagen IAPS y la evaluación SAM, como se muestra en la Figura 3-6.

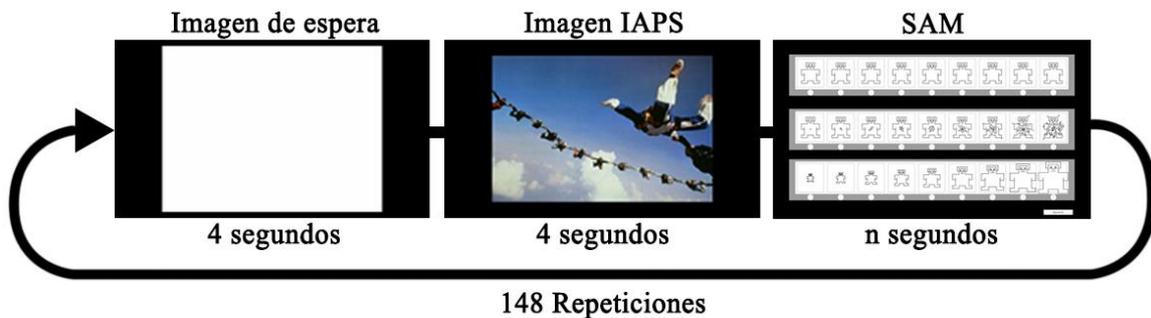


FIGURA 3-6: CICLO DE EVENTOS QUE SE DESARROLLAN EN LA INTERFACE DEL USUARIO

Como se observa en la Figura 3-6, primeramente, se presentó al participante una imagen en blanco por 4 segundos, con el objetivo de lograr un estado afectivo neutro. Posteriormente, se presentó una imagen IAPS por otros 4 segundos para inducir un estado afectivo en el participante. Luego, el participante debe realizar su autoevaluación en donde califica subjetivamente su experiencia previa. Con el objetivo de no sesgar la etapa de

autoevaluación, se dejó que libremente cada participante utilizará el tiempo necesario para completar la misma. Toda vez que se completa la autoevaluación, se reinicia el ciclo para recolectar los datos de otro estado afectivo y se repite hasta completar la recolección de 148 estados afectivos.

Durante cada sesión, fue necesario obtener marcadores de tiempo de los eventos de interés, en este caso, el tiempo de inicio de cada imagen IAPS. Para ello, se generó un archivo de texto en donde se guardó el tiempo del sistema, el número de imagen IAPS, el orden y los valores seleccionados en el cuestionario SAM, ver Tabla 3-4.

**TABLA 3-4: EJEMPLO DE TABLA GENERADA CON VALORES SAM**

<b>Fecha y hora</b>	<b>Diapositiva</b>	<b>Imagen IAPS</b>	<b>Valencia</b>	<b>Arousal</b>	<b>Dominancia</b>
12:48:01.832	1	8206	5	4	9
12:48:19.590	2	6830	2	7	7
12:48:35.961	3	4693	8	6	8
12:48:53.494	4	9145	1	8	6
12:49:09.370	5	4150	9	1	9
12:49:27.704	6	9571	1	7	5
12:49:42.529	7	2393	5	1	9

La primera columna en la Tabla 3-4 muestra el tiempo en el cual se va presentando cada una de las imágenes IAPS. Las columnas “Valencia”, “Arousal” y “Dominancia” muestran el valor que un participante asigna a su experiencia de exposición con la IAPS mostrada. Recordemos que la Valencia nos otorga una percepción entre que tan positiva o que tan negativa es la percepción de la imagen, el Arousal ofrece una medida afectiva para que tanta emoción o excitación una persona experimentó durante el estímulo, y la Dominancia mide la propia sensación de “control” que se tiene sobre la emoción experimentada. [22]

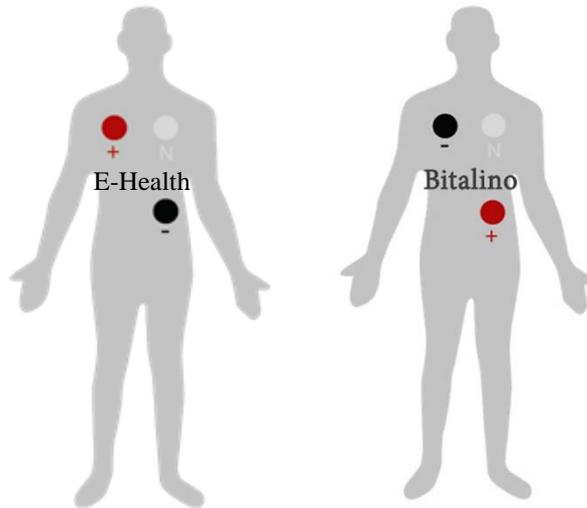
El desarrollo de la interface en Unity y la codificación correspondiente se ofrece en el Apéndice A.

### 3.3. Tecnologías de sensado

Como se observó en la sección de revisión de literatura, dos datos de interés en la detección de emociones son la frecuencia cardiaca, y la respuesta galvánica de la piel (EDA). La primera nos ofrece información con el número de pulsaciones por minuto cuya frecuencia puede cambiar cuando por ejemplo una persona se asusta. La segunda nos ofrece información acerca de las características eléctricas de la piel, las cuales se verán afectadas con la variación de la sudoración del cuerpo humano, cuando, por ejemplo, nos emocionamos. Para la recolección de datos de frecuencia cardiaca y de respuesta galvánica de la piel, proveniente de la inducción de estados afectivos, se utilizaron las tecnologías Bitalino e E-Health. En este apartado se describen algunos aspectos técnicos que deben considerarse para la colocación, recolección y evaluación preliminar de los datos, principalmente porque estamos utilizando dispositivos off-the-shelf.

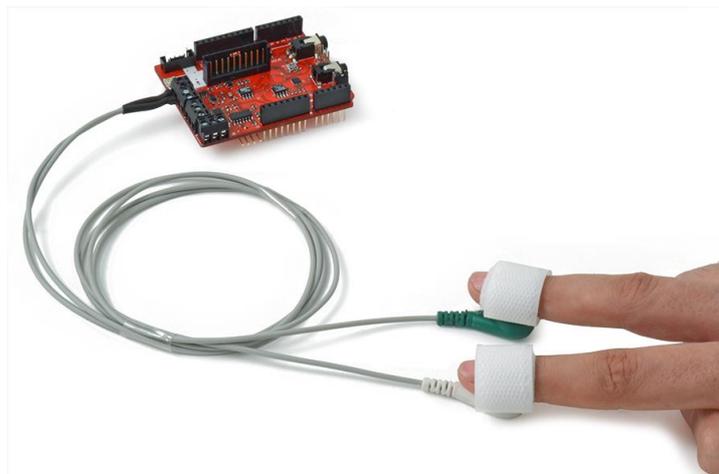
#### 3.3.1. Colocación de electrodos

La electrocardiografía (ECG) es una representación de la actividad eléctrica del corazón en función del tiempo. Tradicionalmente se utiliza un electrocardiógrafo para recolectar datos de la frecuencia cardiaca. Existen diferentes sistemas para realizar los electrocardiogramas y dependiendo del sistema utilizado también puede cambiar la colocación de los electrodos. Tanto para la tecnología Bitalino como para la tecnología e-Health, y específicamente para el sensor que permite la recolección de datos HR, se utilizan tres electrodos, dos de los electrodos sirven para generar un potencial diferencial y un tercer electrodo sirve de referencia para la señal. La Figura 3-7, muestra la colocación del electrodo (+) y del electrodo (-), así como del electrodo de referencia (N). La ubicación de los electrodos es importante ya que su adecuada colocación garantiza la obtención de una buena señal. Por ejemplo, observe en la figura 2 como existe un cambio en la ubicación del electrodo (+) para el Bitalino, este cambio es sugerido por el fabricante debido a que el sistema cuenta con una etapa de amplificación diferencial que internamente invierte la señal.



**FIGURA 3-7: LA COLOCACIÓN DE LOS ELECTRODOS ECG PARA LA TECNOLOGÍA E-HEALTH (IZQUIERDA) Y PARA LA TECNOLOGÍA BITALINO (DERECHA). PARA ESTA ÚLTIMA, OBSERVE LA UBICACIÓN DEL ELECTRODO (+).**

La respuesta galvánica de la piel (GSR), o también conocida como actividad electrodérmica (EDA) y conductancia de la piel (SC), es una representación de las características eléctricas de la piel, producto de la sudoración del cuerpo humano. Para recolectar este tipo de información, tradicionalmente se colocan en el segundo y tercer dedo de la mano (izquierda en este caso) un par de electrodos, por los cuales se hace pasar una corriente eléctrica pequeña. La Figura 3-8, muestra la colocación de los electrodos EDA, la cual fue la misma para ambas tecnologías usadas en este trabajo, aunque Bitalino sugiere que se coloquen los electrodos en la palma de la mano.



**FIGURA 3-8: COLOCACIÓN DE LOS ELECTRODOS EDA EN EL DEDO ÍNDICE Y MEDIO. [17]**

### 3.3.2. Recolección de datos

En relación con la recolección de datos, mientras que Bitalino cuenta con la herramienta OpenSignals [23], para e-Health se programó la placa de Arduino, la cual sirve como host para la placa e-Health, y se utilizó CoolTerm [24] para registrar los datos. El código que se ejecutó sobre la placa de Arduino se ofrece en el Apéndice B (Código Arduino para e-Health). El formato de los datos recolectados con cada una de las tecnologías se muestra en la Tabla 3-5 y Tabla 3-6.

**TABLA 3-5: EXTRACTO DEL REGISTRO DE DATOS GENERADO PARA E-HEALTH.**

<b>Fecha</b>	<b>Hora</b>	<b>EDA (mV)</b>	<b>ECG (V)</b>
2018-09-28	14:08:24	552.297180	1.76
2018-09-28	14:08:24	547.409606	1.68
2018-09-28	14:08:24	547.409606	1.64
2018-09-28	14:08:24	547.409606	1.79
2018-09-28	14:08:24	557.184753	1.74
2018-09-28	14:08:24	547.409606	1.76
2018-09-28	14:08:24	547.409606	1.69

La Tabla 3-5, muestra un extracto de datos EDA y ECG registrados con la tecnología e-Health. Los valores de EDA y ECG se recolectan en volts y milivolts, posteriormente, como la unidad de medida de EDA es en micro-siemens, se realiza esa conversión, explicado más adelante.

Para el caso de la tecnología Bitalino, el software OpenSignals realiza el registro de los datos. Del archivo generado por OpenSignals se removieron las columnas de los demás sensores de Bitalino que no formaban parte de los datos de interés quedando únicamente EDA y ECG, así como se muestra en la Tabla 3-6.

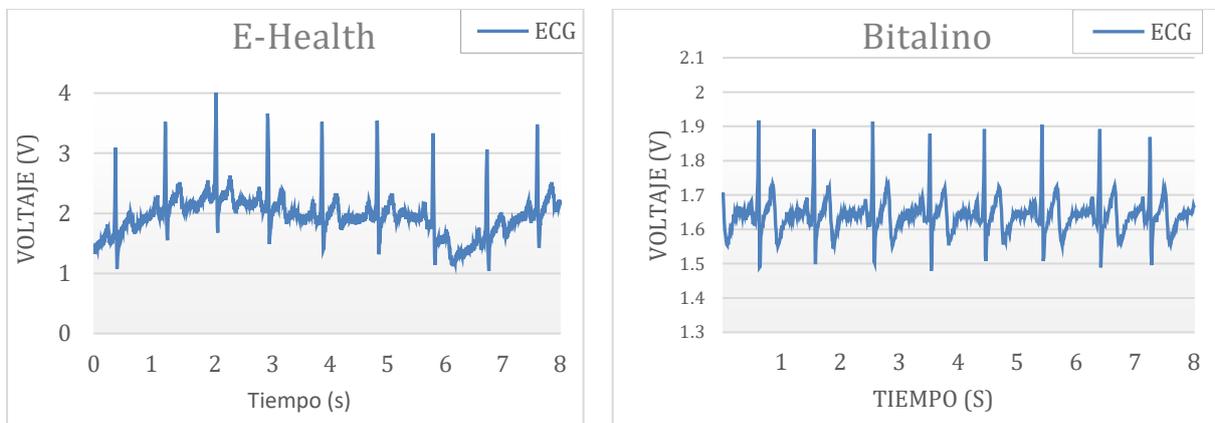
**TABLA 3-6: EXTRACTO DEL REGISTRO DE DATOS GENERADO PARA BITALINO.**

<b>EDA</b>	<b>ECG</b>
419	507
419	511

419	508
419	515
420	511
419	513
419	512

### 3.3.3. Visualización de datos

Es importante revisar que en cada sesión de recolección de datos se visualice la correcta recepción de datos desde los sensores hasta la computadora. La Figura 3-9, muestra las señales ECG tanto para e-Health como para Bitalino.



**FIGURA 3-9: VISUALIZACIÓN DE SEÑALES ECG REGISTRADAS CON LA TECNOLOGÍA E-HEALTH (IZQUIERDA) Y CON LA TECNOLOGÍA BITALINO (DERECHA)**

La Figura 3-10, muestra la visualización de señales para la actividad electrodérmica (EDA) obtenida a través de e-Health y Bitalino.

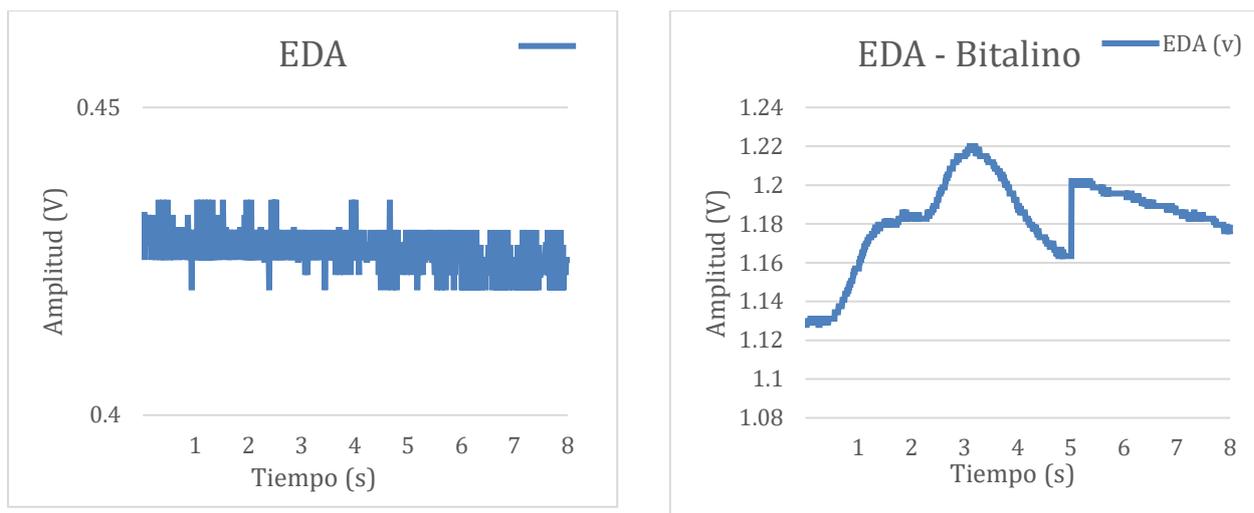


FIGURA 3-10: SEÑAL EDA RECOLECTADA POR E-HEALTH (IZQUIERDA) Y POR BITALINO (DERECHA)

Al final, se obtuvieron 2 archivos por participante, uno generado por la interface de usuario (Tabla 3-4) y otro que contenía las señales EDA y ECG (Tabla 3-5 o Tabla 3-6 dependiendo del sensor). En total, 9 participantes realizaron el experimento usando la tecnología e-Health y 9 la tecnología Bitalino.

### 3.4. Etapa de pre-procesamiento

Una de las tareas que frecuentemente se realiza sobre señales continuas en el tiempo, es el filtrado de la señal. Debido a que las señales del cuerpo humano se encuentran en niveles de mili-volts o micro-volts, estas están sujetas a ruido y por lo tanto este debe separarse de la señal de interés. Estos ruidos pueden tener diferentes orígenes como por ejemplo un mal contacto del adhesivo del electrodo que permita la movilidad de este, más aún, si el paciente no está estático en la posición que se le solicitó. En las siguientes secciones se describe el proceso utilizado para el pre-procesamiento de las señales recolectadas con e-Health y Bitalino, pero antes de eso, se tuvieron que hacer algunos ajustes a las señales adquiridas, como se muestra en la siguiente sección.

#### 3.4.1. Ajustar frecuencia de muestreo y formato

Es importante resaltar, que la frecuencia de muestreo entre ambos dispositivos no fue la misma, ya que en Bitalino se recolectaron datos a 100 Hz, mientras que en e-Health se obtuvo una frecuencia de muestreo variante (aproximadamente a 137 Hz). Para mantener una

misma frecuencia de muestreo entre ambos dispositivos, esta misma, se redujo de los datos de e-Health por medio de un promedio de disminución de frecuencia. Es decir, se calcularon cuántas muestras existían por segundo y se dividió entre 100 para obtener un factor de tiempo, después se obtuvo un promedio de las señales obtenidas entre un tiempo y otro, entonces, con esto se obtuvieron las señales ECG y EDA a 100 Hz de frecuencia, al igual que se tienen en Bitalino. El código del procedimiento anterior se encuentra en el Apéndice C - Downsample. En este mismo procedimiento, se optó por colocar la hora en el mismo formato que se tiene en la Tabla 3-4 (en columna “Fecha y hora”), ya que esta tabla se usará como referencia para saber en qué tiempo exacto se mostró cada imagen IAPS. La Tabla 3-7 muestra el formato en el que quedarían los datos procesados.

**TABLA 3-7: EJEMPLO DE TABLA GENERADA AL REDUCIR LA FRECUENCIA DE MUESTREO DE LA SEÑAL E-HEALTH**

<b>Hora</b>	<b>EDA (mV)</b>	<b>ECG (V)</b>
12:06:20.0	351.90616	1.9
12:06:20.10	349.46234	1.895
12:06:20.20	347.01855	1.91
12:06:20.30	349.46234	1.95
12:06:20.40	349.46234	1.925
12:06:20.50	347.01855	1.95
12:06:20.60	349.46234	1.915

Para el caso de Bitalino, como ya se tenía el tiempo inicial, solo se agregó a la Tabla 3-6 una columna llamada “Hora” y se llenó cada renglón con el tiempo correspondiente a cada registro. Como la frecuencia de muestreo es de 100 Hz, solo se le fue sumando al tiempo inicial 10 milisegundos por registro, un ejemplo de ello se muestra en la Tabla 3-8.

**TABLA 3-8: EJEMPLO DE TABLA GENERADA INCLUYENDO LA HORA DE CADA REGISTRO EN BITALINO**

<b>Hora</b>	<b>EDA (mV)</b>	<b>ECG (V)</b>
12:48:04.0	352	413
12:48:04.10	351	435
12:48:04.20	351	442
12:48:04.30	351	461
12:48:04.40	352	469
12:48:04.50	351	484
12:48:04.60	352	491

### 3.4.2. Pre procesamiento ECG

Cuando el estado afectivo de una persona cambia, también se modifica la frecuencia de bombeo del corazón, por lo tanto, es necesario obtener una medida de este parámetro y asociarlo al cambio en el estado afectivo. La señal ECG recolectada por cada individuo, se separó en ventanas, 148 para ser exactos, en donde cada ventana corresponde a los 4 segundos en que cada imagen IAPS fue mostrada. Para ello, se utilizaron los dos archivos generados por cada participante, en donde de la Tabla 3-4, se hizo un emparejamiento del tiempo en el que apareció cada imagen IAPS y la señal de la Tabla 3-7 o de Tabla 3-8 (dependiendo del sensor utilizado por el participante). Una vez detectado el tiempo en que una imagen IAPS fue mostrada, se seleccionaron 4 segundos de la señal (o 400 datos por estar a una frecuencia de 100 Hz) y se agruparon en un solo archivo por participante. Después, a este archivo por ventanas se le aplicó el algoritmo de la Figura 3-11.

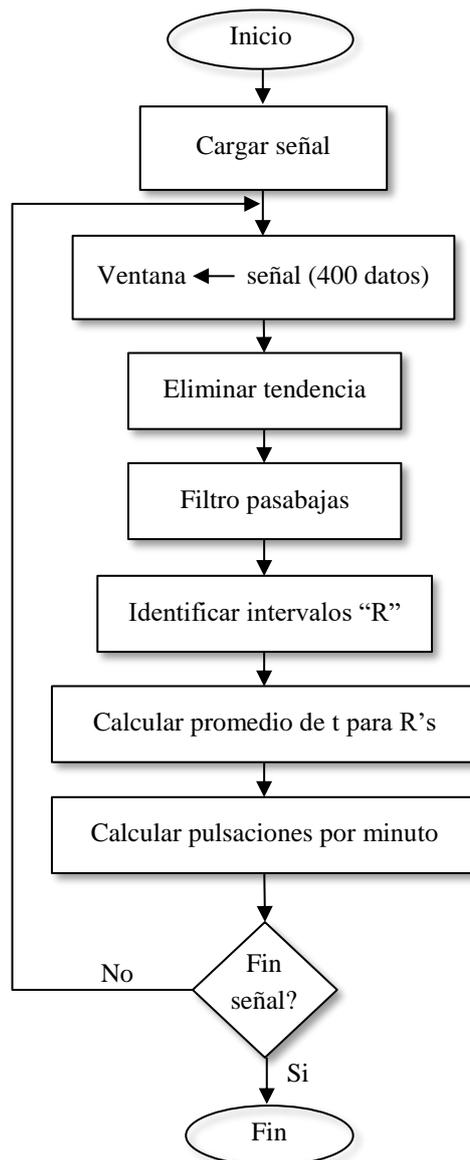
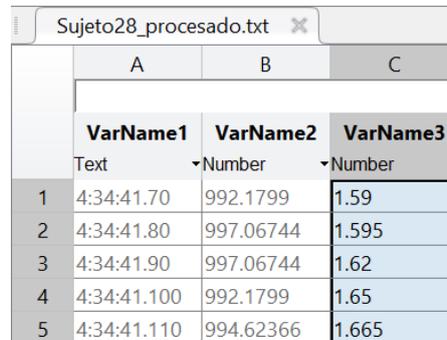


Figura 3-11: Diagrama de flujo para calcular los pulsos por minuto (PPM) de la frecuencia cardíaca de cada ventana

En el algoritmo de la figura anterior, se importó la señal y se realizó un ciclo para analizar cada ventana individualmente. Toda vez que se cuenta con el segmento o ventana de 400 datos, se procesan los mismos para: eliminar tendencia de la señal, quitar ruido de artefactos, identificar eventos  $R_{ECG}$ , y calcular la frecuencia cardíaca. Cada una de estas etapas de procesamiento realizadas sobre la señal ECG se describen en las siguientes subsecciones.

### 3.4.2.1. Importar señal

Para detectar los pulsos o latidos del corazón y su frecuencia por minuto, se hizo uso de Matlab [25]. Para esto, se importaron los datos del archivo por ventanas (extracto de la tabla 3-8), y se tomó la columna correspondiente a los datos de ECG, así como se muestra en la Figura 3-12.

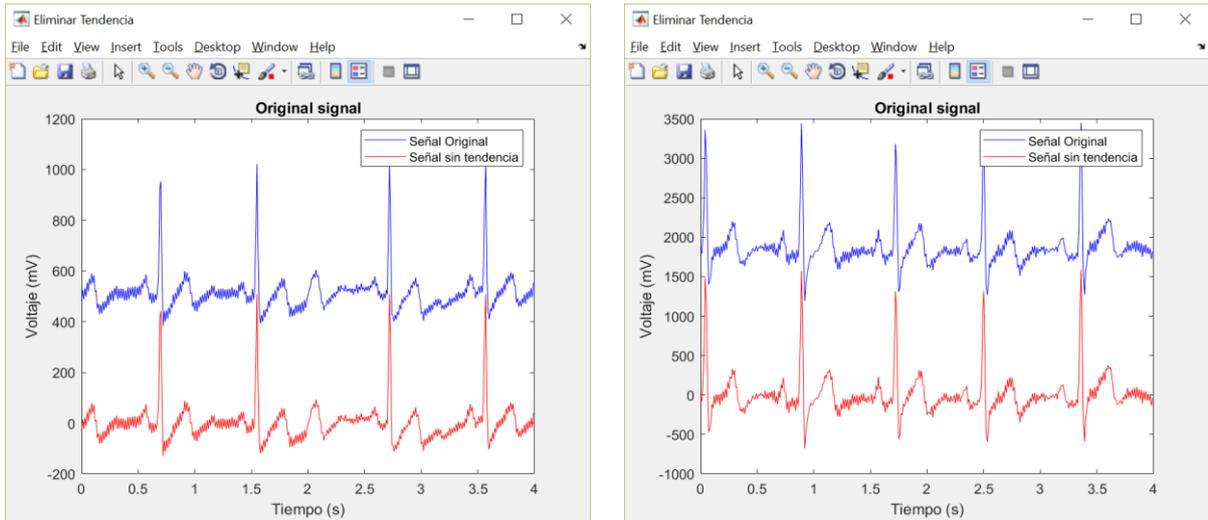


	A	B	C
	VarName1	VarName2	VarName3
	Text	Number	Number
1	4:34:41.70	992.1799	1.59
2	4:34:41.80	997.06744	1.595
3	4:34:41.90	997.06744	1.62
4	4:34:41.100	992.1799	1.65
5	4:34:41.110	994.62366	1.665

FIGURA 3-12: IMPORTANDO DATOS DE ECG

### 3.4.2.2. Eliminar tendencia

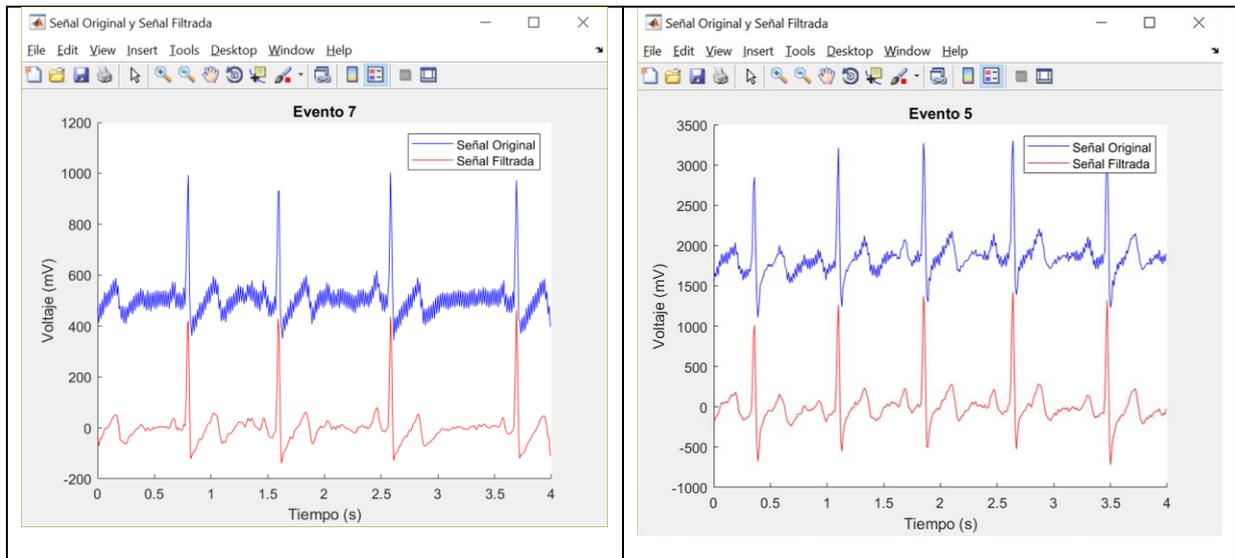
Como se ha mencionado anteriormente, cuando se recolectan señales ECG cualquier movimiento del cuerpo realizado por la persona o incluso con su acción de respiración puede provocar que varíe la impedancia entre la piel y el electrodo, y con ello la aparición de una señal parásita que se suma a la línea base de la señal observándose una tendencia en la misma [Seung-Won Shin et al]. Para la eliminación de la tendencia, en nuestro caso, se hizo uso de la función de Matlab llamada “detrend()”. Esta función resta la media o realiza el mejor ajuste a los datos para remover la tendencia, esto para forzar su media a cero y reducir la variación global. La Figura 3-13, muestra el resultado de aplicar la función “detrend()” de Matlab a la señal ECG. Observe como el resultado de eliminar la tendencia hace que la señal ECG de ambas tecnologías se sujeten al origen.



**FIGURA 3-13: ELIMINACIÓN DE TENDENCIA EN SEÑAL ECG. E-HEALTH (IZQUIERDA), BITALINO (DERECHA)**

### 3.4.2.3. Filtrado de ruido

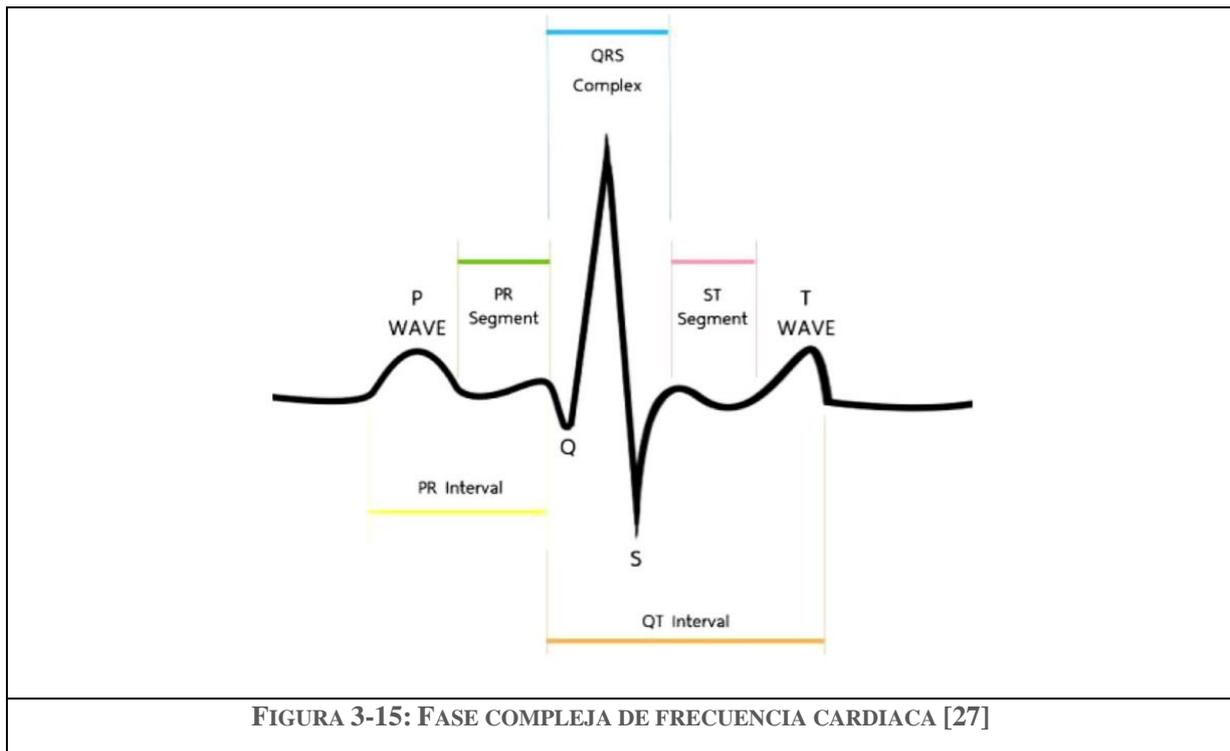
El siguiente paso es la eliminación de ruido, por lo que se hace uso de un filtro pasabajas de 30 Hz en MATLAB, esto para filtrar la señal de ECG, y remover el ruido por artefactos. La Figura 3-14, muestra la aplicación del filtro a la señal ECG. Puede observarse que se ha utilizado la señal original de forma ilustrativa para mostrar el resultado del filtrado, sin embargo, como se planteó en el algoritmo el filtro pasa-bajas se aplica sobre la señal a la cual se le quitó la tendencia. [26]



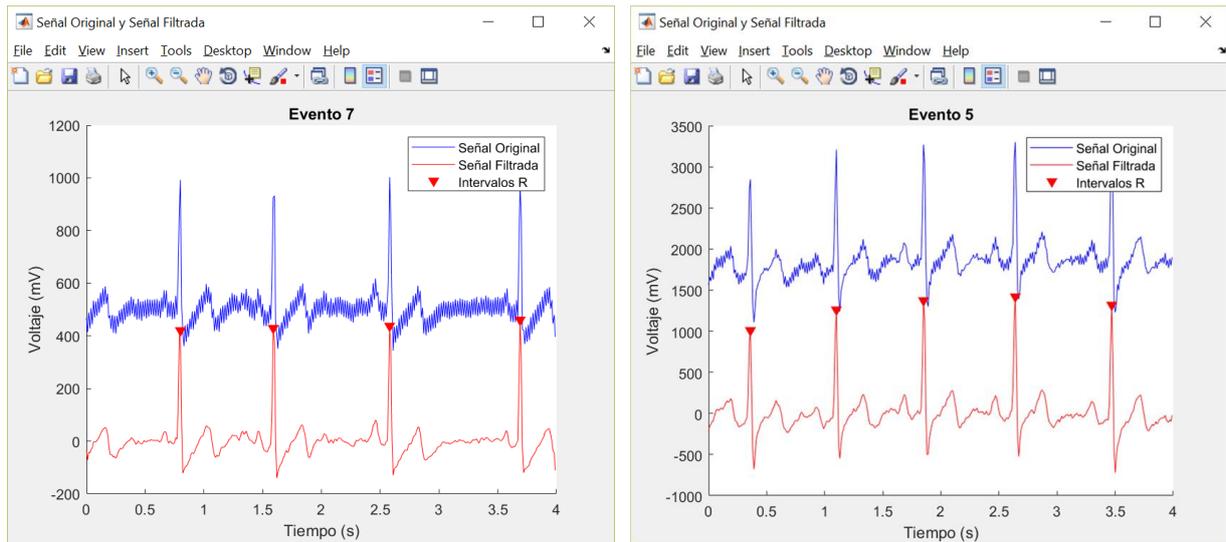
**FIGURA 3-14: SEÑALES ECG DE SALIDA PARA UN FILTRO PASABAJAS. IZQUIERDA E-HEALTH, DERECHA BITALINO. EN CADA IMAGEN EN LA PARTE SUPERIOR SE ENCUENTRA LA SEÑAL ORIGINAL Y EL PARTE INFERIOR LA SEÑAL FILTRADA.**

#### 3.4.2.4. Identificación de pulsos por minuto (PPM)

Como se indicó anteriormente, previo a la determinación de la frecuencia cardiaca se deben procesar los datos para identificar los eventos  $R_{ECG}$  relevantes en la señal. Para ello se parte de observar que la señal del corazón se descompone en 3 fases (ilustradas en la Figura 3-15): la onda P, el conjunto QRS, y la onda T. La sección de señal de interés es el conjunto QRS y en particular, la determinación de los intervalos  $R_{ECG}$ , para a partir de esta información obtener un valor de frecuencia cardiaca en pulsos por minuto (PPM). [27]



Para identificar los eventos R, se utilizó el método “findpeaks” [28] de Matlab. Este método permite establecer umbrales y utilizar ese valor como referencia para marcar los picos que sobresalen en la señal, como se muestra en la Figura 3-16.

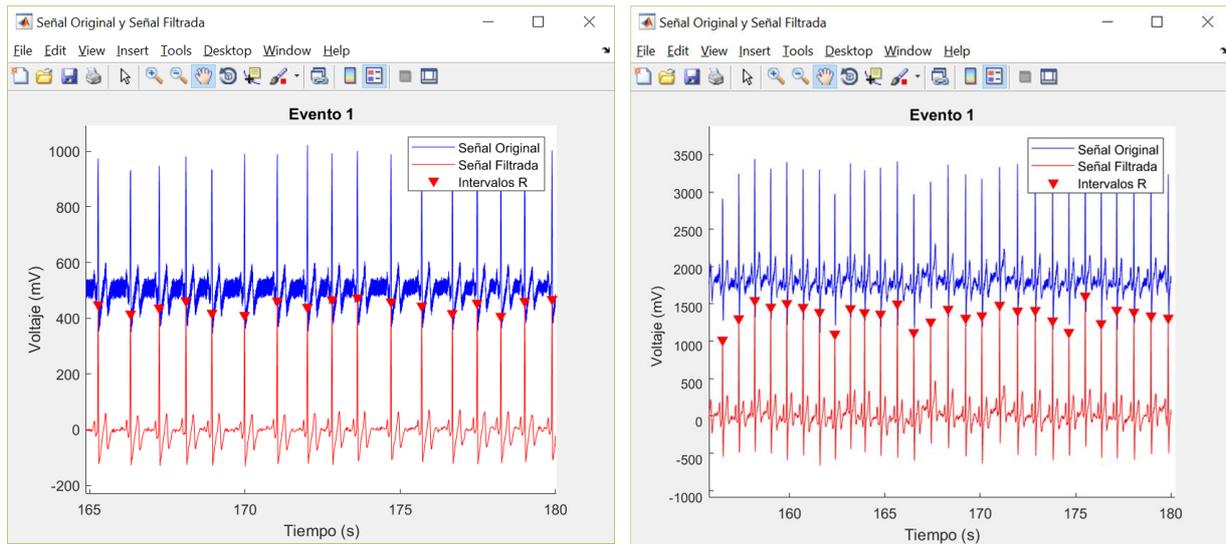


**FIGURA 3-16: IDENTIFICACIÓN DE EVENTOS R EN LA SEÑAL ECG REGISTRADA CON E-HEALTH (IZQUIERDA) Y BITALINO (DERECHA). EL EVENTO R SE MARCA CON UN TRIÁNGULO INVERTIDO SOBRE LA SEÑAL FILTRADA.**

Determinado el número de eventos R, se promedian los intervalos de tiempo entre cada evento, y finalmente se obtiene el número de pulsaciones por minuto utilizando un factor de 1/60 (segundos). Por ejemplo, en la Figura 3-16 (izquierda), al hacer este procedimiento se detectó un promedio de 77.1704 ppm.

#### 3.4.2.5. Identificación de PPM para la Línea base

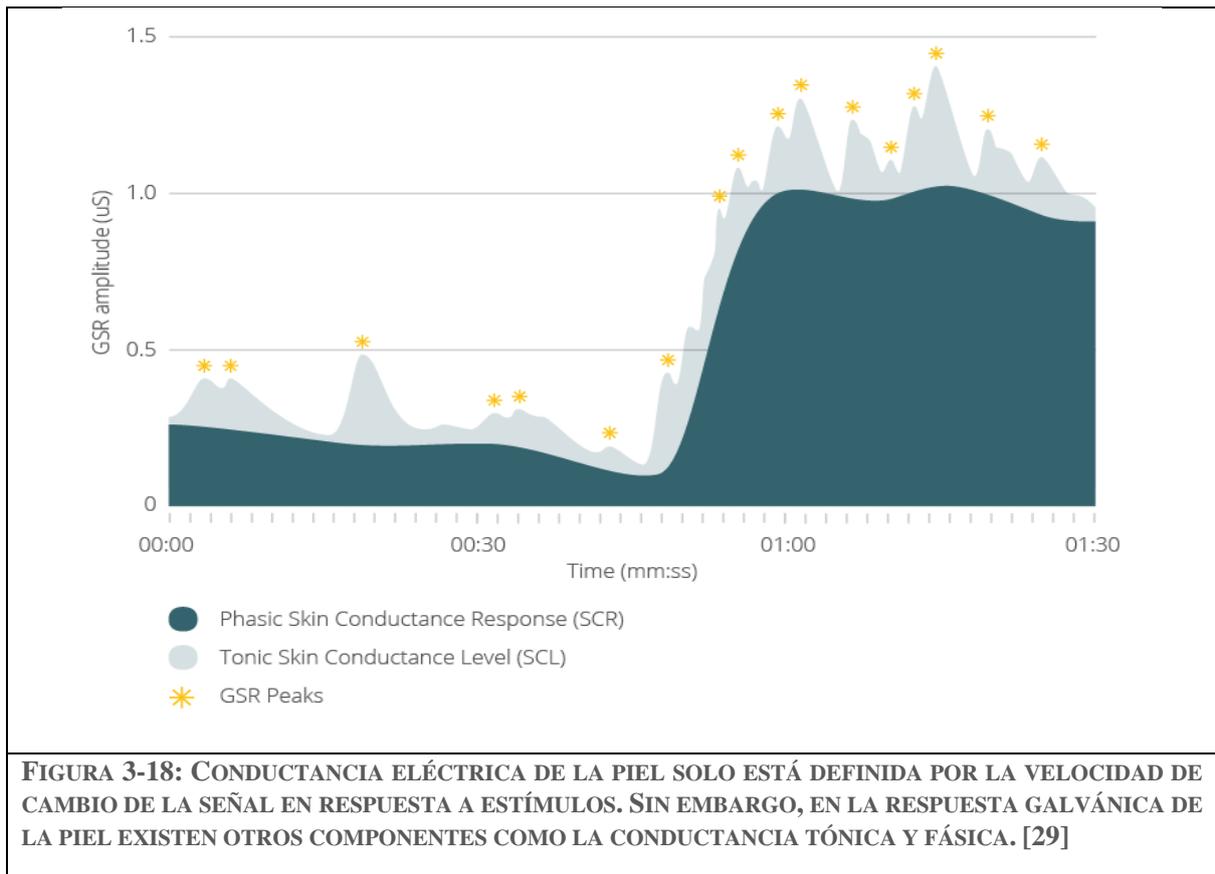
Para el análisis de la línea base se realizó el mismo procedimiento, con la excepción de que la ventana de datos se amplió a 3 minutos o 18,000 datos. Recuerde que en la fase de recolección de datos se recolectan de 3 minutos de señal y que el participante permanece en estado de reposo y con los ojos cerrados, la ventana comprende dicho periodo de tiempo. A continuación, en la Figura 3-17, se pueden observar algunas de las señales clasificadas de ambos dispositivos, mostrando solo un segmento de los 3 minutos recolectados.



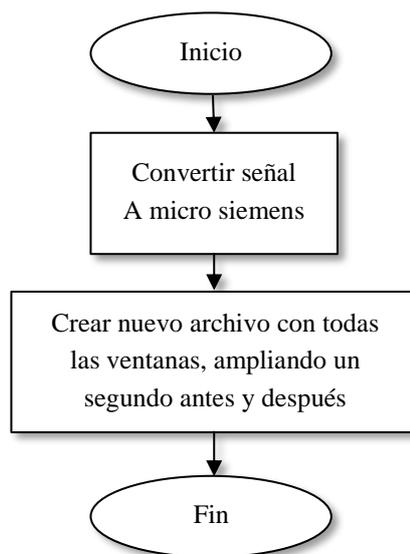
**FIGURA 3-17: IDENTIFICACIÓN DE INTERVALOS R PARA LA LÍNEA BASE, E-HEALTH (IZQUIERDA) Y BITALINO (DERECHA)**

### 3.4.3. Pre procesamiento EDA

Mientras que la frecuencia cardiaca es un evento que mantiene un rango de periodicidad en su comportamiento, la señal correspondiente al nivel de conducción de la electricidad a nivel piel no lo es. Ya se ha comentado que el nivel de respuesta de conducción de la piel depende de glándulas sudoríparas que se activan, y esta activación se puede dar por diferentes factores: carga cognitiva, ansiedad, regulación térmica del cuerpo humano, ejercicio, nerviosismo o algún otro componente afectivo/emocional [29]. Como se observa en la Figura 3-18.



Así entonces, como se verá en este apartado, el pre-procesamiento de la señal EDA requiere de algunas consideraciones particulares, como se muestra en la Figura 3-19, existen dos etapas de pre-procesamiento: las cuales se describen a continuación.



**FIGURA 3-19: PRE-PROCESAMIENTO DE LA SEÑAL EDA**

Como se indica en la Figura 3-19, antes de generar ventanas EDA, se debe convertir la señal de milivolts (mV) a micro siemens ( $\mu\text{S}$ ), tanto para los datos de la tecnología e-Health como para los de Bitalino. Afortunadamente, los mismos fabricantes ofrecen el factor de conversión a aplicar para cambiar de niveles de voltaje para la conductancia de la piel a solamente conductancia ( $\mu\text{S}$ ), ver ecuación 2: [30]

#### ECUACIÓN 1: FORMULA PARA CONVERTIR EDA A MICRO SIEMENS

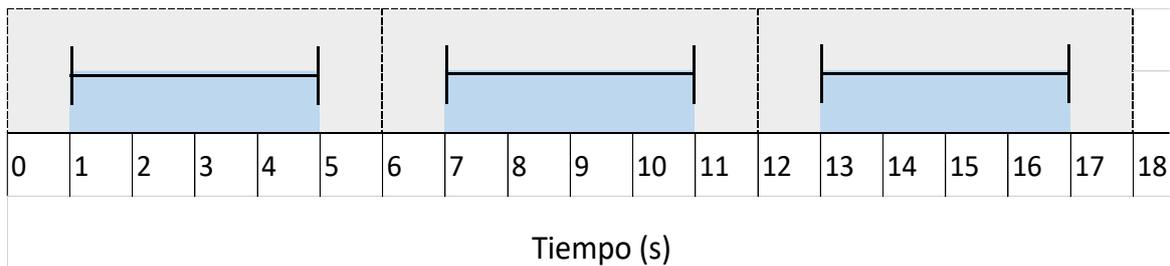
$$EDA(\mu\text{S}) = \frac{\frac{ADC}{2^n} \times VCC}{0.132}$$

En esta ecuación “VCC” equivale a 3.3 voltios, “N” es el número de bits del canal y “ADC” es el valor dado por el sensor.

Toda vez que se cuenta con los datos en valores de micro siemens, se procede a la generación de ventanas, esto tomando como referencia los momentos de inducción de estados afectivos.

##### 3.4.3.1. Asociación de eventos EDA-IAPS

Como se explicó al inicio de esta sección, la señal EDA carece de periodicidad, por lo que la detección del nivel de respuesta galvánica de la piel debe hacerse identificando la magnitud del estado fásico de la conductancia. Para preparar el archivo antes de hacer el análisis, se obtuvieron ventanas, cada una corresponde al tiempo (4 segundos) en que el estímulo o imagen IAPS fue mostrado. Para hacer esto, se hizo uso de la señal EDA y se tomaron los tiempos en que las imágenes IAPS fueron mostradas de la tabla SAM (Tabla 3-4). Además, se amplió dicha ventana en un segundo, 1 segundo al inicio, y otro segundo al final de cada ventana. En la Figura 3-20 se puede apreciar que en azul se expresan las ventanas de 4 segundos, y en gris los segundos agregados al ampliar la ventana. Por este motivo, se creó una nueva tabla para saber en qué segundo empieza la ventana de datos original, en este caso, la primera ventana inicia en el segundo 1 y termina en el segundo 5, después, el evento dos, inicia situado en el segundo 7 y termina en el 11 y así sucesivamente.



**FIGURA 3-20: EJEMPLO DE CÓMO SE TOMARON VENTANAS DE EVENTOS PARA IMPORTAR A LEDALAB**

Basándose en la figura anterior, se pueden apreciar las ventanas y sus posiciones en el tiempo, por lo que el archivo que se generó para gestionar estos tiempos, se expresa a continuación en la Figura 3-21. La columna “Tiempo (s)”, es el segundo en el que inicia el estímulo, la columna “IAPS” es el número de imagen que fue mostrado en esa ventana.



Figura 3-21: Ejemplo de Archivo SC\_Marker\_participante(n)

### 3.4.3.2. Pre procesamiento de la señal

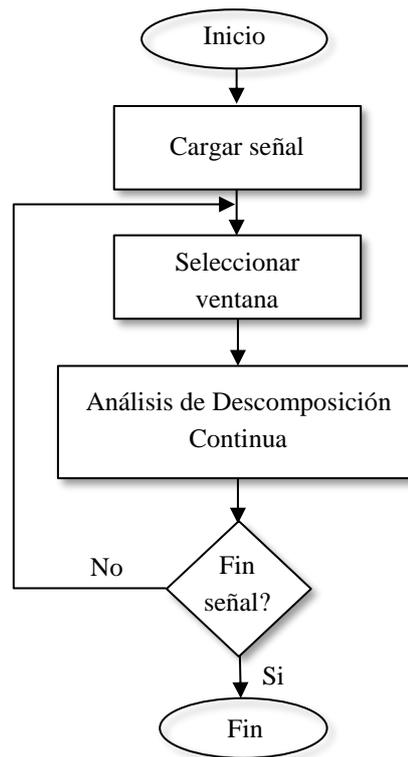
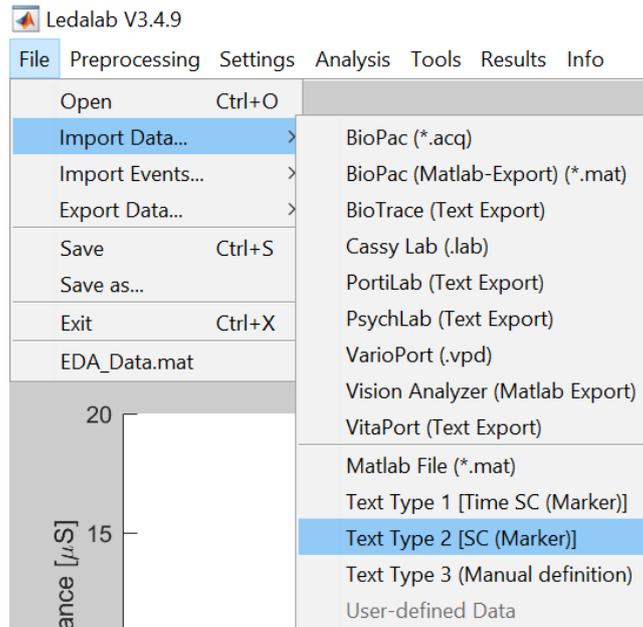


FIGURA 3-22: PRE-PROCESAMIENTO DE LA SEÑAL EDA

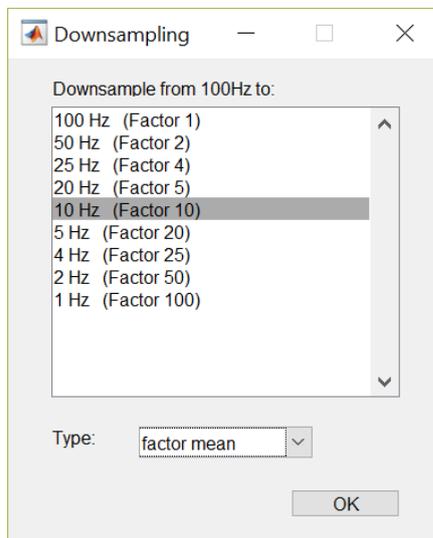
Para realizar el procedimiento de la Figura 3-22, se hizo uso de un programa validado para el análisis de la señal EDA, el cuál es Ledalab. Esta herramienta está desarrollada en Matlab y permite aplicar el análisis de descomposición continua (ADC), mediante el cual es posible obtener valores de actividad fásica y tónica. [31]

Una vez iniciado Ledalab se procedió a importar los archivos generados anteriormente en las Figura 3-20 y Figura 3-21. Para este primer archivo, se seleccionó el formato del archivo, en este caso: File/Import Data.../Text Type 2 [SC (Marker)], mostrado en la Figura 3-23.



**FIGURA 3-23: IMPORTACIÓN DE DATOS EDA A LEDALAB**

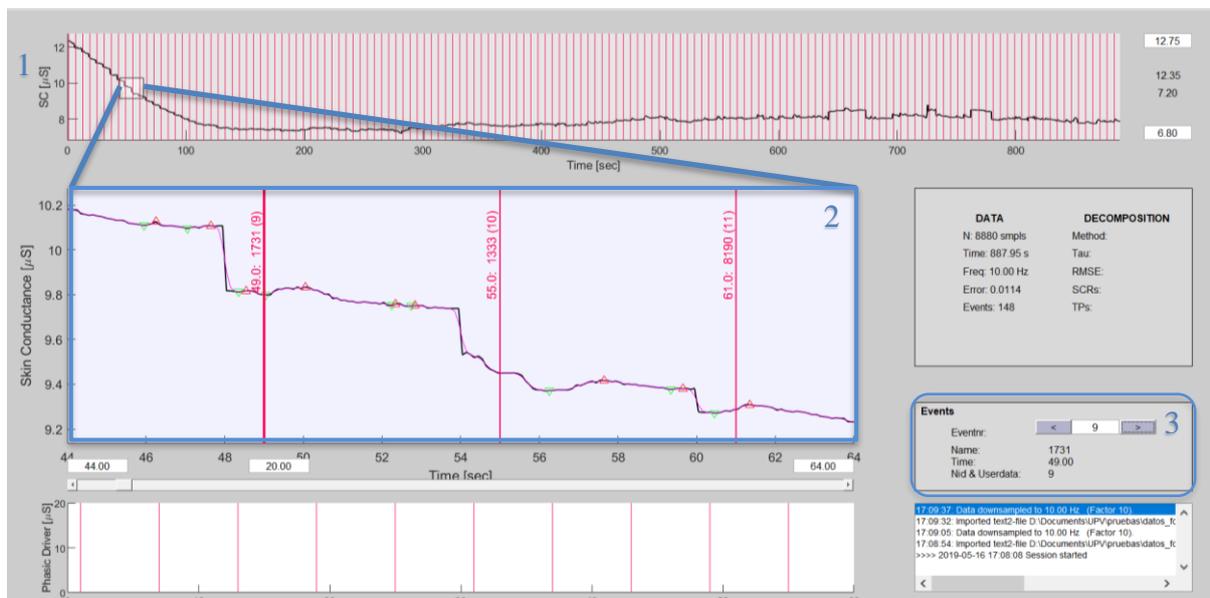
Al hacer esto, aparece una ventana en donde pide ingresar la frecuencia de muestreo en la que están los datos, en este caso son 100. Luego, el programa sugiere reducir la frecuencia de muestreo para agilizar el análisis, para nuestro caso se seleccionó una reducción de frecuencia de muestreo de 100 Hz a 10 Hz, por un factor de promedio, tal y como se ve en la Figura 3-24.



**Figura 3-24: Downsampling a 10 Hz**

A continuación, se importaron los datos EDA-IAPS (mostrado en la Figura 3-21) siguiendo la siguiente ruta: “File/Import Events.../Load New Event Markers (and DELETE

existing ones)”. Estos datos sirven como marcadores de eventos EDA. Tras haber importado estos marcadores, se puede navegar entre eventos utilizando las funciones de navegación provistas por Ledalab, en la Figura 3-25 el área de navegación se ha etiquetado con un número “3”. Ledalab también cuenta con un área de visualización, etiquetada con el número “1”, en la cual se observa la señal completa importada. El área Ledalab etiquetada con un “2”, corresponde a una vista ampliada de un segmento de la señal completa. En cualquier caso, cada línea roja que aparece corresponde al inicio de un único evento EDA.



**FIGURA 3-25: EJEMPLO IMPORTANDO DATOS EDA**

### 3.4.3.3. Análisis de descomposición Continua (ADC)

Para hacer el análisis de la señal, se utilizó el Análisis de Descomposición Continua [31], seleccionando “Analysis/Continuous Decomposition Analysis (extraction of Continuous Phasic/Tonic Activity)”. Con los valores establecidos por default, se inició el proceso de optimización en el botón “Optimize”, esto para reducir el error dentro de la reconstrucción de los datos de conductancia de la piel. El procedimiento es repetir constantemente la optimización hasta minimizar el valor de error del análisis de descomposición continua (ADC). La Figura 3-26 muestra el error inicial de la señal, cuyo valor es de 22.013, mientras que la Figura 3-27 muestra el error resultante, cuyo valor es de 0.399, después de aplicar el análisis ADC en tres ocasiones.

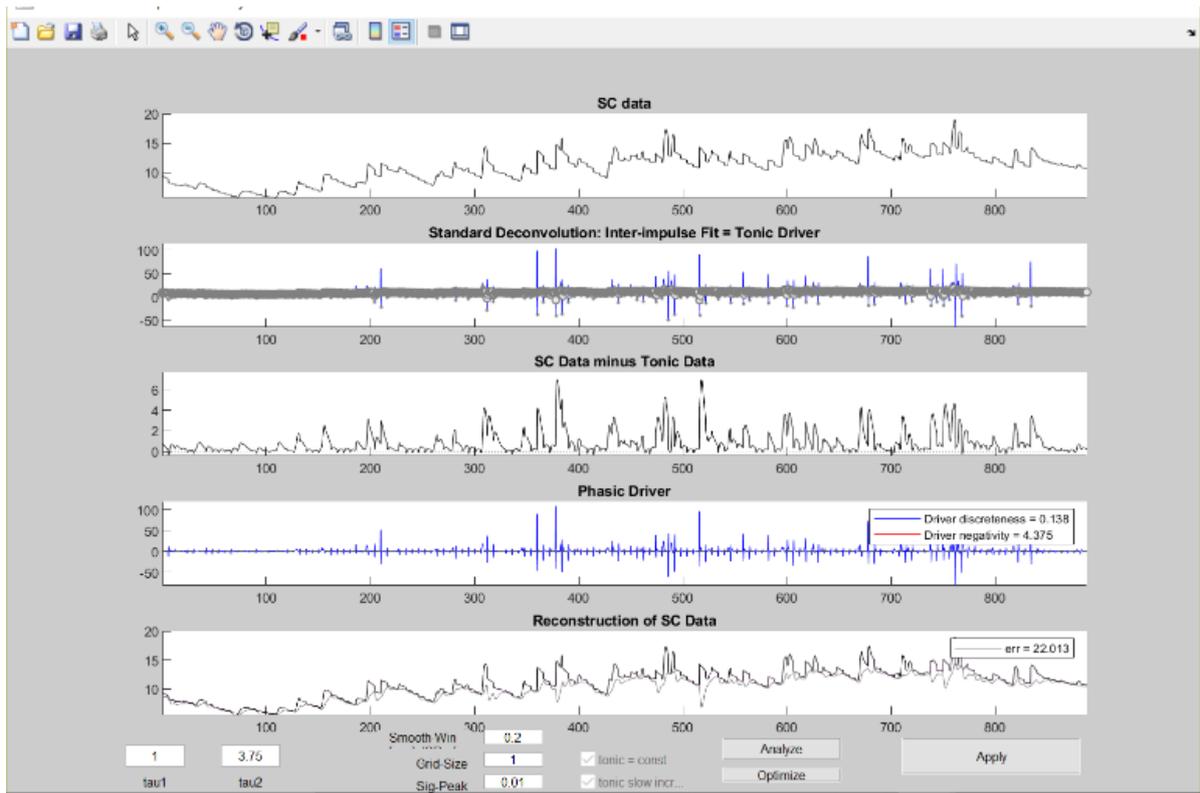


Figura 3-26: Señal inicial en el ADC, con un error de 22.013.

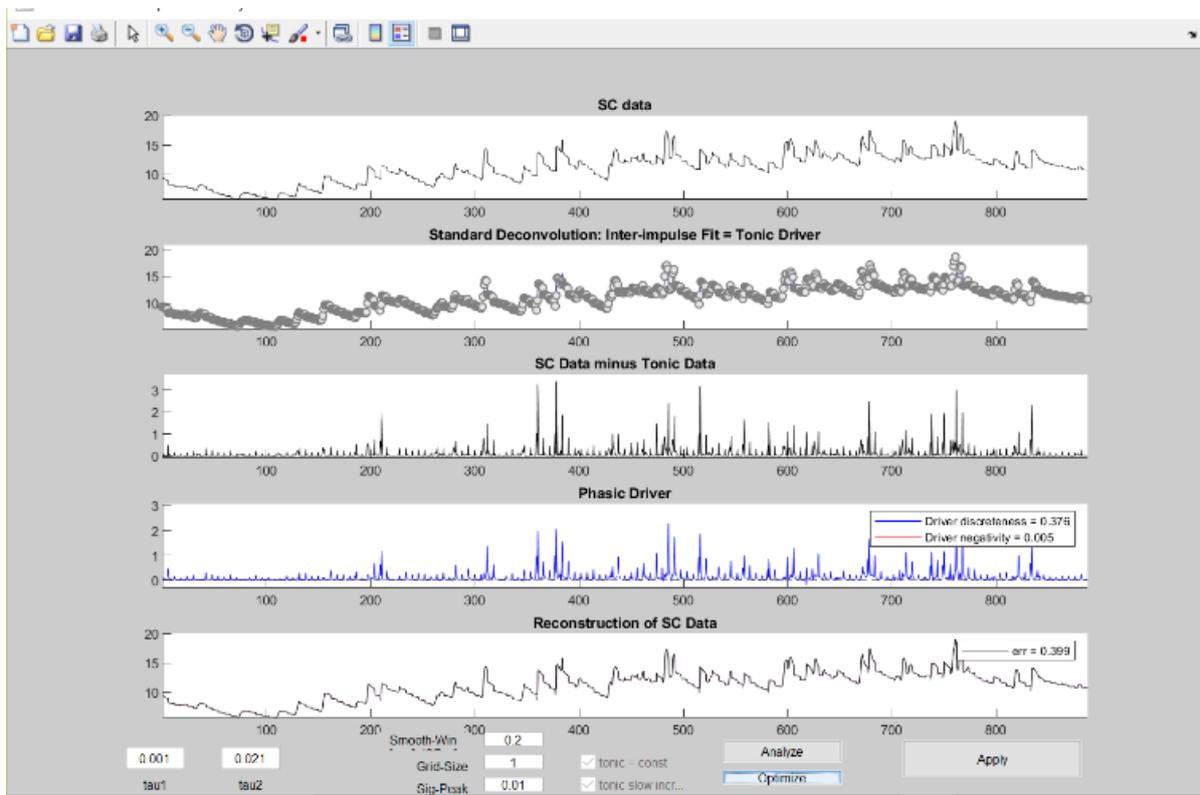


Figura 3-27: Tercera optimización en el ADC, con un error de 0.399

Finalizado el proceso de minimización del error, se selecciona el botón “Apply” para guardar la configuración y los resultados. Después de aplicar el análisis ADC, se pueden visualizar las ondas EDA de cada evento, y algunas características que definen su comportamiento: retraso en aparecer, geometría de la señal, amplitud, entre otras. Un ejemplo de ello se muestra a continuación en la Figura 3-28.

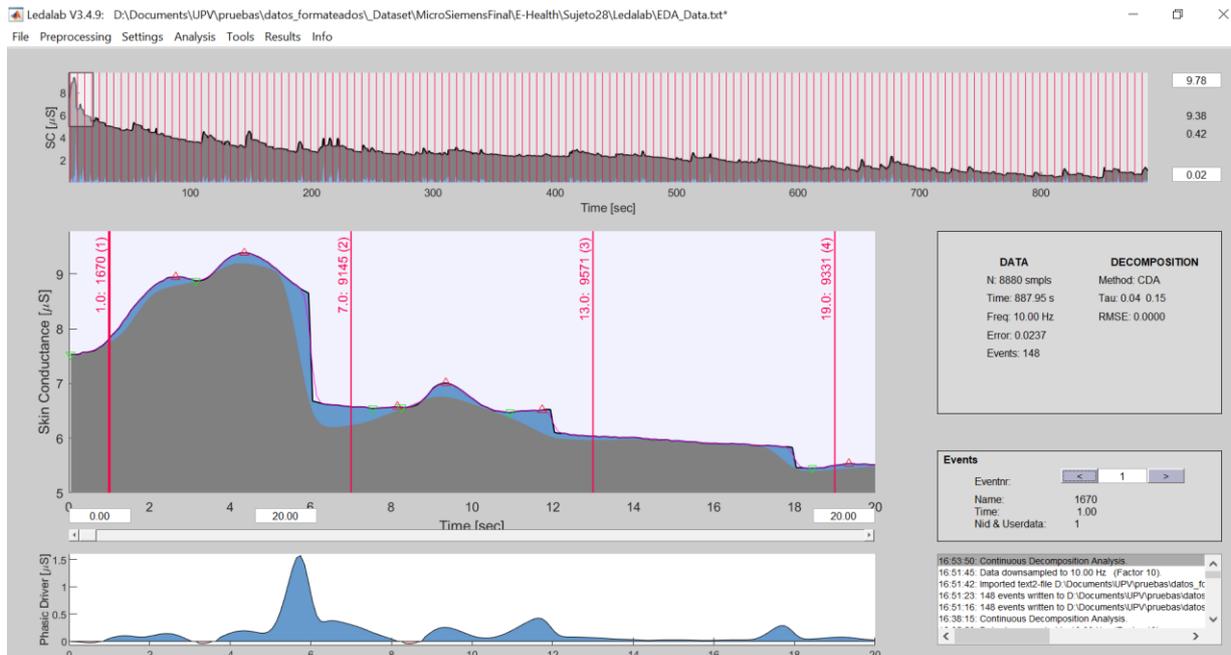


FIGURA 3-28: DESPUÉS DEL ANÁLISIS ADC - EJEMPLO

En este punto se obtuvieron las características EDA de cada evento, y para exportarlas fue necesario ir a: “Results/Export Event-Related Activation Ctrl+E”. Aquí se seleccionó el inicio y fin de la ventana relativa al evento y la amplitud, como se menciona en [32], se utilizó una ventana de 1 a 4 segundos, puesto que el primer segundo puede considerarse como ruido de estímulos anteriores, además de que es una de las configuraciones más utilizadas. Seguido de esto, se seleccionó una amplitud mínima de 0.01  $\mu$ S. Esto se puede observar en la Figura 3-29.

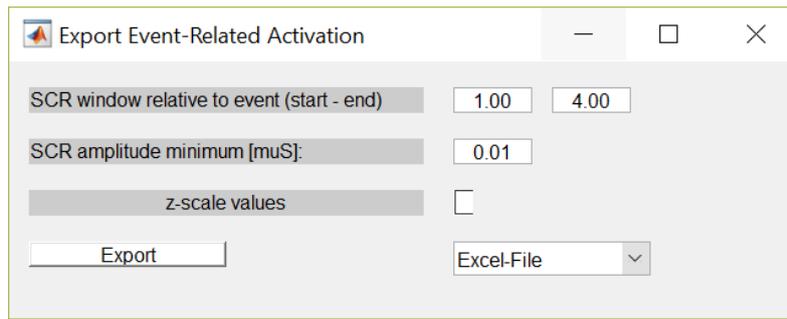


FIGURA 3-29: EXPORTANDO RESULTADOS EDA

### 3.5. Creación de set de datos

En las secciones anteriores se mostró el pre-procesamiento de las señales ECG y EDA con el objetivo de generar el set de datos con pulsaciones por minuto (PPM) y los eventos de respuesta galvánica (GSR) asociados a los eventos de inducción de estados afectivos. La siguiente actividad requiere preparar el set de datos que será utilizado para la etapa de clasificación. El diagrama de flujo de la figura 3-18, muestra los pasos que se desarrollaron para la generación del set de datos.

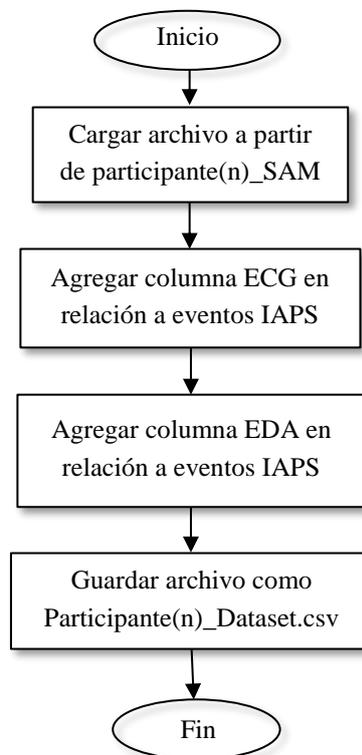


FIGURA 3-30: ACTIVIDADES DESARROLLADAS PARA PREPARAR EL SET DE DATOS A UTILIZAR EN LA ETAPA DE CLASIFICACIÓN.

Como lo muestra la Figura 3-30, para la primera etapa de preparación del dataset se toma como archivo base aquel que cuenta con la evaluación SAM de cada participante (extracto mostrado en la Tabla 3-4). En particular se utilizan las columnas “IAPS”, “Valencia” y “Arousal”, como se ve en la Figura 3-31.

Fecha y hora	Diapositiva	Imagen IAPS	Valencia	Arousal	Dominancia
12:48:01.832	1	8206	5	4	9
12:48:19.590	2	6830	2	7	7
12:48:35.961	3	4693	8	6	8
12:48:53.494	4	9145	1	8	6
12:49:09.370	5	4150	9	1	9
12:49:27.704	6	9571	1	7	5
12:49:42.529	7	2393	5	1	9



Imagen IAPS	Valencia	Arousal
8206	5	4
6830	2	7
4693	8	6
9145	1	8
4150	9	1
9571	1	7
2393	5	1

**FIGURA 3-31: CREACIÓN DE ARCHIVO DE SET DE DATOS DESDE TABLA 3-4**

A este nuevo archivo se le agregaron los datos obtenidos en la sección de procesamiento de ECG y EDA. La Tabla 3-9, muestra un extracto de la información del archivo resultante en el cual se integraron las columnas de interés. Este procedimiento se realiza tomando como referencia las imágenes IAPS y para cada conjunto de datos de cada uno de los participantes, es decir se cuenta con 18 archivos con esta configuración de información.

**TABLA 3-9: EJEMPLO DE TABLA DE SET DE DATOS POR PARTICIPANTE**

ECG (bpm)	CDA.nSCR	CDA.Latency [s]	CDA.AmpSum [muS]	CDA.SCR [muS]	CDA.ISCR [muSxs]	CDA.PhasicMax [muS]	CDA.Tonic [muS]	Valence	Arousal
95.5414013	2	1.945	0.10170768	0.00413371	0.12401144	0.15234376	12.2246376	7	7
84.7457627	1	1.745	0.03546003	0.00318213	0.0954638	0.10209613	11.8815156	1	9
82.6446281	1	2.045	0.0135642	0.00237689	0.07130682	0.07483134	11.6351431	9	4
80.8080808	1	2.045	0.02504344	0.00258644	0.07759315	0.0712776	11.2784894	9	6
80.8625337	2	2.345	0.04233115	0.00258033	0.07740989	0.07590431	11.0473771	1	9
78.4313726	1	2.945	0.02635589	0.00222104	0.06663111	0.07521931	10.4015538	7	3
80.4289544	1	3.045	0.02086085	0.00197313	0.05919404	0.08611994	10.1069747	1	9
82.6446281	1	2.545	0.01603776	0.00270673	0.08120193	0.10588879	9.7433933	5	2

A partir de este tipo de archivos, el proceso siguiente se encarga de normalizar los datos y finalmente generar los conjuntos de datos a usar en la sección de clasificación.

### 3.5.1. Normalización y estandarización

#### 3.5.1.1. Normalización de ECG

Para reducir las diferencias entre usuarios, se utilizó la fórmula descrita en [33], ya que como se menciona, es necesario aplicar una normalización a los datos de la columna “ECG (ppm)”. Esto, debido a que las señales pueden variar dependiendo de la edad, sexo, entre otros factores de cada persona. Para hacer esta normalización se obtiene el número de pulsaciones por minuto de la línea base ( $\bar{X}_{base}$ ), y los valores de ECG de cada evento ( $ECG_{data}$ ). Por lo que la fórmula se puede apreciar en la Ecuación 2.

**ECUACIÓN 2: FORMULA DE NORMALIZACIÓN ECG**

$$ECG_{norm} = \frac{ECG_{data} - \bar{X}_{base}}{\bar{X}_{base}}$$

Aplicando esta fórmula al archivo de la Tabla 3-9, se creó una nueva columna, la cual fue nombrada “ECG norm”. En la Tabla 3-10 se muestra un ejemplo de cómo se visualizarían los datos. Después de esto, la columna “ECG (ppm)” se eliminó de la tabla ya que solo hace falta utilizar la columna de la señal ECG normalizada.

**TABLA 3-10: EJEMPLO DE SEÑAL ECG NORMALIZADA**

ECG (ppm)	ECG norm
78.1758958	0.03898661
74.0740741	-0.01552812
75.4716981	0.00304682
75.2351097	-0.00009753

#### 3.5.1.2. Estandarización EDA (Z-score)

Como se muestra en [34], para mejorar la estandarización de los datos se utiliza Z-score. Para utilizar esta métrica, se necesita calcular la media y la desviación estándar de los

datos de EDA, en la Ecuación 3, se muestra cómo se calcula el valor Z-score, siendo “ $\bar{X}$ ” el promedio y “s” la desviación estándar de los datos EDA.

### ECUACIÓN 3: Z-SCORE EDA

$$EDA_{Z\text{-score}} = \frac{EDA_{fila} - \bar{X}}{s}$$

Aplicando la ecuación anterior a los datos obtenidos por participante en la parte de pre-procesamiento EDA (mostrado en la Tabla 3-9), se sustituirán estos valores por los normalizados con Z-score, dando como resultado la Tabla 3-11 (esta nueva tabla incluye los valores normalizados de ECG).

**TABLA 3-11: EJEMPLO DE DATOS NORMALIZADOS EDA**

ECG norm	CDA.nSCR	CDA.Latency [s]	CDA.AmpSum [muS]	CDA.SCR [muS]	CDA.ISCR [muSxs]	CDA.PhasicMax [muS]	CDA.Tonic [muS]	Valence	Arousal
0.1045093	0.49344974	-0.3901254	1.83845458	1.54414665	1.54414665	1.80879181	4.84600686	7	7
0.23119137	-0.5351497	-0.6409909	-0.1161432	0.8327273	0.8327273	0.8191213	4.44671986	1	9
0.087032887	-0.5351497	-0.2646927	-0.7621669	0.23072505	0.23072505	0.28211752	4.160019	9	4
0.03464576	-0.5351497	-0.2646927	-0.4234786	0.38738317	0.38738317	0.21212353	3.74498526	9	6
0.100969206	0.49344974	0.1116055	0.08658488	0.38281611	0.38281611	0.30325061	3.4760426	1	9
0.080196202	-0.5351497	0.86420184	-0.3847556	0.11420455	0.11420455	0.28975909	2.72450559	7	3
0.152692592	-0.5351497	0.98963457	-0.5468837	-0.0711304	-0.0711304	0.50445644	2.38170737	1	9
0.141203961	-0.5351497	0.36247095	-0.689186	0.47731535	0.47731535	0.89382108	1.95861193	5	2

### 3.5.2. Combinar archivos

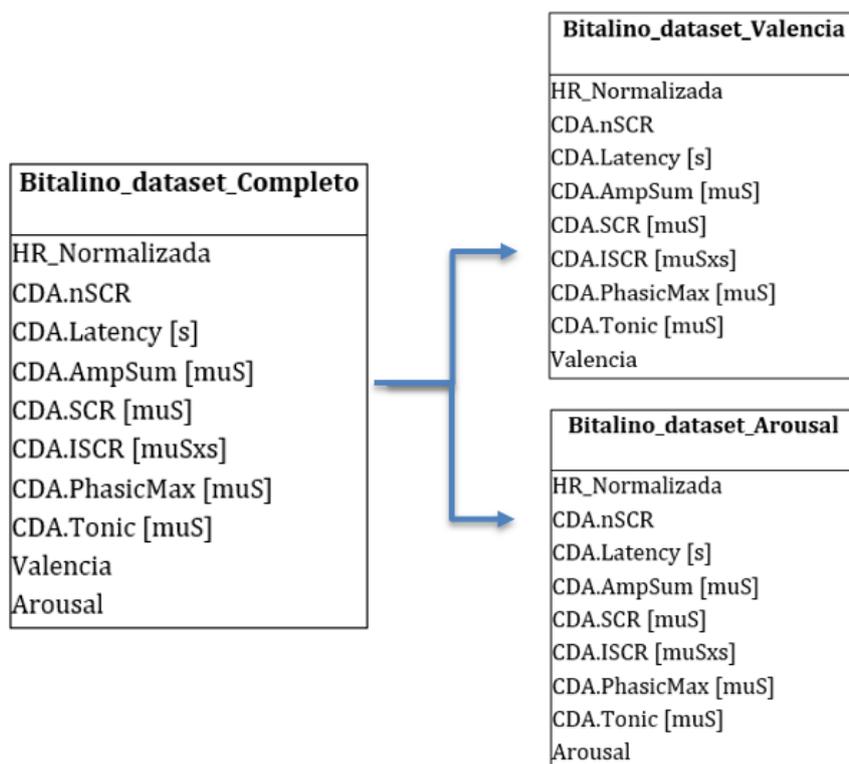
Hasta este momento se han creado 9 archivos por sensor, así como la Tabla 3-11, esto debido a que es un archivo por participante (9 personas que portaron la tecnología Bitalino y 9 personas que portaron la tecnología e-Health). En esta sección se unificarán para formar nuestros conjuntos de datos, los participantes desde el 1 hasta el 9 del sensor Bitalino conformarán un solo archivo llamado “Bitalino\_Dataset.csv”, de igual manera para E-Health, los 9 participantes que usaron este sensor se agruparán en un archivo llamado “E-Health\_Dataset.csv”.

**TABLA 3-12: EJEMPLO DE SET DE DATOS (BITALINO\_DATASET.CSV)**

ECG norm	CDA.nSCR	CDA.Latency [s]	CDA.AmpSum [muS]	CDA.SCR [muS]	CDA.ISCR [muSxs]	CDA.PhasicMax [muS]	CDA.Tonic [muS]	Valence	Arousal
0.1045093	-0.0589569	-0.2993298	-0.2888297	-0.294422	-0.294422	-0.5163375	-0.0357031	7	7
0.23119137	-1.1496597	-2.6951998	-0.6581969	-0.6404296	-0.6404296	-0.6377799	-0.0386174	1	9
0.08703288	-0.0589569	-0.6688727	-0.1083382	-0.2843525	-0.2843525	-0.1297228	0.0541102	9	4
0.03464576	-0.0589569	-0.5456917	-0.259081	-0.2371786	-0.2371786	-0.3499594	-0.1266856	9	6
0.10096920	-1.1496597	-2.6951998	-0.6581969	-0.4860102	-0.4860102	-0.5900994	-0.0527769	1	9

### 3.5.3. Preparación del Dataset

El último paso para la generación del dataset final requiere filtrar algunos eventos y asignar categorías a los datos numéricos de Arousal y Valencia. Por ejemplo, una persona no pudo haber tenido 0 pulsaciones durante 4 segundos, es por eso que los renglones que contenían estos datos fueron removidos. Finalmente, dado que en nuestro análisis solo es de interés los valores bajo o alto de Arousal y bajo o alto de Valencia, la conversión de estas columnas de valores numéricos a categóricos se hace a través del establecimiento de umbrales, “1” o bajo: valor $\leq$ 4; “2” o medio: valor = 5; “3” o alto: valor $\geq$ 6. Y como al hacer esto, la categoría media no se usará en el dataset, esta remoción de la categoría media se debe realizar de forma separada para los estados afectivos (ver Figura 3-32 como ejemplo), esto con el objetivo de evitar la doble eliminación.



**FIGURA 3-32: EJEMPLO DE SEPARAR EL CONJUNTO DE DATOS COMPLETO DE BITALINO EN VALENCIA Y AROUSAL**

Una vez separadas los archivos, se eliminan los valores medios de Valencia y de Arousal y como resultado, junto con los procesos anteriores, se obtienen para cada sensor, dos conjuntos de datos, una para Valencia y otro para Arousal. Estos, en el formato que se muestra en la Tabla 3-13. El formato o estructura presentada del set de datos, es el que se

utiliza en el proceso de clasificación, en donde se muestran 8 características fisiológicas, y la clase a la que se hace referencia.

Tabla 3-13: Ejemplo de formato del dataset final a ser utilizado en la etapa de clasificación (Bitalino\_Dataset\_Valencia.csv)

ECG norm	CDA.nSCR	CDA.Latency [s]	CDA.AmpSum [muS]	CDA.SCR [muS]	CDA.ISCR [muSxs]	CDA.PhasicMax [muS]	CDA.Tonic [muS]	Valence
0.1045093	-0.0589569	-0.2993298	-0.2888297	-0.294422	-0.294422	-0.5163375	-0.0357031	3
0.23119137	-1.1496597	-2.6951998	-0.6581969	-0.6404296	-0.6404296	-0.6377799	-0.0386174	1
0.08703288	-0.0589569	-0.6688727	-0.1083382	-0.2843525	-0.2843525	-0.1297228	0.0541102	3
0.03464576	-0.0589569	-0.5456917	-0.259081	-0.2371786	-0.2371786	-0.3499594	-0.1266856	3
0.10096920	-1.1496597	-2.6951998	-0.6581969	-0.4860102	-0.4860102	-0.5900994	-0.0527769	1

### 3.6. Clasificación

En la Tabla 3-14 se observan algunos trabajos que han estudiado la clasificación de estados afectivos con diferentes técnicas. Aunque existen otros trabajos que abordan la clasificación de estados afectivos, se seleccionaron estos cuatro trabajos como referencia base para el trabajo desarrollado con la clasificación de estados afectivos en esta tesis. Estos trabajos también utilizan datos de la frecuencia cardiaca y respuesta galvánica de la piel, así como las IAPS y algunos el sistema de evaluación de la percepción del usuario SAM.

TABLA 3-14: TRABAJOS RELACIONADOS RESPECTO A CLASIFICACIÓN DE EMOCIONES

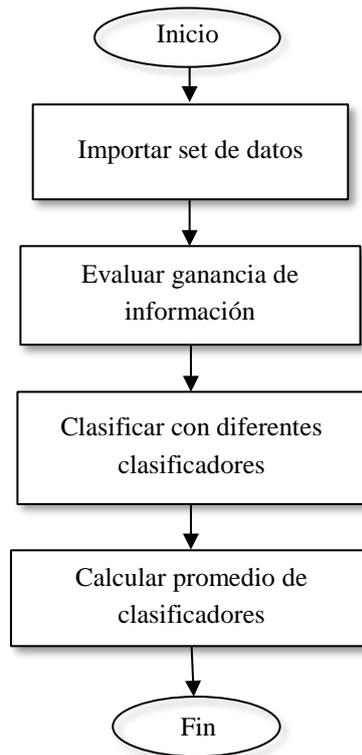
Título	Año	Clasificación	Clasificadores utilizados	Mejor porcentaje			
Affect Detection in Normal Groups with the Help of Biological Markers [35]	2015	Arousal y Valencia altos y bajos	KNN, Linear Discriminant Analysis, Quadratic Discriminant Classifier y Naïve Bayes	VALENCIA Y AROUSAL QDC			
				9 SEÑALES	66.25%		
Emotion Detection Using Noninvasive Low Cost Sensors [11]	2017	Arousal y Valencia altos y bajos	SVM, NB y J48	VALENCIA SVM		AROUSAL SVM	
				EEG + GSR + EMG	58.5%	EEG + GSR	63.8%
Emotion sensing from physiological signals using three defined areas in arousal-valence model [36]	2017	Arousal y Valencia alto, medio y bajo	SVM	VALENCIA SVM GAUSSIANO		AROUSAL SVM GAUSSIANO	
				FISIÓN (ECG, GSR, RESP, TEMP)	54,73%	FISIÓN (ECG, GSR, RESP, TEMP)	56,83%
Identification of Arousal and Relaxation by using SVM-Based Fusion of PPG Features [3]	2011	Arousal y relajación	SVM	RELAJACIÓN SVM		AROUSAL SVM	
				PPG	76.00%	PPG	73.33%

El trabajo “Identificación of Arousal and Relaxation by using SVM-Based Fusion of PPG Features”, en la Tabla 3-14, utiliza una configuración de estudio similar al nuestro e inicialmente se realizó la replicación de la metodología de clasificación propuesta por los autores, pero ajustado a los datos recolectados en nuestro estudio, sin embargo, como se muestra en el apéndice F, nuestros resultados son bajos comparados contra el trabajo de referencia.

En virtud de que no existe una técnica de clasificación única de referencia, o al menos hasta donde se ha podido observar, y que al parecer no existen base de datos disponibles recolectados para contextos de estudio de estados afectivos similares, nuestra orientación en lo referente a la clasificación fue la exploración de diferentes técnicas de clasificación: Random-Forest, Support Vector Machine (SVM) con kernel RBF, redes neuronales (MLP), KNN y Decision-Tree.

### 3.6.1. Proceso de clasificación

El diagrama de flujo de la Figura 3-33 describe de forma general el proceso realizado para la clasificación de los estados afectivos. Para ello, primeramente, se importa el set de datos a trabajar (se tienen 2 sets de datos por sensor). Sobre este set de datos se aplica el algoritmo para la selección de características teniendo como referencia la contribución individual de cada una de las características, información obtenida con el método *mutual\_info\_classif*, disponible en las librerías de scikit-learn en Python [20].



**FIGURA 3-33: DIAGRAMA DE FLUJO PARA LA CLASIFICACIÓN**

Para cada vector de características resultante, se generaba un nuevo vector de características a partir de ir retirando la característica menos significativa, y se aplicó cada uno de los algoritmos de clasificación arriba mencionados. Para calificar la contribución del vector de características se promediaron los resultados de salida de cada clasificador. El promedio más alto se utilizó como referencia para elegir el vector de características a utilizar para cada estado afectivo, de cada una de las dos tecnologías de biosensado. Un ejemplo de este proceso se muestra en la Figura 3-34, en donde al excluir las columnas del vector de ganancia de información más bajo, cada vez fue subiendo más el promedio de clasificación de los clasificadores.

Selección de características									
mutual_info_classif									
[:,[0,1,2,3,4,5,6,7,8]]	[:,[0,1,2,3,5,6,7,8]]	[:,[0,1,2,3,5,7,8]]	[:,[0,1,2,3,5,8]]	[:,[0,1,2,3,8]]					
RandomForest	RandomForest	RandomForest	RandomForest	RandomForest					
0.6785 (+/- 0.0924)	0.6693 (+/- 0.0921)	0.6820 (+/- 0.1030)	0.6700 (+/- 0.1308)	0.7067 (+/- 0.0847)					
SVM rbf	SVM rbf	SVM rbf	SVM rbf	SVM rbf					
0.3673 (+/- 0.0005)	0.3673 (+/- 0.0005)	0.3673 (+/- 0.0005)	0.3673 (+/- 0.0005)	0.3673 (+/- 0.0005)					
MLP	MLP	MLP	MLP	MLP					
0.4227 (+/- 0.1175)	0.4483 (+/- 0.0899)	0.4325 (+/- 0.0968)	0.4634 (+/- 0.0931)	0.4249 (+/- 0.1608)					
KNN	KNN	KNN	KNN	KNN					
0.6221 (+/- 0.1216)	0.6312 (+/- 0.0958)	0.6436 (+/- 0.0751)	0.6220 (+/- 0.0677)	0.6482 (+/- 0.0875)					
DecisionTree	DecisionTree	DecisionTree	DecisionTree	DecisionTree					
0.6671 (+/- 0.0891)	0.6640 (+/- 0.0746)	0.6795 (+/- 0.1017)	0.6877 (+/- 0.0540)	0.6784 (+/- 0.0928)					
prom:	0.55154431	0.55602338	0.56099102	0.56206154	0.56510656				

**FIGURA 3-34: EJEMPLO DE SELECCIÓN DE CARACTERÍSTICAS**

Al finalizar, se obtuvieron las columnas o las características que mejor resultados obtuvieron, estas características se pueden observar en la Tabla 3-15, hay que resaltar que las columnas empiezan a contar en el índice 0.

**TABLA 3-15: CARACTERÍSTICAS SELECCIONADAS POR DATASET**

<b>Nombre del Dataset</b>	<b>Columnas utilizadas</b>
Bitalino_Dataset_Valencia.csv	[0,1,2,3,4,7]
Bitalino_Dataset_Arousal.csv	[0,1,2,3]
E-Health_Dataset_Valencia.csv	[0,1,2,3,4,5,6,7]
E-Health_Dataset_Arousal.csv	[0,1,3,5,7]

## Capítulo 4

### Resultados

En la sección de metodología se presentaron algunos de los resultados que integran este trabajo de tesis, en relación con la configuración del estudio, la integración de recursos visuales/tecnológicos, y finalmente la recolección del conjunto de datos a partir del cual se realiza la identificación de estados afectivos.

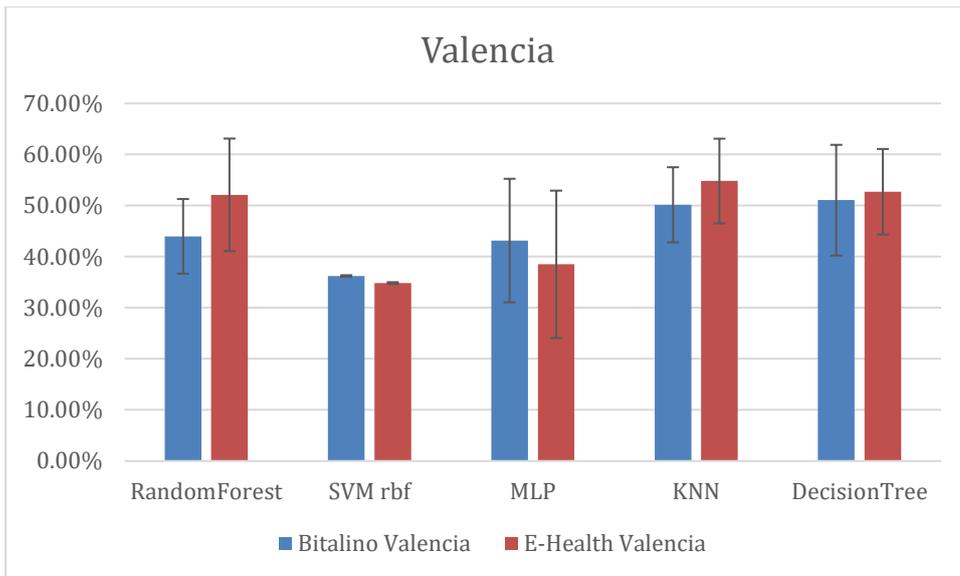
En esta sección se muestran los resultados de clasificación de los componentes afectivos de Valencia y Arousal al aplicar las técnicas de Random-Forest, Support Vector Machine (SVM) con kernel RBF, redes neuronales (MLP), KNN y Decision-Tree.

En la Tabla 4-1, se muestran los resultados de los clasificadores y se resaltan aquellos que obtuvieron un porcentaje mayor para cada sensor (e-Health y Bitalino). El porcentaje de error (valor entre paréntesis), se obtuvo a partir de la desviación estándar.

**TABLA 4-1: RESULTADOS DE CLASIFICACIÓN (CROSS VALIDATION)**

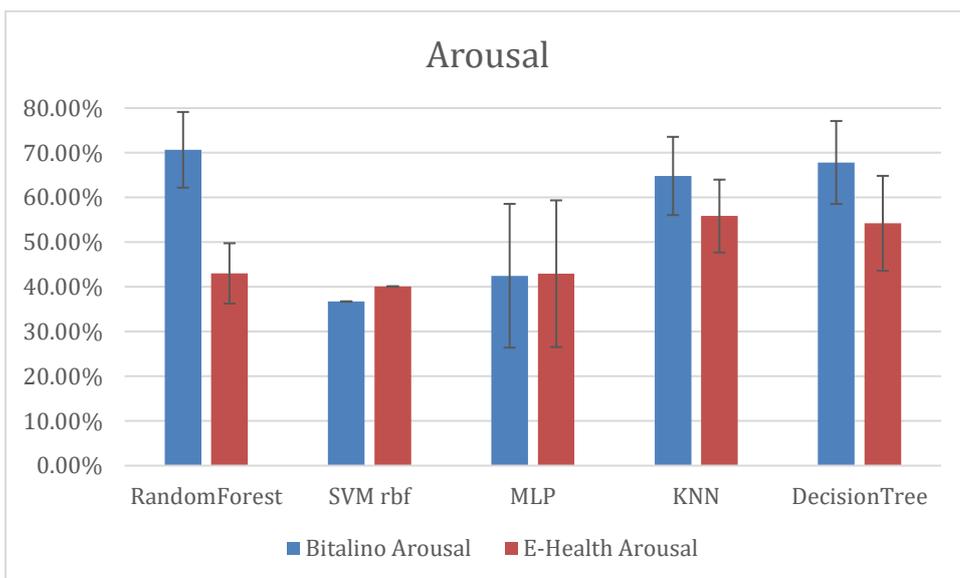
		RandomForest	SVM rbf	MLP	KNN	DecisionTree
Valencia	Bitalino	0.4398 (+/- 0.0731)	0.3621 (+/- 0.0016)	0.4315 (+/- 0.1210)	0.5016 (+/- 0.0735)	<b>0.5106 (+/- 0.1084)</b>
	E-Health	0.5211 (+/- 0.1101)	0.3483 (+/- 0.0018)	0.3850 (+/- 0.1443)	<b>0.5481 (+/- 0.0829)</b>	0.5271 (+/- 0.0837)
Arousal	Bitalino	<b>0.7067 (+/- 0.0847)</b>	0.3673 (+/- 0.0005)	0.4249 (+/- 0.1608)	0.6482 (+/- 0.0875)	0.6784 (+/- 0.0928)
	E-Health	0.4302 (+/- 0.0675)	0.4010 (+/- 0.0006)	0.4295 (+/- 0.1641)	<b>0.5585 (+/- 0.0815)</b>	0.5422 (+/- 0.1062)

En base al porcentaje de clasificación mostrado en ambos dispositivos, se puede observar que, para la clasificación de Valencia, hubo una mejora poco significativa entre los dos sensores, obteniendo un 51.06% con “DecisionTree” para Bitalino y un 54.81% en “KNN” para E-Health. Mientras que al clasificar Arousal, en Bitalino se alcanzó una clasificación de 70.67% con “RandomForest” y para E-Health un porcentaje bastante menor, obteniendo un 55.85% con “KNN”. Los datos presentados en la Tabla 4-1, se graficaron en la Figura 4-1 y Figura 4-2, presentadas a continuación:



**FIGURA 4-1: GRAFICA COMPARATIVA ENTRE SENSORES – VALENCIA**

En la Figura 4-1, se puede apreciar que el sensor E-Health, aparte de tener la mejor clasificación para Valencia, supera la clasificación de Bitalino en tres de los 5 clasificadores.



**FIGURA 4-2: GRAFICA COMPARATIVA ENTRE SENSORES - AROUSAL**

En esta tabla comparativa de Arousal (Figura 4-2), se observa que el sensor Bitalino supera por mucho la clasificación con el clasificador “Random-Forest”, además de ganar en clasificación en 3 de las 5 mejores clasificaciones obtenidas.

## Capítulo 5

### Discusión de resultados

En este trabajo de tesis, uno de los componentes interesantes es la comparación en el desempeño de dos tecnologías de biosensado, e-Health y Bitalino, la utilidad de los datos recolectados a través de estas tecnologías y el nivel de clasificación de estados afectivos que puede obtenerse.

Por una parte, debe observarse que de los trabajos encontrados en la literatura se identifican una variedad de tecnologías de sensado, desde las más costosas como la pulsera empática<sup>3</sup>, hasta sistemas embebidos diseñados a nivel laboratorio, pero en cada trabajo individual solo se utiliza una tecnología para la recolección de datos. Por otra parte, en cada trabajo se asume que la fuente de recolección de datos utilizada será suficiente para poder estudiar la clasificación e identificación de las variables de interés. Estas dos premisas sirvieron de motivador para el trabajo de tesis aquí presentado.

*Desarrollar un trabajo comparativo en el desempeño de dos tecnologías de biosensado, de bajo costo, a partir de la clasificación de estados afectivos.*

Primero, de los resultados obtenidos a partir de los clasificadores no puede establecerse de forma categórica un nivel de predominancia en el desempeño de alguna de las tecnologías de sensado: e-Health y Bitalino. Mientras que una tecnología proporciona datos y características del componente de Arousal que se ajustan mejor a un clasificador, el uso de los datos de la otra tecnología ofrece resultados más bajos. Sin embargo, para el caso del componente de Valencia el caso es a la inversa.

Segundo, si bien es cierto que las diferencias en los resultados de clasificación no establecen márgenes de comparación significativa que permita limitar el uso de cualquiera de las dos tecnologías (e-Health y Bitalino), tampoco puede establecerse que estas tecnologías no son competitivas ya que trabajos reportados en la literatura que utilizan metodologías similares a las utilizadas en esta tesis obtienen niveles similares de clasificación.

En resumen, podemos establecer que las emociones que se inducen con el conjunto de imágenes IAPS pueden ser clasificadas por los parámetros fisiológicos ECG y EDA. Respecto a la comparación del desempeño de las tecnologías de biosensado puede decirse que ya sea e-Health y Bitalino pueden cubrir las necesidades de recolección de datos. Cualquiera de estas tecnologías también puede permitir obtener niveles de clasificación comparables con otras tecnologías utilizadas en otros trabajos de la literatura. Sin embargo, existen otros autores que sugieren aumentar el número y tipo de tecnología de sensado para, al parecer, complementar las limitantes de clasificación a partir de frecuencia cardiaca y respuesta galvánica de la piel. Por ejemplo [11], hace una fusión de sensores agregando datos electroencefalográficos.

## Capítulo 6

### Conclusiones y trabajo futuro

En este trabajo de tesis se desarrolló un estudio comparativo de las tecnologías de biosensado e-Health y Bitalino tomando como contexto la inducción de estados afectivos mediante el IAPS (Sistema Internacional de Imágenes Afectivas) y recolectando la percepción del usuario utilizando SAM (Self-Assessment Maniquin). Para cada una de estas tecnologías se realizó el pre-procesamiento necesario para las señales, se desarrolló la interface del usuario a través de la cual se indujeron los estados afectivos, y se obtuvo el set de datos tanto para e-Health como para Bitalino. Se obtuvieron los vectores de características que mejor contribuyeron con la clasificación de los estados afectivos, y en general se obtuvieron niveles de clasificación similares a los identificados en la literatura.

Sin embargo, a pesar de haber obtenido una adecuada clasificación de los componentes afectivos de Arousal y Valencia, no se logra ser concluyente respecto a diferenciar el nivel de competitividad entre las tecnologías e-Health y Bitalino. Una de las limitantes en los resultados obtenidos posiblemente está relacionada con el tamaño de la población que participó en este estudio, además de otras características como la heterogeneidad de la misma. La falta de un criterio de inclusión o selección de participantes pudo haber influido en el nivel de percepción del usuario respecto al estado afectivo inducido. Por ejemplo, en nuestro rango de edad de participantes, 19 a 54 años, existieron diferentes cohortes generacionales y algunos de los participantes estudiantes en el área de la tecnología, así que para este grupo de personas algunas imágenes parecían algo antiguas. Una persona adulta, por ejemplo, indicaba que “La imagen de la pistola, parecía más una pintura que una pistola de verdad”. Por lo tanto, una estrategia a incorporar en futuros estudios sería considerar otras fuentes de inducción de estímulos afectivos. Con estas consideraciones podría ser posible contar con un data set más homogéneo que posiblemente pueda ayudar en incrementar la eficiencia de algunos de los clasificadores aquí explorados.

Finalmente, una vía adicional de trabajo está relacionada con un estudio a profundidad del comportamiento de las técnicas de clasificación. Primero, sería interesante desarrollar un sistema automatizado para el pre-procesamiento de los datos, segundo, estudiar

la posibilidad de generalizar un clasificador para un tipo de dato, ECG o EDA, independientemente de la tecnología utilizada. Por último, sería recomendable extender este estudio a contextos de aplicación más específicos como por ejemplo en el tratamiento de ansiedad generalizada o de fobias, tema de interés en el área de la salud mental del ser humano.

## Referencias

- [1] J. T. Cacioppo, L. G. Tassinary and G. G. Berntson, Handbook of psychophysiology, 3 ed., Cambridge University Press, 2007.
- [2] J. Beck and R. Egger, "Emotionalise Me: Self-reporting and Arousal Measurements in Virtual Tourism Environments," *Springer, Cham*, pp. 3-15, 2018.
- [3] C. J. Kim, W. Mincheol y E. C. Lee, Identification of Arousal and Relaxation by using SVM-Based Fusion of PPG Features, International Conference on Cognitive and Neural Systems Engineering, 2011.
- [4] M. Merone, P. Soda, M. Sansone y C. Sansone, «ECG databases for biometric systems: A systematic review,» *Expert Systems With Applications*, nº 67, pp. 189-202, 2017.
- [5] S. Masafi, O. Rezaei and H. Ahadi, Efficacy of biofeedback associated with relaxation in decreasing anxiety in women with breast cancer during chemotherapy, *Procedia-Social and Behavioral Sciences*, 2011, pp. 143-148.
- [6] R. C. Freire, M. R. De Carvalho, J. Mateus, W. A. Zin and A. E. Nardi, "Anxiogenic properties of a computer simulation for panic disorder with agoraphobia," *Journal of Affective Disorders*, no. 125, p. 301–306, 2010.
- [7] A. Betella and P. F. Verschure, The affective slider: A digital self-assessment scale for the measurement of human emotions, *PloS one*, 2016.
- [8] A. Greco, G. Valenza, L. Citi y E. P. Scilingo, Arousal and Valence Recognition of Affective Sounds based on Electrodermal Activity, *IEEE Sensors Journal*, 2016, pp. 716-725.
- [9] M. M. Bradley y P. J. Lang, The International Affective Digitized Sounds (2nd Edition; IADS-2): Affective ratings of sounds and instruction manual, University of Florida, Gainesville, FL, 2007.
- [10] D. Kukolja, Popović, Siniša, B. Dropuljić, Horvat, Marko and K. Čosić, "Real-Time Emotional State Estimator for Adaptive Virtual Reality Stimulation," *Springer*, pp. 175-184, Julio 2009.
- [11] D. Girardi, F. Lanubile y N. Novielli, «Emotion Detection Using Noninvasive Low Cost Sensors,» *IEEE*, pp. 125-130, Octubre 2017.

- [12] H. Yun, A. Fortenbacher, Pinkwart, Niels, T. Bisson and F. Moukayed, A Pilot Study of Emotion Detection using Sensors in a Learning Context: Towards an Affective Learning Companion, Chemnitz, Germany: Carsten Ullrich, Martin Wessner (Eds.): Proceedings of DeLFI and GMW Workshops 2017 , 2017.
- [13] C. Xia, A. Touroutoglou, K. S. Quigley, L. Feldman Barrett and B. C. Dickerson, "Salience Network Connectivity Modulates Skin Conductance Responses in Predicting Arousal Experience," *Journal of cognitive neuroscience*, pp. 827-83, Mayo 2017.
- [14] G. Riva, A. Gorini y A. Gaggioli, The Intrepid project-biosensor-enhanced virtual therapy for the treatment of generalized anxiety disorders, Italia: Studies in health technology and informatics, 2009.
- [15] M. Alcañiz, R. Baños, C. Botella y B. Rey, The EMMA project: emotions as a determinant of presence, *PsychNology Journal*, 2003, pp. 141-150.
- [16] F. Gargiulo, A. Fratini, M. Sansone y C. Sansone, «Subject identification via ECG fiducial-based systems: Influence of the type of qt interval correction,» *Computer Methods and Programs in Biomedicine*, vol. III, nº 121, pp. 127-135, 2015.
- [17] Libelium, «Cooking Hacks,» 07 Octubre 2015. [En línea]. Available: <https://www.cooking-hacks.com/documentation/tutorials/ehealth-v1-biometric-sensor-platform-arduino-raspberry-pi-medical>. [Último acceso: 13 Septiembre 2018].
- [18] D. J. Matich, *Informática Aplicada a la Ingeniería de Procesos – Orientación I*, Universidad Tecnológica Nacional, 2001.
- [19] OpenCV, «Introduction to Support Vector Machines,» 24 Nov 2017. [En línea]. Available: [https://docs.opencv.org/2.4/doc/tutorials/ml/introduction\\_to\\_svm/introduction\\_to\\_svm.html](https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html). [Último acceso: Nov 2017].
- [20] L. Buitinck, G. Louppe, M. Blondel, P. F. A. Mueller, O. Grisel and R. Layton, API design for machine learning software: experiences from the scikit-learn project, arXiv preprint arXiv:1309.0238, 2013.
- [21] scikit-learn developers, "scikit-learn user guide," 15 Mayo 2019. [Online]. Available: [https://scikit-learn.org/stable/\\_downloads/scikit-learn-docs.pdf](https://scikit-learn.org/stable/_downloads/scikit-learn-docs.pdf). [Accessed 17 Mayo 2019].
- [22] P. J. Lang, M. M. Bradley y B. N. Cuthbert, *International Affective Picture System (IAPS)*, Florida: University of Florida, 2008.
- [23] Bitalino, "OpenSignals (r)evolution (v.2018)," *Plux*, 23 Julio 2018. [Online]. Available:

- <https://bitalino.com/en/software>. [Accessed Septiembre 2018].
- [24] R. Meier, «CoolTerm 1.6.0,» Roger Meier's Freeware, 2018. [En línea]. Available: <http://freeware.themeiers.org>. [Último acceso: Septiembre 2018].
- [25] The MathWorks, Inc., MATLAB and Statistics Toolbox Release R2018a, Natick, Massachusetts, United States, 2018.
- [26] H. P. da Silva, A. Lourenço, A. Fred, N. Raposo and M. Aires-de-Sousa, Check Your Biosignals Here: A new dataset for off-the-person ECG biometrics, Lisboa, Portugal: computer methods and programs in biomedicine 113, 2014, p. 503–514.
- [27] C. So-In, C. Phaudphut and K. Rujirakul, "Real-Time ECG Noise Reduction with QRS Complex Detection for Mobile Health Services," *Arabian Journal for Science and Engineering*, pp. 2503-2514, 2015.
- [28] MathWorks, "MathWorks," 2019. [Online]. Available: <https://es.mathworks.com/help/signal/ref/findpeaks.html>. [Accessed Abril 2019].
- [29] iMotions, "iMotions," Enero 29 2019. [Online]. Available: <https://imotions.com/blog/skin-conductance-response/>. [Accessed 10 Marzo 2019].
- [30] PLUX – Wireless Biosignals, "REVOLUTION EDA Sensor Datasheet," 2016. [Online]. Available: [https://bitalino.com/datasheets/REVOLUTION\\_EDA\\_Sensor\\_Datasheet.pdf](https://bitalino.com/datasheets/REVOLUTION_EDA_Sensor_Datasheet.pdf). [Accessed Abril 2019].
- [31] M. & K. C. Benedek, A continuous measure of phasic electrodermal activity, *Journal of neuroscience methods*, 190(1), 80-91., 2010.
- [32] W. Boucsein, *Electrodermal Activity*, Springer, 2012.
- [33] K. Schaaff and M. T. P. Adam, Measuring Emotional Arousal for Online Applications: Evaluation of Ultra-Short Term Heart Rate Variability Measures, Humaine Association Conference on Affective Computing and Intelligent Interaction: IEEE, 2013.
- [34] J. J. Braithwaite, D. G. Watson, R. Jones and M. Rowe, "A Guide for Analysing Electrodermal Activity (EDA) & Skin Conductance Responses (SCRs) for Psychological Experiments," *Psychophysiology*, pp. 1017-1034, 2013.
- [35] S. Basu, A. Bag, M. Mahadevappa, J. Mukherjee and R. Guha, Affect detection in normal groups with the help of biological markers, 2015 Annual IEEE India Conference (INDICON), 2015.

- [36] M. B. H. Wiem and Z. Lachiri, "Emotion sensing from physiological signals using three defined areas in arousal-valence model," *IEEE*, pp. 219-223, Enero 2017.
- [37] J. Penders, B. Grundlehner, R. Vullers y B. Gyselinckx, «Potential and Challenges of Body Area Networks for Affective Human Computer Interaction,» *Springer*, p. 202–211, Julio 2009.

## Glosario

**TABLA 6-1: GLOSARIO**

<b>Nombre</b>	<b>Significado</b>
ADC	Análisis de Descomposición Continua
Decision Tree	Árboles de decisión
Down sample	Disminución de frecuencia de muestreo
ECG	Electrocardiograma
EEG	Electroencefalograma
IAPS	International Affective Picture System (Sistema Internacional de Imágenes Afectivas)
KNN	K vecinos más cercanos
MLP	Perceptrón Multicapa
PPG	Photoplethysmogram (Fotoplethysmografía)
PPM	Pulsaciones por minuto
QDC	Quadratic Discriminant Classifier (Clasificador Discriminante Cuadrático)
Random Forest	Bosques Aleatorios
SVM	Máquinas de vectores de Soporte

# Apéndices

## Apéndice A

La implementación en Unity consta de tres componentes principales: escenario, control IAPS y control SAM, el código de dichos componentes se encuentra a continuación:

### Guardar documento

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using System.IO;
5. using System;
6.
7. public class SaveDoc : MonoBehaviour {
8.     public static SaveDoc instance = null;
9.     private StreamWriter OutputStream;
10.    DateTime now;
11.
12.    public static SaveDoc Instance
13.    {
14.        get {
15.            return instance;
16.        }
17.    }
18.
19.    void Awake () {
20.        if (instance != null) {
21.            UnityEngine.Debug.LogError("An instance of SaveDoc already exist!");
22.            return;
23.        }
24.
25.        instance = this;
26.        OutputStream = new StreamWriter("SetA.csv", true);
27.        Write("0,new,session,started");
28.    }
29.
30.
31.    public void Write(string message){
32.        now = DateTime.Now;
33.        message = string.Format("{0:H:mm:ss}{1}", now, ";" + DateTime.UtcNow.Millisecond + "\," + message);
34.
35.        if(OutputStream != null) {
36.            Debug.Log("Entré..." + DateTime.UtcNow.Millisecond);
37.            OutputStream.WriteLine(message);
38.            Debug.Log(message);
39.            OutputStream.Flush();
40.        }
41.    }
42.
43.    void OnDestroy()
44.    {
45.        if (OutputStream != null) {
46.            OutputStream.Close();
47.            OutputStream = null;
48.        }
49.    }
50. }
```

### Controlar diapositivas

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using System.IO;
5. using System.Linq;
6.
7.
8. public class Slider : MonoBehaviour {
9.
```

```

10. [Header("Tiempo en imágenes (segundos)")]
11. public int tiempoImagenEnBlanco = 1;
12. public int tiempoImagenIAPS = 2;
13. [Space(10)]
14. public List<string> spritesList;
15. int i=0;
16. public int spriteIndex = 0;
17. public int numMaximoDeImagenes = 3;
18. public Sprite spriteEnBlanco;
19. public string lastSlider = null;
20.
21. void Start() {
22.
23.
24.
25.     spritesList = new List<string>();
26.     foreach (string file in System.IO.Directory.GetFiles(Application.dataPath + "/Resources/Sprites")) {
27.
28.         if (!file.Contains("0000.jpg") && !file.Contains("meta")) {
29.             string img = file.Replace(Application.dataPath + "/Resources/", "");
30.             img = img.Replace(".jpg", "");
31.             spritesList.Add(img);
32.         }
33.     }
34.     spritesList = ShuffleList<string>(spritesList);
35.     foreach (string t in spritesList)
36.         Debug.Log(t);
37.
38.     ChangeImage();
39. }
40.
41. private List<E> ShuffleList<E>(List<E> inputList)
42. {
43.     List<E> randomList = new List<E>();
44.
45.     System.Random r = new System.Random();
46.     int randomIndex = 0;
47.     while (inputList.Count > 0) {
48.         randomIndex = r.Next(0, inputList.Count); //Choose a random object in the list
49.         randomList.Add(inputList[randomIndex]); //add it to the new, random list
50.         inputList.RemoveAt(randomIndex); //remove to avoid duplicates
51.     }
52.
53.     return randomList; //return the new random list
54. }
55.
56.
57. public void ChangeImage() {
58.     StartCoroutine(Enu());
59.
60. }
61.
62. IEnumerator Enu() {
63.     if (spriteIndex < numMaximoDeImagenes) {
64.         this.GetComponent<SpriteRenderer>().sprite = spriteEnBlanco;
65.         yield return new WaitForSecondsRealtime(tiempoImagenEnBlanco);
66.
67.         print("Usando Sprite: " + spritesList[spriteIndex]);
68.         this.GetComponent<SpriteRenderer>().sprite = Resources.Load<Sprite>(spritesList[spriteIndex]) as Sprite;
69.         //print("Entre 2");
70.         //this.GetComponent<SpriteRenderer>().sprite = sprite2;
71.         spriteIndex++;
72.
73.         yield return new WaitForSecondsRealtime(tiempoImagenIAPS);
74.         lastSlider = this.GetComponent<SpriteRenderer>().sprite.name;
75.         this.GetComponent<SpriteRenderer>().sprite = null;
76.         this.GetComponent<ShowSAM>().HideSAM(false);
77.     } else {
78.         Debug.Log("El experimento ha finalizado");
79.         Debug.Break();
80.     }
81. }
82. }
83.

```

## Obtener botón activo

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;

```

```

5. using System.Linq;
6.
7. public class GetActiveRB : MonoBehaviour {
8.
9.     SaveDoc saveDoc;
10.
11.     public ToggleGroup valenceGroup = null;
12.     public ToggleGroup arousalGroup = null;
13.     public ToggleGroup dominanceGroup = null;
14.
15.     public GameObject sliderObject = null;
16.
17.     void Start() {
18.         saveDoc = SaveDoc.Instance;
19.     }
20.
21.     public void getValenceRB() {
22.         if (valenceGroup.ActiveToggles().FirstOrDefault() == null || arousalGroup.ActiveToggles().FirstOrDefault() == null ||
23.             dominanceGroup.ActiveToggles().FirstOrDefault() == null) {
24.             Debug.Log("Se tienen que seleccionar todos los campos de cada emoción.");
25.         } else {
26.             Debug.Log("Valence: " + valenceGroup.ActiveToggles().FirstOrDefault()
27.                 + ", Arousal: " + arousalGroup.ActiveToggles().FirstOrDefault()
28.                 + ", Dominance: " + dominanceGroup.ActiveToggles().FirstOrDefault());
29.
30.             saveDoc.Write(sliderObject.GetComponent<Slider>().spriteIndex+" "+
31.                 sliderObject.GetComponent<Slider>().lastSlider+
32.                 " "+ valenceGroup.ActiveToggles().FirstOrDefault().name +
33.                 " "+ arousalGroup.ActiveToggles().FirstOrDefault().name +
34.                 " "+ dominanceGroup.ActiveToggles().FirstOrDefault().name);
35.
36.             //saveDoc.Write(" "+ valenceGroup.ActiveToggles().FirstOrDefault()+
37.             // " "+ arousalGroup.ActiveToggles().FirstOrDefault()+
38.             // " "+ dominanceGroup.ActiveToggles().FirstOrDefault());
39.
40.             sliderObject.transform.GetComponent<ShowSAM>().HideSAM(true);
41.             sliderObject.GetComponent<Slider>().ChangeImage();
42.
43.             foreach(Transform t in valenceGroup.transform){
44.                 t.GetComponent<Toggle>().isOn = false;
45.             }foreach(Transform t in arousalGroup.transform){
46.                 t.GetComponent<Toggle>().isOn = false;
47.             }foreach(Transform t in dominanceGroup.transform){
48.                 t.GetComponent<Toggle>().isOn = false;
49.             }
50.         }
51.     }
52. }

```

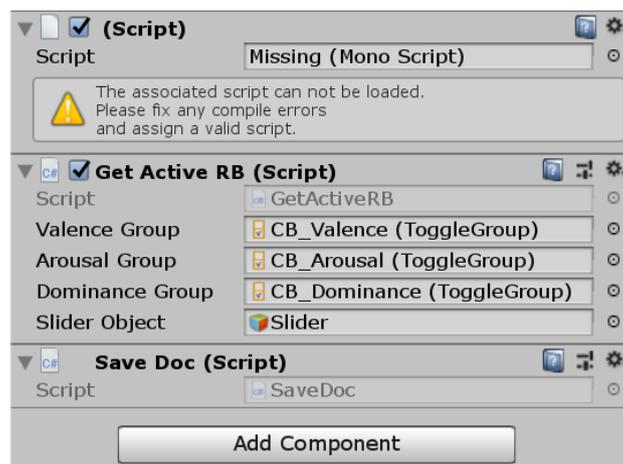


FIGURA 0-1: VENTANA DE UNITY GENERADA CON EL CÓDIGO ANTERIOR.

## Mostrar cuestionario SAM

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class ShowSAM : MonoBehaviour {
6.     public GameObject[] itemToHide;
7.
8.     void Start () {
9.         HideSAM(true);
10.    }
11.
12.    void Update () {
13.
14.    }
15.
16.    public void HideSAM(bool hide) {
17.        foreach (GameObject item in itemToHide) {
18.            item.SetActive(!hide);
19.        }
20.    }
21. }
```

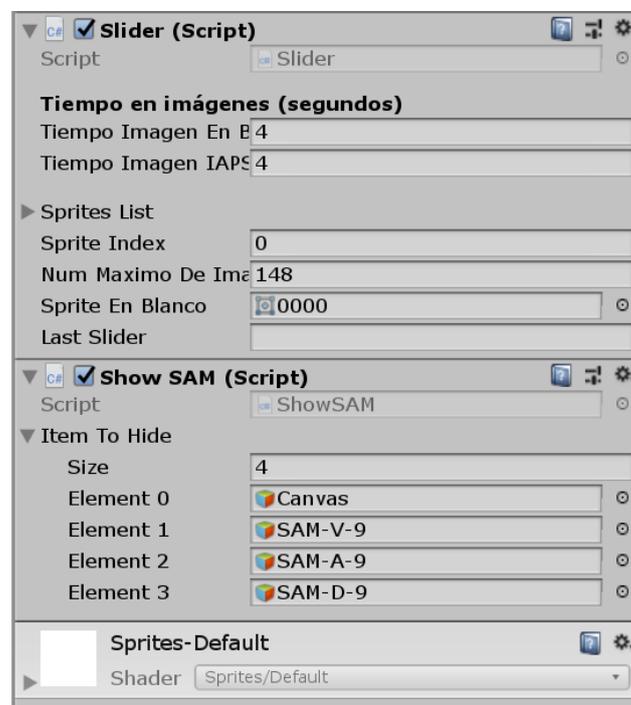


FIGURA 0-2: VENTANA DE UNITY DONDE SE CONTROLAN LOS PARÁMETROS DE LA INTERFAZ (CÓDIGO ANTERIOR)

## Apéndice B (Código Arduino para e-Health)

```
1. #include <PinChangeInt.h>
2. #include <PinChangeIntConfig.h>
3.
4. #include <eHealth.h>
5. #include <eHealthDisplay.h>
6.
7. int cont = 0;
8.
9. extern volatile unsigned long timer0_overflow_count;
10. float fanalog0;
11. int analog0;
12. unsigned long time;
13. float myTime=0;
```

```

14. float conductanceVol;
15.
16. byte serialByte;
17.
18. void setup() {
19.   Serial.begin(115200);
20.   eHealth.initPulsioximeter();
21.
22.   //Attach the intruptions for using the pulsioximeter.
23.   PCintPort::attachInterrupt(6, readPulsioximeter, RISING);
24.
25. }
26.
27. void loop() {
28.
29.   time = (timer0_overflow_count << 8) + TCNT0;
30.   // Microseconds conversion.
31.   time = (time * 4);
32.   myTime= time/100000;
33.   Serial.print(myTime/10,1);
34.
35.   Serial.print(",");
36.   GSR();
37.   Serial.print(",");
38.   ecg();
39. }
40. void readPulsioximeter() {
41.   cont ++;
42.   if (cont == 50) //Get only of one 50 measures to reduce the latency
43.     eHealth.readPulsioximeter();
44.   cont = 0;
45. }
46. }
47.
48. void pulso() {
49.   Serial.print(eHealth.getBPM());
50.   Serial.print(",");
51. }
52.
53. void respiracion() {
54.
55.   int airFlow = eHealth.getAirFlow();
56.   Serial.print(airFlow);
57.   Serial.print(",");
58. }
59.
60. void ecg() {
61.   fanalog0 = eHealth.getECG();
62.   Serial.println(fanalog0, 2);
63. }
64.
65. void GSR() {
66.   conductanceVol = eHealth.getSkinConductanceVoltage();
67.   Serial.print(conductanceVol*1000, 6);
68. }
69.
70. void tem() {
71.   Serial.print(",");
72.   Serial.print(eHealth.getTemperature(), 2);
73. }

```

## Código para Clasificar las Señales de E-Health

```

1.   package signaldataseparator;
2.
3.   import java.io.File;
4.   import java.io.FileNotFoundException;
5.   import java.io.FileReader;
6.   import java.io.FileWriter;
7.   import java.io.IOException;
8.   import java.text.ParseException;
9.   import java.text.SimpleDateFormat;
10.  import java.util.ArrayList;
11.  import java.util.Calendar;
12.  import java.util.Date;
13.  import java.util.GregorianCalendar;
14.  import java.util.Scanner;
15.  import java.util.logging.Level;
16.  import java.util.logging.Logger;
17.  import java.util.regex.Matcher;
18.  import java.util.regex.Pattern;

```

```

19. import javax.swing.JFileChooser;
20. import javax.swing.filechooser.FileNameExtensionFilter;
21.
22. /**
23.  *
24.  * @author florentino
25.  */
26. public class FormatFiles_EHealth {
27.
28.     public static void main(String[] args) throws IOException, ParseException {
29.         FileWriter fw = null;
30.         FileWriter fwDataset = null;
31.
32.         try {
33.             //Read SAM files
34.             JFileChooser fcSAM = new JFileChooser("C:\\Users\\flore\\Desktop\\pruebas\\datos_formateados\\_Dataset");
35.             fcSAM.setFileFilter(new FileNameExtensionFilter("SAM file", "csv"));
36.             fcSAM.showSaveDialog(null);
37.             File fSAM = fcSAM.getSelectedFile();
38.             String myPath = fSAM.getParent();
39.             ArrayList<ArrayList<Object>> samList = new ArrayList<ArrayList<Object>>();// new Object[200][6];
40.             File samFile = new File(fSAM.getAbsolutePath());
41.             Scanner sc1 = new Scanner(new FileReader(samFile));
42.             String rowSam = null;
43.             SimpleDateFormat sdfSam = new SimpleDateFormat("HH:mm:ss.SSS");
44.             GregorianCalendar gc = null;
45.             int samplesInlaps = 500;
46.
47.             // Para saltar los titulos
48.             sc1.nextLine();
49.
50.             while (sc1.hasNextLine()) {
51.                 rowSam = sc1.nextLine();
52.                 String[] splitRow = rowSam.split(",");
53.                 if (splitRow.length == 6) {
54.                     ArrayList<Object> rowList = new ArrayList<>();
55.                     for (int i = 0; i < 6; i++) {
56.                         if (i == 0) {
57.                             Date date = sdfSam.parse(splitRow[0]);
58.                             gc = new GregorianCalendar();
59.                             gc.setTime(date);
60.                             gc.add(Calendar.SECOND, 4);
61.                             rowList.add(gc);
62.                         }
63.                         if (i == 2) {
64.                             rowList.add(splitRow[i]);
65.                         }
66.                         if (i == 3 || i == 4 || i == 5) {
67.                             rowList.add(splitRow[i].substring(1));
68.                         }
69.                     }
70.                     samList.add(rowList);
71.                 }
72.
73.                 String[] splitTime = rowSam.split("\t");
74.                 if (splitTime.length >= 8) {
75.                     gc.add(Calendar.MILLISECOND, 10);
76.                 }
77.             }
78.
79.             // Second file
80.             JFileChooser fcSignal = new JFileChooser(myPath + "");
81.             fcSignal.setFileFilter(new FileNameExtensionFilter("Signal file", "txt"));
82.             fcSignal.showSaveDialog(null);
83.             File file = fcSignal.getSelectedFile();
84.             String subjNum = (new Scanner(file.getName())).findInLine(Pattern.compile("[0-9]+"));
85.             fw = new FileWriter(file.getParent() + "\\Sujeto" + subjNum + "_procesado.txt");
86.             fwDataset = new FileWriter(file.getParent() + "\\Sujeto" + subjNum + "_dataset.txt");
87.
88.             System.out.println("IAPS\tHR(bpm)\tGSR\tValence\tArousal\tDominance\tclass(V,A)\tclass(V,A,D)");
89.             fwDataset.append("IAPS\tHR(bpm)\tGSR\tValence\tArousal\tDominance\tclass(V,A)\tclass(V,A,D)\n");
90.             for (ArrayList<Object> arrayList : samList) {
91.                 fwDataset.append(arrayList.get(1) + "\t\t\t"
+arrayList.get(2).toString()+"\t"+arrayList.get(3).toString()+"\t"+arrayList.get(4).toString()+"\t"+
Cluster2Classifier(Integer.parseInt(arrayList.get(2).toString()),Integer.parseInt(arrayList.get(3).toString())) +
92.                 "\t"+Cluster3Classifier(Integer.parseInt(arrayList.get(2).toString()),Integer.parseInt(arrayList.get(3).toString()),
Integer.parseInt(arrayList.get(4).toString()))+"\n");
93.                 +arrayList.get(2).toString()+"\t"+arrayList.get(3).toString()+"\t"+arrayList.get(4).toString()+"\t"+
Cluster2Classifier(Integer.parseInt(arrayList.get(2).toString()), Integer.parseInt(arrayList.get(3).toString()),
Integer.parseInt(arrayList.get(4).toString()));
94.             }
95.
96.             int samListIterator = 0;
97.             Scanner sc = new Scanner(new FileReader(file));

```

```

99.     String row = null;
100.    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss.SSS");
101.
102.    int linesInlaps = 0; // 0-400
103.    while (sc.hasNextLine()) {
104.        row = sc.nextLine();
105.        Matcher mtime = Pattern.compile("[0-9]{1,2};[0-9]{1,2};[0-9]{1,2};[0-9]+").matcher(row);
106.        if (mtime.find()) {
107.            Date date = sdf.parse(mtime.group().substring(0, mtime.group().length() - 1));
108.            gc = new GregorianCalendar();
109.            gc.setTime(date);
110.        }
111.        String[] splitTime = row.split("\\t");
112.        boolean tiempoEnSam = false;
113.        if (splitTime.length >= 8) {
114.            if (gc.get(Calendar.HOUR) == ((GregorianCalendar) samList.get(samListIterator).get(0)).get(Calendar.HOUR)) {
115.                if (gc.get(Calendar.MINUTE) == ((GregorianCalendar) samList.get(samListIterator).get(0)).get(Calendar.MINUTE)) {
116.                    if (gc.get(Calendar.SECOND) == ((GregorianCalendar) samList.get(samListIterator).get(0)).get(Calendar.SECOND)) {
117.                        if (gc.get(Calendar.MILLISECOND) >= ((GregorianCalendar)
samList.get(samListIterator).get(0)).get(Calendar.MILLISECOND)
118.                            && linesInlaps == 0) {
119.                            linesInlaps++;
120.                        }
121.                    }
122.                }
123.            }
124.            if (linesInlaps <= samplesInlaps && linesInlaps != 0) {
125.                fw.append(gc.get(Calendar.HOUR_OF_DAY) + ":" + gc.get(Calendar.MINUTE) + ":" + gc.get(Calendar.SECOND) + "." +
gc.get(Calendar.MILLISECOND)
126.                    + "\\t" + splitTime[7] + "\\t" + splitTime[8] + "\\t" + samList.get(samListIterator).get(1)
127.                    + "\\t" + samList.get(samListIterator).get(2) + "\\t" + samList.get(samListIterator).get(3)
128.                    + "\\t" + samList.get(samListIterator).get(4) + "\\n");
129.                linesInlaps++;
130.            } else if (linesInlaps == samplesInlaps + 1) {
131.                if (samListIterator < samList.size() - 1) {
132.                    samListIterator++;
133.                    linesInlaps = 0;
134.                }
135.            }
136.            gc.add(Calendar.MILLISECOND, 10);
137.        }
138.    }
139.    } catch (FileNotFoundException ex) {
140.        Logger.getLogger(FormatFiles_EHealth.class.getName()).log(Level.SEVERE, null, ex);
141.    } catch (IOException ex) {
142.        Logger.getLogger(FormatFiles_EHealth.class.getName()).log(Level.SEVERE, null, ex);
143.    } finally {
144.        fw.close();
145.        fwDataset.close();
146.    }
147.    }
148.
149.    public static char RateValue(Integer e) {
150.        if (e > 6) {
151.            return 'H';
152.        } else if (e > 3) {
153.            return 'N';
154.        } else if (e > 0) {
155.            return 'L';
156.        }
157.        return '_';
158.    }
159.
160.    public static int Cluster3Classifier(Integer v0, Integer a0, Integer d0) {
161.        char v = RateValue(v0);
162.        char a = RateValue(a0);
163.        char d = RateValue(d0);
164.        int value = 1;
165.        if (v == 'N') {
166.            value += 9;
167.        } else if (v == 'L') {
168.            value += 18;
169.        }
170.        if (a == 'N') {
171.            value += 3;
172.        } else if (a == 'L') {
173.            value += 6;
174.        }
175.        if (d == 'N') {
176.            value += 1;
177.        } else if (d == 'L') {
178.            value += 2;
179.        }
180.        return value;

```

```

181.     }
182.
183.     public static int Cluster2Classifier(Integer v0, Integer a0) {
184.         char v = RateValue(v0);
185.         char a = RateValue(a0);
186.         int value = 1;
187.         if (v == 'N') {
188.             value += 3;
189.         } else if (v == 'L') {
190.             value += 6;
191.         }
192.         if (a == 'N') {
193.             value += 1;
194.         } else if (a == 'L') {
195.             value += 2;
196.         }
197.         return value;
198.     }
199. }

```

## Código para clasificar señales Bitalino

```

1.     package signaldataseparator;
2.
3.     import java.io.File;
4.     import java.io.FileNotFoundException;
5.     import java.io.FileReader;
6.     import java.io.FileWriter;
7.     import java.io.IOException;
8.     import java.text.ParseException;
9.     import java.text.SimpleDateFormat;
10.    import java.util.ArrayList;
11.    import java.util.Calendar;
12.    import java.util.Date;
13.    import java.util.GregorianCalendar;
14.    import java.util.Scanner;
15.    import java.util.logging.Level;
16.    import java.util.logging.Logger;
17.    import java.util.regex.Matcher;
18.    import java.util.regex.Pattern;
19.    import javax.swing.JFileChooser;
20.    import javax.swing.filechooser.FileNameExtensionFilter;
21.
22.    /**
23.     *
24.     * @author florentino
25.     * Format BitalineFiles
26.     */
27.    public class FormatFiles {
28.
29.        public static void main(String[] args) throws IOException, ParseException {
30.            FileWriter fw = null;
31.            FileWriter fwDataset = null;
32.
33.            try {
34.                //Read SAM files
35.                JFileChooser fcSAM = new JFileChooser("Dataset");
36.                fcSAM.setFileFilter(new FileNameExtensionFilter("SAM file", "csv"));
37.                fcSAM.showSaveDialog(null);
38.                File fSAM = fcSAM.getSelectedFile();
39.                String myPath = fSAM.getParent();
40.                ArrayList<ArrayList<Object>> samList = new ArrayList<ArrayList<Object>>();// new Object[200][6];
41.                File samFile = new File(fSAM.getAbsolutePath());
42.                Scanner sc1 = new Scanner(new FileReader(samFile));
43.                String rowSam = null;
44.                SimpleDateFormat sdfSam = new SimpleDateFormat("HH:mm:ss.SSS");
45.                GregorianCalendar gc = null;
46.                int samplesInlaps = 600;
47.
48.                // Para saltar los titulos
49.                sc1.nextLine();
50.
51.                while (sc1.hasNextLine()) {
52.                    rowSam = sc1.nextLine();
53.                    String[] splitRow = rowSam.split(",");
54.                    if (splitRow.length == 6) {
55.                        ArrayList<Object> rowList = new ArrayList<>();
56.                        for (int i = 0; i < 6; i++) {
57.                            if (i == 0) {
58.                                Date date = sdfSam.parse(splitRow[0]);

```

```

59.         gc = new GregorianCalendar();
60.         gc.setTime(date);
61.         gc.add(Calendar.SECOND, 2);
62.         rowList.add(gc);
63.     }
64.     if (i == 2) {
65.         rowList.add(splitRow[i]);
66.     }
67.     if (i == 3 || i == 4 || i == 5) {
68.         rowList.add(splitRow[i].substring(1));
69.     }
70. }
71. samList.add(rowList);
72. }
73.
74. String[] splitTime = rowSam.split("\t");
75. if (splitTime.length >= 8) {
76.     gc.add(Calendar.MILLISECOND, 10);
77. }
78. }
79.
80. // Second file
81. JFileChooser fcSignal = new JFileChooser(myPath + "");
82. fcSignal.setFileFilter(new FileNameExtensionFilter("Signal file", "txt"));
83. fcSignal.showSaveDialog(null);
84. File file = fcSignal.getSelectedFile();
85. String subjNum = (new Scanner(file.getName())).findInLine(Pattern.compile("[0-9]+"));
86. fw = new FileWriter(file.getParent() + "\\Sujeto" + subjNum + "_procesado.txt");
87. fwDataset = new FileWriter(file.getParent() + "\\Sujeto" + subjNum + "_dataset.txt");
88.
89. System.out.println("IAPS\tHR(bpm)\tGSR\tValence\tArousal\tDominance\tclass(V,A)\tclass(V,A,D)");
90. fwDataset.append("IAPS\tHR(bpm)\tGSR\tValence\tArousal\tDominance\tclass(V,A)\tclass(V,A,D)\n");
91. for (ArrayList<Object> arrayList : samList) {
92.     fwDataset.append(arrayList.get(1) + "\t\t" + Cluster1Classifier(Integer.parseInt(arrayList.get(2).toString()) + "\t" +
Cluster1Classifier(Integer.parseInt(arrayList.get(3).toString()) + "\t" + Cluster1Classifier(Integer.parseInt(arrayList.get(4).toString()) + "\t" +
93.         + Cluster2Classifier(Integer.parseInt(arrayList.get(2).toString()), Integer.parseInt(arrayList.get(3).toString())
94.         + "\t" + Cluster3Classifier(Integer.parseInt(arrayList.get(2).toString()), Integer.parseInt(arrayList.get(3).toString()),
Integer.parseInt(arrayList.get(4).toString()) + "\n");
95. +arrayList.get(2).toString()+"\t"+arrayList.get(3).toString()+"\t"+arrayList.get(4).toString()+"\t"+
Cluster2Classifier(Integer.parseInt(arrayList.get(2).toString()), Integer.parseInt(arrayList.get(3).toString()),
Integer.parseInt(arrayList.get(4).toString()));
96.     }
97.
98.     int samListIterator = 0;
99.     Scanner sc = new Scanner(new FileReader(file));
100.    String row = null;
101.    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss.SSS");
102.    int linesInlaps = 0; // 0-400
103.    int rowHour = 0;
104.    int rowMin = 0;
105.    int rowSec = 0;
106.    int rowMil = 0;
107.    int samHour = 0;
108.    int samMin = 0;
109.    int samSec = 0;
110.    int samMil = 0;
111.
112.    while (sc.hasNextLine()) {
113.        row = sc.nextLine();
114.        Matcher mtime = Pattern.compile("[0-9]{1,2};[0-9]{1,2};[0-9]{1,2};[0-9]+").matcher(row);
115.        if (mtime.find()) {
116.            Date date = sdf.parse(mtime.group().substring(0, mtime.group().length() - 1));
117.            gc = new GregorianCalendar();
118.            gc.setTime(date);
119.        }
120.        String[] splitTime = row.split("\t");
121.        boolean tiempoEnSam = false;
122.        if (splitTime.length >= 8) {
123.
124.            rowHour = gc.get(Calendar.HOUR);
125.            rowMin = gc.get(Calendar.MINUTE);
126.            rowSec = gc.get(Calendar.SECOND);
127.            rowMil = gc.get(Calendar.MILLISECOND);
128.            samHour = ((GregorianCalendar) samList.get(samListIterator).get(0)).get(Calendar.HOUR);
129.            samMin = ((GregorianCalendar) samList.get(samListIterator).get(0)).get(Calendar.MINUTE);
130.            samSec = ((GregorianCalendar) samList.get(samListIterator).get(0)).get(Calendar.SECOND);
131.            samMil = ((GregorianCalendar) samList.get(samListIterator).get(0)).get(Calendar.MILLISECOND);
132.
133.            if (rowHour == samHour) {
134.                if (rowMin == samMin) {
135.                    if (rowSec == samSec) {
136.                        if (rowMil >= samMil && linesInlaps == 0) {
137.                            linesInlaps++;
138.                        }

```

```

139.         } else if (rowSec >= samSec && linesInlaps == 0) {
140.             linesInlaps++;
141.         }
142.     } else if (rowMin >= samMin && linesInlaps == 0) {
143.         linesInlaps++;
144.     }
145. } else if (rowHour >= samHour && linesInlaps == 0) {
146.     linesInlaps++;
147. }
148.
149.     fw.append(gc.get(Calendar.HOUR_OF_DAY) + ":" + gc.get(Calendar.MINUTE) + ":" + gc.get(Calendar.SECOND) + "." +
gc.get(Calendar.MILLISECOND)
150.         + "\t" + splitTime[7] + "\t" + splitTime[8] + "\t" + samList.get(samListIterator).get(1)
151.         + "\t" + samList.get(samListIterator).get(2) + "\t" + samList.get(samListIterator).get(3)
152.         + "\t" + samList.get(samListIterator).get(4) + "\n");
153.     linesInlaps++;
154. } else if (linesInlaps == samplesInlaps + 1) {
155.     if (samListIterator < samList.size() - 1) {
156.         samListIterator++;
157.         linesInlaps = 0;
158.     }
159. }
160.     gc.add(Calendar.MILLISECOND, 10);
161. }
162. }
163. } catch (FileNotFoundException ex) {
164.     Logger.getLogger(FormatFiles.class.getName()).log(Level.SEVERE, null, ex);
165. } catch (IOException ex) {
166.     Logger.getLogger(FormatFiles.class.getName()).log(Level.SEVERE, null, ex);
167. } finally {
168.     fw.close();
169.     fwDataset.close();
170. }
171. }
172.
173. public static char RateValue(Integer e) {
174.     if (e > 6) {
175.         return 'H';
176.     } else if (e > 3) {
177.         return 'N';
178.     } else if (e > 0) {
179.         return 'L';
180.     }
181.     return '_';
182. }
183.
184. public static int Cluster3Classifier(Integer v0, Integer a0, Integer d0) {
185.     char v = RateValue(v0);
186.     char a = RateValue(a0);
187.     char d = RateValue(d0);
188.     int value = 1;
189.     if (v == 'N') {
190.         value += 9;
191.     } else if (v == 'L') {
192.         value += 18;
193.     }
194.     if (a == 'N') {
195.         value += 3;
196.     } else if (a == 'L') {
197.         value += 6;
198.     }
199.     if (d == 'N') {
200.         value += 1;
201.     } else if (d == 'L') {
202.         value += 2;
203.     }
204.     return value;
205. }
206.
207. public static int Cluster2Classifier(Integer v0, Integer a0) {
208.     char v = RateValue(v0);
209.     char a = RateValue(a0);
210.     int value = 1;
211.     if (v == 'N') {
212.         value += 3;
213.     } else if (v == 'L') {
214.         value += 6;
215.     }
216.     if (a == 'N') {
217.         value += 1;
218.     } else if (a == 'L') {
219.         value += 2;
220.     }
221. }

```

```

222.     return value;
223. }
224.
225. public static int Cluster1Classifier(Integer a0) {
226.     char a = RateValue(a0);
227.     int value = 1;
228.
229.     if (a == 'N') {
230.         value += 1;
231.     } else if (a == 'L') {
232.         value += 2;
233.     }
234.
235.     return value;
236. }
237. }
238.

```

## Apéndice C - Downsample

```

1.     public static void Downsample(ArrayList<ArrayList<Object>> samList) throws ParseException, IOException {
2.         float increment = (float) (samList.size() / 100.0);
3.         float index = increment;
4.         float avgGSR = 0;
5.         int indexSum = 0;
6.         float avgHR = 0;
7.
8.         GregorianCalendar gc = null;
9.         if (samList.size() > 0) {
10.            gc = (GregorianCalendar) samList.get(0).get(0);
11.        }
12.        for (int i = 0; i < samList.size(); i++) {
13.            avgGSR += Float.parseFloat(samList.get(i).get(1).toString());
14.            avgHR += Float.parseFloat(samList.get(i).get(2).toString());
15.            indexSum++;
16.            if (((int) index == i + 1) {
17.                ArrayList<Object> rowList = new ArrayList<>();
18.                rowList.add(gc.get(Calendar.HOUR_OF_DAY) + ":" + gc.get(Calendar.MINUTE) + ":" + gc.get(Calendar.SECOND) + "." +
gc.get(Calendar.MILLISECOND));
19.                rowList.add((avgGSR / indexSum));
20.                rowList.add((avgHR / indexSum));
21.                downsampleList.add(rowList);
22.                fw.append(rowList.get(0) + "\t" + rowList.get(1) + "\t" + rowList.get(2) + "\n");
23.                index += increment;
24.                avgGSR = 0;
25.                avgHR = 0;
26.                indexSum = 0;
27.                gc.add(Calendar.MILLISECOND, 10);
28.            }
29.        }
30.        samList = new ArrayList<>();
31.    }

```

## Apéndice D - Determinar PPM en Matlab

### Código en Matlab para determinar el promedio de pulsaciones por ventana (evento)

```

1.     array = (1:148)';
2.
3.     minHRValue = 40;
4.     maxHRValue = 160;
5.
6.     for i = 1
7.
8.         % Load ECG Signal column and init variables
9.         data = table2array(Data(i*400-399:i*400,1));
10.        % data = table2array(Data(:,1));
11.        Fs = 100;
12.        Necg = length(data);
13.        xn_new = data;
14.        tn = (0:Necg-1)/Fs;%(0:Necg-1)/Fs;
15.
16.        % Plot Original signal

```

```

17. % plot(tn,xn_new, 'b');
18. % title('Original signal'); xlabel('Time (s);ylabel('Voltage (mV));
19. % hold on;
20. % grid on;
21.
22. % Plot Detrend
23. xn = detrend(xn_new);
24. % plot(tn,xn, 'r');
25. % legend({'Before De-trending','After De-trending'});
26. % xlabel('Time (s);
27. % ylabel('Voltage (mV));
28. % grid on;
29.
30. % Filter design (Fp is based on the plot signal)
31. N = 1;
32. Fp = 30;
33. Ap =1;
34. h = fdesign.lowpass('N,Fp,Ap', N,Fp, Ap, Fs);
35. d = design(h,'cheby1');
36. xfilter = filter(d, xn);
37.
38. figure;
39. [pxx,fx] = pwelch(xn,[],[],Fs);
40. [pff,ff] = pwelch(xfilter,[],[],Fs);
41. plot(fx, pxx);
42. plot(ff, pff);
43.
44.
45. figure ;
46. grid on;
47. xfiltfilt = filtfilt(d.sosMatrix,d.ScaleValues,xn);
48. plot(tn,xn, 'b', tn,xfiltfilt, 'r');
49. plot(tn,xn, 'r', tn,xfilter, 'w', tn,xfiltfilt, 'b');
50.
51.
52. [pks, locs] = findpeaks(xfiltfilt, 'MinPeakHeight',150,'MinPeakDistance', 20);
53. figure;
54. hold on;
55. plot(tn,data, 'b',tn,xfiltfilt, 'r');
56. plot(tn(locs),pks,'vr', 'MarkerFaceColor','r','MarkerSize',6);
57. legend({'Original Signal','Filtered Signal','R Peaks'});
58. title(i); xlabel('Time (s);ylabel('Voltage (mV));
59. set(gcf, 'NumberTitle', 'Off', 'Name', 'Filtered Signal vs Actual Signal');
60.
61. avgBpm = 0;
62. for x = 1:length(locs)-1
63.     avgBpm = avgBpm + locs(x+1)-locs(x);
64. end
65. avgBpm = 6000/(avgBpm/(length(locs)-1)); % es igual a 60/averageBpm*100
66. if (avgBpm<minHRValue || avgBpm>maxHRValue)
67.     avgBpm=0;
68. end
69. disp("Bpm: " + avgBpm);
70. array(i)= avgBpm;
71.
72. end

```

## Apéndice E

### Diferentes clasificadores

```

1. import decimal
2. import itertools
3. import numpy as np
4. from sklearn import svm
5. from sklearn import tree
6. from sklearn import neighbors
7. import matplotlib.pyplot as plt
8. from sklearn.metrics import f1_score
9. from sklearn.preprocessing import StandardScaler
10. from sklearn.neural_network import MLPClassifier
11. from sklearn.model_selection import cross_val_score
12. from sklearn.ensemble import RandomForestClassifier
13. from sklearn.model_selection import train_test_split
14. from sklearn import feature_selection
15. import warnings
16. warnings.filterwarnings("ignore")
17.
18. # Import data
19. # Bitalino Valence
20. originalData = np.loadtxt("Altos_y_Bajos_Valence_Bitalino_sin2.csv", delimiter=",", skiprows=1)[::[0,1,2,3,4,7,8]]

```

```

21. # Bitalino Arousal
22. originalData = np.loadtxt("Altos_y_Bajos_Arousal_Bitalino_sin2.csv", delimiter=",", skiprows=1)[:,[0,1,2,3,8]]
23. # E-Health Valence
24. originalData = np.loadtxt("Altos_y_Bajos_Valence_E-Health_sin2.csv", delimiter=",", skiprows=1)[:,[0,1,2,3,4,5,6,7,8]]
25. # E-Health Arousal
26. originalData = np.loadtxt("Altos_y_Bajos_Arousal_E-Health_sin2.csv", delimiter=",", skiprows=1)[:,[0,1,3,5,7,8]]
27.
28. # Standardization
29. originalData[:,0:-1] = (originalData[:,0:-1]-np.mean(originalData[:,0:-1]))/np.std(originalData[:,0:-1])
30.
31. # Feature selection
32. fselect = feature_selection.mutual_info_classif(originalData[0:,0:-1], originalData[0:-1], discrete_features='auto', n_neighbors=3, copy=False,
random_state=None)
33. print("feature_selection.mutual_info_classif:\n", fselect)
34. fselect+=fselect;
35.
36. data = np.random.permutation(originalData)
37. # Feaures
38. X = data[0:,0:-1]
39. # Class
40. y = data[0,-1]
41.
42. # Separate data into train and test
43. # 80% train, 20% test
44. X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
45.
46. # PCA
47. # from sklearn.decomposition import PCA
48. # pca = PCA(n_components=2) #n_components=2
49. # X_train = pca.fit_transform(X_train)
50. # X_test = pca.transform(X_test)
51. # explained_variance = pca.explained_variance_ratio_
52. # print('hola',explained_variance)
53.
54. # Initialize score list
55. totalScores = [0] * 5
56.
57.
58. def Classify(classifierName):
59.     global count
60.     # Print Classifier name
61.     print(classifierName)
62.     # Fit model
63.     classifier.fit(X_train, y_train)
64.     # Predict test data
65.     y_pred = classifier.predict(X_test)
66.     # 10 fold cross val
67.     score = cross_val_score(classifier,X,y,cv=10,scoring="f1_macro")
68.     # Add this score mean into totalScores
69.     totalScores[count]=score.mean()
70.     # Increase count
71.     count+=1
72.     # Print cross validation score, f1_macro (mean +/- standard deviation)
73.     print("      %0.4f (+/- %0.4f)" % (score.mean(), score.std() * 2))
74.
75.
76. # initialize totalScores count
77. count = 0
78. # RandomForestClassifier
79. classifier = RandomForestClassifier(criterion="entropy", max_depth=6, random_state=4, bootstrap=False)
80. Classify("RandomForest")
81.
82. # SVM rbf
83. classifier = svm.SVC(kernel='rbf', C=0.001)
84. Classify("SVM rbf")
85.
86. # MLP
87. classifier = MLPClassifier(hidden_layer_sizes=(10,), solver='sgd', learning_rate='invscaling')
88. Classify("MLP")
89.
90. # KNN
91. n_neighbors = 5
92. classifier = neighbors.KNeighborsClassifier(n_neighbors, weights='uniform')
93. Classify("KNN")
94.
95. # DecisionTree
96. classifier = tree.DecisionTreeClassifier()
97. Classify("DecisionTree")
98.
99. print("-----\nAverage score:\n",np.mean(totalScores))

```

## Apéndice F - Clasificación Alto, medio y Bajo

Basándose en el trabajo [36], en donde también se clasificaron emociones con los sensores ECG y GSR, la comparación entre sensores fue realizada con las mismas métricas. Estas fueron *10 cross validation* utilizando un clasificador SVM. La Tabla 0-1, muestra los resultados obtenidos en dicho estudio para valencia y Arousal Alto, medio y bajo (alto: valor SAM  $\geq 7$ ; medio:  $4 \leq \text{valor} \leq 6$ ; bajo:  $\text{valor} \leq 3$ ; Para cada valor de SAM).

Tabla 0-1: Ejemplo para comparación entre sensores

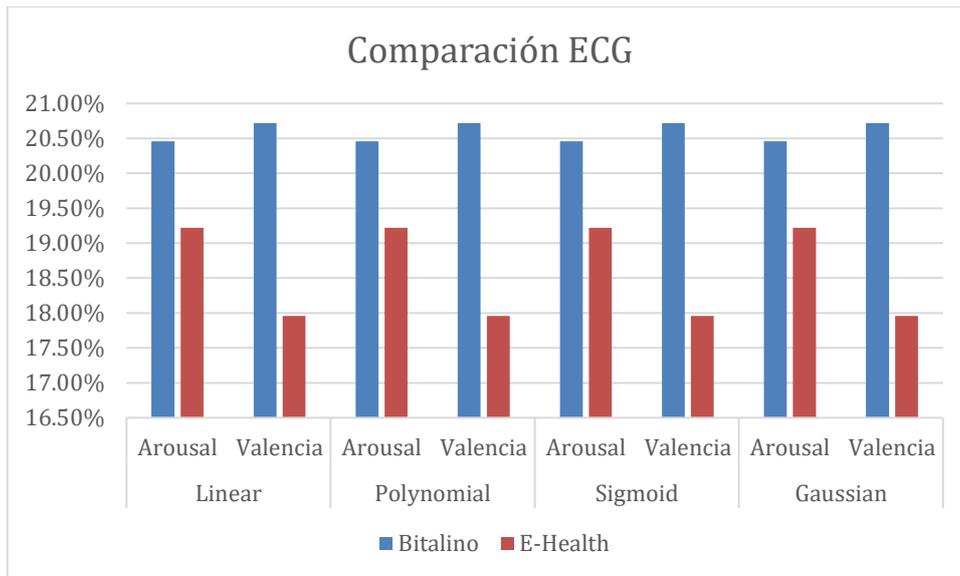
	Linear		Polynomial		Sigmoid		Gaussian	
	Arousal	Valence	Arousal	Valence	Arousal	Valence	Arousal	Valence
ECG	51,4%	44,05%	50%	52,12%	52,63%	46,80%	50,07%	46,8%
GSR	47,36%	48,93%	48,42%	48,93%	49,37%	45,74%	50,52%	47,87%

En la Tabla 0-2 se muestran los resultados obtenidos con nuestro set de datos con las mismas métricas del trabajo anterior:

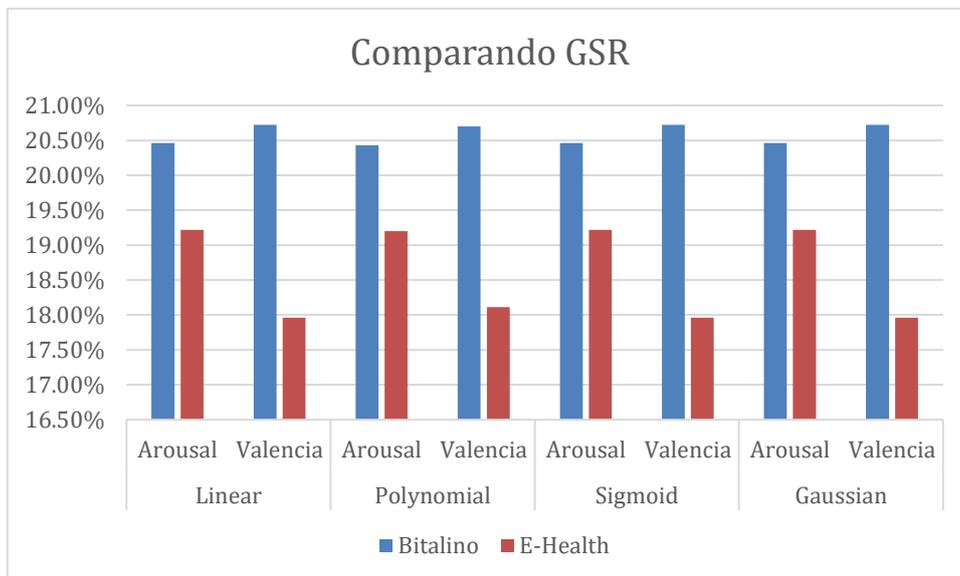
TABLA 0-2: TABLA COMPARATIVA SVM CON DIFERENTES KERNELS

		Linear		Polynomial		Sigmoid		Gaussian	
		Arousal	Valencia	Arousal	Valencia	Arousal	Valencia	Arousal	Valencia
ECG	Bitalino	0.2046 (+/- 0.0016)	0.2072 (+/- 0.0008)	0.2046 (+/- 0.0016)	0.2072 (+/- 0.0008)	0.2046 (+/- 0.0016)	0.2072 (+/- 0.0008)	0.2046 (+/- 0.0016)	0.2072 (+/- 0.0008)
	E-Health	0.1922 (+/- 0.0009)	0.1796 (+/- 0.0006)	0.1922 (+/- 0.0009)	0.1796 (+/- 0.0006)	0.1922 (+/- 0.0009)	0.1796 (+/- 0.0006)	0.1922 (+/- 0.0009)	0.1796 (+/- 0.0006)
GSR	Bitalino	0.2046 (+/- 0.0016)	0.2072 (+/- 0.0008)	0.2043 (+/- 0.0020)	0.2070 (+/- 0.0016)	0.2046 (+/- 0.0016)	0.2072 (+/- 0.0008)	0.2046 (+/- 0.0016)	0.2072 (+/- 0.0008)
	E-Health	0.1922 (+/- 0.0009)	0.1796 (+/- 0.0006)	0.1920 (+/- 0.0023)	0.1811 (+/- 0.0109)	0.1922 (+/- 0.0009)	0.1796 (+/- 0.0006)	0.1922 (+/- 0.0009)	0.1796 (+/- 0.0006)

Los resultados de la tabla anterior, también se muestran en la Figura 0-3 y Figura 0-4, en donde se grafican los porcentajes de clasificación obtenidos comparando un sensor contra otro. Al hacer esto, pareciera obtenerse una ventaja en Bitalino para todas las clasificaciones, pero esta ventaja realmente no es significativa.



**FIGURA 0-3: COMPARACIÓN ECG CON VALORES ALTO, MEDIO Y BAJO**



**FIGURA 0-4: COMPARACIÓN GSR PARA VALORES ALTO, MEDIO Y BAJO**