

**UNIVERSIDAD DEL VALLE DE GUATEMALA**  
**Facultad de Ingeniería**



**EMOTRIX: Plataforma para el desarrollo de tecnología BCI  
enfocada en la identificación y manipulación de emociones  
humanas**

**Trabajo de graduación en modalidad de Megaproyecto presentado por:**

Cesar Aníbal Luis Alvarado y Henzer Ottoniel García Cruz para optar al grado académico de Licenciados en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Jackelin Dolores Guerra Vargas para optar al grado académico de Licenciada en Ingeniería Industrial

José Matías Vásquez para optar al grado de Licenciado en Ingeniería Mecatrónica

**Guatemala,  
2016**



**EMOTRIX: Plataforma para el desarrollo de tecnología BCI  
enfocada en la identificación y manipulación de emociones  
humanas**

**UNIVERSIDAD DEL VALLE DE GUATEMALA**  
**Facultad de Ingeniería**



**EMOTRIX: Plataforma para el desarrollo de tecnología BCI  
enfocada en la identificación y manipulación de emociones  
humanas**

**Trabajo de graduación en modalidad de Megaproyecto presentado por:**

Cesar Aníbal Luis Alvarado y Henzer Ottoniel García Cruz para optar al grado académico de Licenciados en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

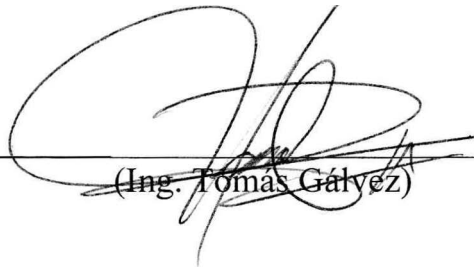
Jackelin Dolores Guerra Vargas para optar al grado académico de Licenciada en Ingeniería Industrial

José Matías Vásquez para optar al grado de Licenciado en Ingeniería Mecatrónica

**Guatemala,  
2016**

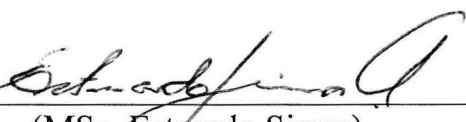


Vo.Bo.

(f)   
(Ing. Tomás Gálvez)

(f)   
(MSc. Douglas Barrios)

(f)   
(MSc. Carlos Esquit)

(f)   
(MSc. Estuardo Sierra)

Fecha de Aprobación: Guatemala 01 de diciembre del 2016

## **PREFACIO**

El reconocimiento de emociones fue, es y será una de las ramas de estudio que la ciencia ha estudiado a lo largo de los años hasta encontrar la mejor manera de la detección de las mismas. El nombre que recibe este proyecto se debe a que, en principio, se quiere utilizar como referencia el dispositivo EMOTIV EPOC para la detección de las emociones, como una base y un ejemplo a seguir para la elaboración del proyecto EMOTRIX. Luego, la última parte del nombre viene dada de la famosa expresión en Guatemala, “A la tortrix”, haciendo referencia a una forma en que los guatemaltecos se distinguen de las demás personas alrededor del mundo al momento de realizar acciones que sólo los guatemaltecos saben y conocen. Al unirlos, se desea dar la intención de que en Guatemala se puede desarrollar software para el estudio y reconocimiento de emociones.

El Proyecto EMOTRIX se ha desarrollado durante el transcurso del año 2016. La primera parte del proyecto se desarrolló en los primeros seis meses del año en la cual se realizó la investigación y la planeación del proyecto. En esta fase se determinó el alcance, el enfoque general del proyecto, la metodología a trabajar en el mismo y las bases teóricas del proyecto. En la segunda fase, a partir del mes de julio, se terminaron las investigaciones teóricas y se empezó la implementación práctica del proyecto. En esta fase se desarrollaron los dispositivos de hardware a utilizar, así como los módulos de software que se establecieron desarrollar en la fase inicial. Al tener estas fases realizadas, se realizaron los trabajos correspondientes de los mismos y los reportes finales respectivamente.

# ÍNDICE

PREFACIO .....	vii
INDICE .....	viii
LISTADO DE FIGURAS .....	xv
LISTADO DE CUADROS .....	xxii
RESUMEN.....	xxiv
I. INTRODUCCIÓN .....	1
II. OBJETIVOS .....	2
A. Objetivos generales.....	2
B. Objetivos específicos por módulo .....	2
1. Módulo de Procesamiento y clasificación de señales para interfaz brazo computadora. ....	2
2. Módulo Procesamiento de señales encefalograficas para interfaz cerebro computador. ....	2
3. Módulo Comunicación con casco y pulsera para recepción y almacenamiento de datos.....	2
4. Módulo Desarrollo de SDK enfocado en el procesamiento de señales EEG para la .....	
clasificación y reconocimiento de emociones .....	2
5. Módulo Diseño e implementación de la plataforma Figura para Emotrix.....	2
6. Módulo Administrar los recursos en el desarrollo de la plataforma de tecnología Brain.....	
Computer Interface, registrar sistemáticamente con métricas el avance del proyecto .....	
y proporcionar la información mediante un análisis de costos para fundamentar la toma de .....	
decisiones en las siguientes fases de Emotrix.....	3
III. JUSTIFICACIÓN .....	4
IV. MÓDULO 1: Procesamiento de señales encefalograficas para interfaz cerebro-computador .....	5
A. Marco teórico.....	5
1. Anatomía del encéfalo .....	5

2.	Secciones corteza cerebral .....	7
3.	Electroencefalografía .....	8
4.	Clasificación señales EEG .....	12
5.	Filtrado de señales eléctricas .....	13
6.	Filtros analógicos .....	15
7.	Filtro digitales .....	20
8.	Impresoras 3D .....	26
9.	Amplificador instrumental .....	26
10.	DSPIC30F4013 .....	27
B.	Marco metodológico .....	28
1.	Amplificación de señales EEG .....	28
2.	Construcción de casco.....	29
3.	Filtrado digital de las señales EEG .....	30
C.	Resultados .....	30
1.	Amplificación y acondicionamiento de señales EEG .....	30
2.	Diseño y fabricación del casco .....	41
3.	Filtrado de señales.....	48
D.	Discusión .....	58
E.	Conclusiones.....	61
F.	Recomendaciones .....	61
V.	MÓDULO 2: procesamiento y clasificación de señales para interfaz brazo computadora .....	62
A.	Marco teórico.....	62
1.	Electromiografía .....	62
2.	Electrocardiograma .....	65
3.	Herramienta de análisis y diseño de filtros FDATool .....	66
4.	Herramienta para aprendizaje de clasificación Classification Learner .....	68

B.	Marco metodológico.....	69
1.	Obtención de señales EMG y EKG .....	69
2.	Procesamiento de señales EMG.....	73
3.	Procesamiento de señales EKG .....	75
4.	Clasificación de señales EMG .....	79
C.	Resultados.....	91
D.	Discusión .....	94
E.	Conclusiones.....	95
F.	Recomendaciones .....	96
VI.	MÓDULO 3: Comunicación con casco y pulsera para recepción y almacenamiento de datos.....	97
A.	Marco teórico.....	97
1.	Electroencefalografía y electromiografía .....	97
2.	Los artefactos .....	98
3.	Análisis de señal EEG y EMG con estadísticas .....	98
4.	PySerial .....	99
5.	Arduino y Arduino nano .....	100
6.	MongoDB.....	101
B.	Antecedentes.....	102
C.	Marco metodológico.....	102
1.	Proceso general de Emotrix .....	102
2.	Herramientas .....	103
3.	Simulación de dispositivos con el Arduino .....	104
4.	Lectura de los datos .....	108
D.	Resultados.....	111
1.	Funciones del casco para la API.....	111
2.	Funciones de la pulsera para la API .....	113

3.	Prueba de conectividad .....	114
4.	Lecturas del casco .....	115
5.	Lecturas de la pulsera.....	117
6.	Prueba de checksum.....	119
7.	Prueba de persistencia de datos .....	120
E.	Discusión .....	121
F.	Conclusiones.....	123
G.	Recomendaciones .....	124
VII.	MÓDULO 4: desarrollo de sdk enfocado en el procesamiento de señales eeg para la .....	
	clasificación y el reconocimiento de emociones .....	125
A.	Marco teórico.....	125
1.	Emociones.....	125
2.	Estímulo .....	126
3.	Aprendizaje de máquina .....	127
4.	Métodos de evaluación de un sistema .....	128
5.	Señales electroencefalográficas (EEG).....	128
6.	Pre-procesamiento de una señal.....	129
7.	Extracción de características.....	129
8.	Transformada de Wavelets .....	130
9.	Algoritmo de máquina de soporte vectorial .....	130
10.	Electrodos.....	131
11.	Sistema internacional 10-20.....	131
12.	Kit de desarrollo de software (SDK) .....	131
13.	API .....	132
B.	Marco metodológico.....	132
1.	Investigación y selección de tecnologías.....	132

2.	Desarrollo del API de Emotrix .....	132
3.	Toma de datos .....	133
4.	Pre-procesamiento de los datos.....	133
5.	Definición del conjunto de datos de entrenamiento .....	134
6.	Implementación del algoritmo de clasificación.....	134
7.	Proceso de selección de características.....	135
8.	Implementación del reconocimiento en tiempo real .....	135
C.	Resultados .....	135
1.	Estructura del API de Emotrix.....	135
2.	Pre-procesamiento de los datos.....	136
3.	Eficacia del algoritmo .....	138
D.	Discusión .....	141
E.	Conclusiones.....	143
F.	Recomendaciones .....	144
VIII.	MÓDULO 5: Diseño y desarrollo de la plataforma Figura para emotrix .....	145
A.	Marco teórico.....	145
1.	Python .....	145
2.	WxPython.....	145
3.	Aplicación .....	145
4.	Interfaz .....	145
5.	¿Cómo funciona EMOTIV?.....	146
6.	Interacción humano computador .....	147
7.	Interfaz cerebro computador .....	150
B.	Marco metodológico.....	151
1.	Investigación .....	151
2.	Diseño .....	153

C.	Resultados.....	165
1.	Entrevistas con el prototipo de papel.....	165
2.	Entrevistas de usabilidad con Mockup .....	167
3.	Entrevistas de usabilidad con la aplicación .....	173
4.	Vídeo de las entrevistas .....	178
D.	Discusión .....	179
E.	Conclusiones.....	181
F.	Recomendaciones .....	181
IX.	MÓDULO 6: administrar los recursos en el desarrollo de la plataforma de tecnología brain .....	
	computer interface, registrar sistemáticamente con métricas el avance del proyecto y proporcionar .....	
	la información mediante un análisis de costos para fundamentar la toma de decisiones en las siguientes .....	
	fases de emotrix.....	182
A.	Marco teórico.....	182
1.	Gestión de proyectos.....	182
2.	Gestión de proyectos de software .....	183
3.	Calendarización del proyecto.....	187
4.	Administración del riesgo .....	187
5.	Documentación de proyectos.....	187
6.	Mediciones en producto y proceso de software.....	188
B.	Marco metodológico .....	191
1.	Iniciación.....	191
2.	Planificación.....	191
3.	Ejecución.....	192
4.	Seguimiento y control.....	192
5.	Cierre.....	192
C.	Resultados.....	192

1.	Administración de Emotrix.....	192
2.	Estimación de costos.....	194
3.	Manejo de riesgos .....	196
4.	Métricas.....	198
D.	Conclusiones.....	201
E.	Recomendaciones .....	201
X.	Bibliografía .....	204
XI.	Anexos.....	209
A.	Módulo: Comunicación con casco y pulsera para recepción y almacenamiento de datos .....	209
B.	Módulo: Diseño y desarrollo de la plataforma Figura para EMOTRIX.....	210
C.	Modulo Administrar los recursos en el desarrollo de la plataforma de tecnología Brain Computer Interface, registrar sistemáticamente con métricas el avance del proyecto y proporcionar la información mediante un análisis de costos para fundamentar la toma de decisiones en las siguientes fases de Emotrix	
	215	
1.	Seguimiento y control de las actividades .....	215
2.	Seguimiento vía Asana .....	218
3.	Plantillas para minutas .....	221

## LISTADO DE FIGURAS

Figura 1. Partes principales del encéfalo .....	5
Figura 2. La corteza cerebral .....	6
Figura 3. División de corteza cerebral .....	7
Figura 4. Ejemplo de posicionamiento de electrodos según el sistema 10-20, colocados de forma vertical .... de inion a nasion .....	9
Figura 5. Ejemplo de posicionamiento de electrodos según el sistema 10-20, colocados de forma ..... horizontal de inion a nasion. ....	10
Figura 6. Ejemplo de posicionamiento de electrodos según el sistema 10-20, colocados de forma vertical .... de oreja a oreja .....	10
Figura 7. Sistema de posicionamiento 10-20 completo.....	11
Figura 8. Sistema de posicionamiento 10-10 completo.....	12
Figura 9. Respuesta en magnitud ideal para un filtro pasa baja .....	14
Figura 10. Respuesta en magnitud ideal para un filtro pasa alta .....	14
Figura 11. Respuesta en magnitud ideal para un filtro pasa banda .....	14
Figura 12. Respuesta en magnitud ideal para un filtro rechaza banda .....	15
Figura 13. Topología filtro Sallen-Key pasa baja.....	16
Figura 14. Ganancia en frecuencia de filtro Butterworth pasa baja de orden 10. ....	17
Figura 15. Ganancia en frecuencia de filtro Chevyshev pasa baja de orden 10.....	17
Figura 16. Ganancia en frecuencia de filtro Elíptico pasa baja de orden 10.....	18
Figura 17. Esquema del amplificador operacional .....	19
Figura 18. Símbolo esquemático del amplificador operacional .....	20
Figura 19. Componentes para implementación de filtro digital .....	20
Figura 20. Ejemplo de una señal muestreada de forma correcta, y una con el efecto ..... de Aliasing .....	21
Figura 21. Ejemplo de señal digitalizada.....	22
Figura 22. Estructura general de filtro digital.....	23
Figura 23. Estructura general filtro digital FIR .....	23
Figura 24. Diagrama de bloques de filtro FIR.....	23
Figura 25. Estructura de filtro IIR.....	24
Figura 26. Diagrama de bloques de filtro IIR.....	25

Figura 27. Aproximación bilineal .....	25
Figura 28. Topología amplificador instrumental .....	26
Figura 29. Diagrama de pines de DSPIC30F4013 .....	27
Figura 30. Esquemático de amplificador Instrumental con ganancia 15810. ....	31
Figura 31. Señal de entrada al amplificador instrumental, Señal amplificada .....	
por el amplificador instrumental con ganancia 15810.....	32
Figura 32. Circuito esquemático de filtro Antialias. ....	33
Figura 33. Respuesta en frecuencia del filtro Antialias. ....	34
Figura 34. Circuito esquemático de filtro pasa alta. ....	34
Figura 35. Respuesta en frecuencia de filtro pasa alta con frecuencia de corte de 0.33Hz. ....	35
Figura 36. Circuito esquemático de amplificador inversor con ganancia máxima de 50. ....	36
Figura 37. Etapa completa de acondicionamiento. ....	36
Figura 38. Esquemático completo de circuito de amplificación y acondicionamiento de las señales EEG. .	37
Figura 39. Diseño final PCB. ....	38
Figura 40. Placa de amplificación de señales EEG. ....	38
Figura 41. Señal de electrodo F4, después de ser amplificada por el amplificador instrumental. ....	39
Figura 42. Señal amplificada luego del filtro pasa baja. ....	39
Figura 43. Señal luego del filtro pasa alta. ....	40
Figura 44. Señal después de amplificarla en la última etapa. ....	40
Figura 45. Frecuencias que componen la señal amplificada final. ....	41
Figura 46. Sketch original de diadema horizontal. ....	42
Figura 47. Sketch de diadema horizontal modificado para encajar mejor con la cabeza. ....	42
Figura 48. Vista frontal lateral izquierda desde arriba de pieza central. ....	43
Figura 49. Vista trasera lateral de pieza central. ....	43
Figura 50. Vista frontal lateral derecha desde arriba de pieza del lado derecho. ....	44
Figura 51. Vista trasera lateral izquierda desde arriba de pieza del lado derecho. ....	44
Figura 52. Vista frontal lateral izquierda desde arriba de pieza del lado izquierdo. ....	45
Figura 53. Vista trasera lateral derecha desde arriba de pieza del lado izquierdo. ....	45
Figura 54. Vista frontal lateral derecha de la diadema para colocar los electrodos F3 y F4. ....	46
Figura 55. Ensamble de piezas de casco. ....	46
Figura 56. Diadema horizontal. ....	47
Figura 57. casco completo .....	47
Figura 58. casco colocado sobre persona.....	48

Figura 59. Herramienta de MikroC para diseño de filtros IIR. ....	49
Figura 60. Parámetros acogidos para filtro IIR a implementar. ....	49
Figura 61. Funcion de transferencia de filtro a implementar. ....	50
Figura 62. Respuesta en frecuencia de filtro a implementar. ....	51
Figura 63. Coeficientes de filtro escalados y su factor de escalamiento. ....	51
Figura 64. Ecuación de diferencias implementada por la función "IIR_Radix" de MikroC. ....	51
Figura 65. Función que implementa filtro IIR, y sus parámetros de entrada. ....	52
Figura 66. Señal de 60 Hz de 1 Voltio de amplitud sumada a señal de 22 Hz de 0.5 Voltios de amplitud... 53	53
Figura 67. Señal de 60 Hz de 1 Voltio de amplitud sumada a señal de 22 Hz de 0.1 Voltios de amplitud... 53	53
Figura 68. Señal de 60 Hz de 1 Voltio de amplitud sumada a señal de 22 Hz de 1 Voltio de amplitud. ....	54
Figura 69. Respuesta del filtro de orden 5 a la señal de prueba 1. ....	54
Figura 70. Respuesta del filtro de orden 15 a la señal de prueba 1. ....	55
Figura 71. Respuesta del filtro de orden 25 a la señal de prueba 1. ....	55
Figura 72. Respuesta del filtro de orden 5 a la señal de prueba 2. ....	55
Figura 73. Respuesta del filtro de orden 15 a la señal de prueba 2. ....	56
Figura 74. Respuesta del filtro de orden 25 a la señal de prueba 2. ....	56
Figura 75. Respuesta del filtro de orden 5 a la señal de prueba 3. ....	57
Figura 76. Respuesta del filtro de orden 15 a la señal de prueba 3. ....	57
Figura 77. Respuesta del filtro de orden 25 a la señal de prueba 3. ....	57
Figura 78. Señal EEG proveniente de electrodo F4, digitalizada y filtrada. ....	58
Figura 79. Reducción de señal por tejido adiposo.....	63
Figura 80. Señales ECG, EMG con y sin ruido ECG.....	64
Figura 81. señal de un electrocardiograma.....	65
Figura 82. Partes del corazón.....	66
Figura 83. Herramienta de análisis y diseño de filtros.....	67
Figura 84. Filtro anti alias.....	69
Figura 85. Filtro pasa alta Filtrado de componente DC.....	70
Figura 86. Filtro tipo "notch" a 60 Hz.....	71
Figura 87. Proceso de filtrado de señal para muestreo analógico-digital.....	72
Figura 88. Diseño conceptual de pulsera.....	73
Figura 89. filtro notch, comportamiento de señales a distintas frecuencias.....	74
Figura 90. Filtro "notch" en frecuencias mayores a la frecuencia de muestreo.....	75
Figura 91. Filtro pasa baja con frecuencia de corte a 4Hz tipo IIR.....	75

Figura 92. Filtro pasa baja con frecuencia de corte a 4Hz tipo FIR.....	76
Figura 93. Transformada rápida de Fourier.....	77
Figura 94. Señal ECG junto con su transformada de Fourier y acercamientos..... en las frecuencias de interés.....	79
Figura 95. Señal electromiográfica de mujer #1 sosteniendo un cilindro .....	80
Figura 96. Detalle de señal electromiográfica de mujer #1 sosteniendo un cilindro .....	80
Figura 97. Señal electromiográfica de mujer #1 sosteniendo una esfera .....	81
Figura 98. Señal electromiográfica de mujer #1 haciendo un gancho con la mano.....	81
Figura 99. Señal electromiográfica de mujer #1 sosteniendo un objeto plano de forma lateral.....	82
Figura 100. Señal electromiográfica de mujer #1 sosteniendo un objeto con la punta de los .....	82
dedos índice y pulgar .....	82
Figura 101. Señal electromiográfica de mujer #1 sosteniendo un bolígrafo.....	82
Figura 102. Comparación de señales por gesto para mujer #1 .....	83
Figura 103. Promedio de 60 señales por gesto de mujer #1 .....	84
Figura 104. Desviación estándar por muestra.....	84
Figura 105. Transformada de Fourier de todas las señales correspondientes a la mujer #1 .....	85
Figura 106. Densidad espectral de potencia de cada señal para mujer #1 .....	86
Figura 107. Comparación de transformada de Fourier con densidad espectral de potencia.....	86
Figura 108. Correlación de señales en las columnas 2 y 3 de la matriz de densidad espectral de potencia..	88
Figura 109. Matriz de confusión para máquina de soporte vectorial para análisis en el dominio de .....	89
densidad espectral de potencia.....	89
Figura 110. Región de convergencia para los gestos 3 y 5 con análisis de densidad espectral de potencia..... por medio de SVM.....	90
Figura 111: Iteración 1 de interfaz Figura para reconocimiento de gestos con FFT .....	91
Figura 112: Iteración 2 de interfaz Figura para reconocimiento de gestos con FFT .....	92
Figura 113. Iteración 3 de interfaz Figura para reconocimiento de gestos con FFT.....	92
Figura 114. Iteración 1 de interfaz Figura para reconocimiento de gestos con PSD .....	93
Figura 115. Iteración 2 de interfaz Figura para reconocimiento de gestos con PSD .....	93
Figura 116. Iteración 3 de interfaz Figura para reconocimiento de gestos con PSD .....	93
Figura 117. ECG mostrando los picos sobre el límite inferior de 0.4 mV.....	94
Figura 118. Clasificación de las señales EEG .....	97
Figura 119. Arduino Nano .....	101
Figura 120. Fases principales de Emotrix.....	103

Figura 121. Proceso de conversión de las lecturas del casco a 2 caracteres .....	105
Figura 122. Ejemplo de conversión de una lectura del casco a 2 caracteres.....	105
Figura 123. Prueba de concepto para la recepción de datos utilizando PySerial .....	109
Figura 124. Proceso general para la simulación del casco y la recepción de los datos.....	111
Figura 125. Proceso general para la simulación de la pulsera y la recepción de los datos .....	111
Figura 126. Programa de Python para prueba de conectividad .....	114
Figura 127. Output de conectividad exitosa .....	114
Figura 128. Output de conexión fallida .....	114
Figura 129. Programa de Python para la lectura del casco.....	115
Figura 130. Output de prueba de buena señal del casco.....	116
Figura 131. Output de prueba de mala señal del casco .....	116
Figura 132. Output de prueba de ausencia de señal del casco.....	117
Figura 133. Programa de Python para la lectura de la pulsera .....	117
Figura 134. Output de prueba de buena señal de la pulsera .....	118
Figura 135. Output de prueba de mala señal de la pulsera .....	118
Figura 136. Output de prueba de ausencia de señal de la pulsera .....	119
Figura 137. Output de alteración del checksum para el casco .....	119
Figura 138. Lecturas del casco almacenados en MongoDB.....	120
Figura 139. Lecturas de la pulsera almacenadas en MongoDB .....	120
Figura 140. Modelo dimensional de emociones .....	126
Figura 141. Proceso de adquisición de emociones .....	126
Figura 142. Sistema Internacional 10-20. Distribución de 10s electrodos sobre la superficie de la ..... cabeza con las posiciones básicas obtenidas a partir del 10% y 20% de la medida total del cráneo.....	131
Figura 143. Metodología de trabajo representada en un diagrama de flujo. ....	132
Figura 144. Procedimiento para la exposición de los estímulos visuales .....	133
Figura 145. División de cada bloque de datos en segmentos de 1 segundo.....	134
Figura 146. Estructura del API de Emotrix .....	135
Figura 147. Señales EEG para los electrodos F3, F4, AF3, AF4 del sujeto #5. ....	136
Figura 148. Señal EEG contenida en un segmento de 1 segundo de duración (128 muestras). ....	136
Figura 149. Comparación de las señales EEG, para los electrodos F3, F4, AF3, AF4. Segmento ..... etiquetado como tristeza=rojo, alegría = azul y neutral = negro. ....	137
Figura 150. Aplicación de la transformada de Wavelet sobre el electrodo AF4, con 4 niveles ..... de descomposición, utilizando una wavelet madre Daubechies 4 (db4), para separar las distintas bandas .....	

de frecuencia de las señales EEG. ....	137
Figura 151. Obtención de la densidad espectral de potencia, luego de haber sido aplicada la transformada Wavelet en el electrodo AF4, para el sujeto #5. ....	138
Figura 152. Tipos de onda detectados por electrodos ....	147
Figura 153. Esquema general de la tecnología BCI ....	151
Figura 154. Primer prototipo en papel de la interfaz de la aplicación.....	154
Figura 155. Mockup de inicio de la aplicación.....	156
Figura 156. Mockup de la interfaz del casco.....	156
Figura 157. Mockup de interfaz de la pulsera ....	157
Figura 158. Mockup de la interfaz de la ventana de instrucciones ....	157
Figura 159. Mockup de la interfaz de instrucciones del casco.....	158
Figura 160. Mockup de la interfaz de instrucciones del casco.....	158
Figura 161. Mockup de la interfaz del demo de la aplicación.....	159
Figura 162. Mockup de la interfaz de la sección de ayuda.....	159
Figura 163. Mockup de la interfaz de la ventana acerca de ....	160
Figura 164. Ventana inicial de la aplicación ....	161
Figura 165. Ventana general de instrucciones.....	161
Figura 166. Ventana de instrucciones sobre el casco de EMOTRIX.....	162
Figura 167. Ventana de instrucciones sobre la pulsera de EMOTRIX.....	162
Figura 168. Ventana del casco de EMOTRIX.....	163
Figura 169. Ventana de la pulsera de EMOTRIX.....	163
Figura 170. Ventana de información de ayuda al usuario.....	164
Figura 171. Ventana de información sobre EMOTRIX.....	164
Figura 172. Ventana de demo de EMOTRIX.....	165
Figura 173. Menú inicial agregado.....	166
Figura 174. Cambios realizados a la interfaz del casco.....	166
Figura 175. Cambios realizados a la interfaz de la pulsera.....	167
Figura 176. Salón utilizado para entrevistas de Usability Testing.....	174
Figura 177. Materiales utilizados para entrevistas de Usability Testing.....	174
Figura 178. Iteraciones de Scrum.....	184
Figura 179. Desarrollo de Lean Development.....	184
Figura 180. Planificación de Extreme Programming.....	185
Figura 181. DSDM metodología.....	185

Figura 182. Desarrollo con Adaptive Software Development .....	186
Figura 183. Adecuar el método correcto de Crystal family .....	186
Figura 184. Salarios para programadores de aplicaciones sin experiencia en Guatemala.....	195
Figura 185. Salarios para programador de sitios web sin experiencia en Guatemala.....	195
Figura 186. Métricas extraídas de Asana.....	198
Figura 187. Métricas modificables en tiempo real extraídas del 5 de Octubre.....	199
Figura 188. Desarrollo de métricas de los prototipos.....	199
Figura 189. Métricas de los avances del proyecto por objetivos.....	199
Figura 190. Detalle de los avances hasta el 20 de Octubre .....	200
Figura 191. Código de Arduino utilizado para pruebas de comunicación por bluetooth .....	209
Figura 192. Código utilizado para buscar dispositivos de bluetooth con PyBluez .....	209
Figura 193. Programa utilizado para pruebas de comunicación por bluetooth con PyBluez .....	210
Figura 194. Gantt Megaproyecto para la fase de planificación.....	215
Figura 195. Gantt del Módulo del desarrollo del Software Development Kit .....	215
Figura 196. Gantt del Módulo del casco de electrodos .....	216
Figura 197. Gantt del Módulo de la pulsera de señales electromiográficas.....	216
Figura 198. Gantt del Módulo de la comunicación .....	217
Figura 199. Gantt Módulo de gestión del proyecto .....	217
Figura 200. Gestión y seguimiento de las actividades en la plataforma de Asana .....	218
Figura 201. Gestión y seguimiento de las actividades en la plataforma de Asana .....	219
Figura 202. Líneas de código desarrolladas y eliminadas extraído de GitHub.....	220

## LISTADO DE CUADROS

Cuadro 1. Tiempo requerido para transformaciones de Fourier.....	78
Cuadro 2. Matriz de datos para entrenamiento de máquina de aprendizaje clasificadora .....	87
Cuadro 3. Rendimiento de algoritmos clasificadores entrenados .....	88
Cuadro 4. Rangos de frecuencias que componene las señales EEG .....	129
Cuadro 5. Eficacia del clasificador SVM, con tamaños de ventana de 128 (1 seg) y 256 (2 seg), sin traslape. ....	139
Cuadro 6. Eficacia del clasificador SVM, con tamaños de ventana de 128 (1 seg) y 256 (2 seg), con traslape. ....	139
Cuadro 7. Eficacia del clasificador SVM, sin un límite de calidad permitido. ....	140
Cuadro 8. Eficacia del clasificador SVM, con un límite de calidad de la lectura mayor a 50% .....	140
Cuadro 9. Eficacia del clasificador SVM, con un límite de calidad de la lectura mayor a 70% .....	140
Cuadro 10. Algunas plataformas BCI existentes .....	152
Cuadro 11. Resultados obtenidos en la tarea no.1 en la entrevista con Mockup .....	167
Cuadro 12. Resultados obtenidos en la tarea no.2 en la entrevista con mockup.....	168
Cuadro 13. Resultados obtenidos de la tarea no.3 en la entrevista con mockup.....	169
Cuadro 14. Resultados obtenidos de la tarea no.4 en la entrevista con mockup.....	169
Cuadro 15. Resultados obtenidos de la tarea no.5 en la entrevista con mockup.....	170
Cuadro 16. Resultados obtenidos de la tarea no.6 en la entrevista con mockup.....	170
Cuadro 17. Resultados obtenidos de la tarea no.7 en la entrevista con mockup.....	171
Cuadro 18. Resultados obtenidos de la tarea no.8 en la entrevista con mockup.....	172
Cuadro 19. Información general de los entrevistados en la entrevista con mockup .....	172
Cuadro 20. Cantidad de personas según su grado académico en la entrevista con mockup.....	172
Cuadro 21. Experiencia de los entrevistados en sistemas relacionados en la entrevista con mockup.....	173
Cuadro 22. Tiempo promedio de los entrevistados por cada tarea realizada en la entrevista con mockup. ....	173
Cuadro 23. Resultados obtenidos en la tarea no.1 en la entrevista con la aplicación .....	175
Cuadro 24. Resultados obtenidos en la tarea no.2 en la entrevista con la aplicación .....	175
Cuadro 25. Resultados obtenidos de la tarea no.3 en la entrevista con la aplicación .....	175
Cuadro 26. Resultados obtenidos de la tarea no.4 en la entrevista con la aplicación .....	176
Cuadro 27. Resultados obtenidos de la tarea no.5 en la entrevista con la aplicación .....	176
Cuadro 28. Resultados obtenidos de la tarea no.6 en la entrevista con la aplicación .....	176
Cuadro 29. Resultados obtenidos de la tarea no.7 en la entrevista con la aplicación .....	177
Cuadro 30. Resultados obtenidos de la tarea no.8 en la entrevista con la aplicación .....	177

Cuadro 31. Información general de los entrevistados en la entrevista con la aplicación.....	177
Cuadro 32. Cantidad de personas según su grado académico en la entrevista con la aplicación .....	178
Cuadro 33. Experiencia de los entrevistados en sistemas relacionados en la entrevista con la aplicación .	178
Cuadro 34. Tiempo promedio de los entrevistados por cada tarea realizada en la entrevista con la .....	178
aplicación .....	178
Cuadro 35. Dificultad de un proyecto en función de sus líneas de código .....	189
Cuadro 36. Ecuaciones básicas del modelo Integrado .....	190
Cuadro 37. Recurso Humano para Emotrix.....	193
Cuadro 38. Consolidación del Gantt general e enero a noviembre de 2016 .....	193
Cuadro 39. Selección del modelo constructivo de costos .....	194
Cuadro 40. Estimaciones del modelo constructivo de costos básico .....	194
Cuadro 41. Tiempo de desarrollo estimado total.....	195
Cuadro 42. Tipo de cambio de (Dólares de EE.UU. a Quetzales de enero de 2016 .....	195
Cuadro 43. Costo de mano de obra del desarrollo de la plataforma de Emotrix .....	195
Cuadro 44. Costo de materiales utilizados en el desarrollo de prototipos de Emotrix .....	196
Cuadro 45. Consolidación de los costos incurridos y estimados para Emotrix .....	196
Cuadro 46. Identificación, probabilidad y ponderación de los riesgos .....	197
Cuadro 47. Plan de respuesta a los riesgos de desarrollo .....	197
Cuadro 48. Definición de las métricas para cada módulo .....	198

## RESUMEN

El objetivo principal del proyecto Emotrix es proveer una plataforma abierta e integrable con otras tecnologías, enfocada en el reconocimiento y estimulación de emociones humanas, el cual incluye desde el diseño hasta el prototipo de dispositivos de hardware capaces de recopilar información necesaria para la detección de emociones. Además de un kit de herramientas de desarrollo de software (SDK), que permitirán a los desarrolladores implementar dicha plataforma.

Se diseñó y fabricó un casco adaptable a diferentes personas, el cual contiene los circuitos necesarios para amplificar y acondicionar las señales electroencefalográficas para su procesamiento por medio de un microcontrolador DSPIC30F4013 utilizando una frecuencia de muestreo de 1kHz, y así, enviar las señales por un puerto serial a una tasa de transmisión de 425k baudios. También, se diseñó y fabricó un prototipo con electrodos y otros dispositivos para captar impulsos electromiográficos (EMG), signos vitales como el ritmo cardiaco (EKG) y el posicionamiento angular del brazo en un principio. Estas señales se utilizaron en el software de inteligencia emocional artificial para determinar si hay algún factor que ayude al momento de establecer un parámetro sobre las emociones de los individuos.

Para lograr la detección de dos emociones básicas: felicidad y tristeza se utilizó las señales electroencefalográficas (EEG) producidas en una parte específica del cerebro junto con la descripción del análisis matemático y los algoritmos de aprendizaje de máquina empleados en el proceso; con los cuales se logró un porcentaje de clasificación del 74% de efectividad.

Por otro lado, para la conectividad con los dispositivos y la recepción de los datos tomados por los mismos se implementó dos canales de comunicación serial a través del puerto USB de manera que cada dispositivo tenga su propio canal de transmisión. Del lado de la computadora se reciben los datos mediante la librería PySerial, se aplica un filtro para no dejar pasar datos fuera de rango y se almacena de una base de datos MongoDB.

De la misma manera, se presentan los resultados obtenidos de la implementación de la metodología Interacción Humano Computador para el diseño de la interfaz Figura del proyecto. Se muestra el proceso de entrevistas y la prototipación de los diseños para la interfaz.

La organización y gestión del proyecto se realizó aplicando e integrando la metodología Crystal Clear la cual se adaptaba al tipo de desarrollo de proyectos que implicaba Emotrix. El cálculo de mano de obra según el Modelo Constructivo fue de Q 88,730.82 y los costos totales incurridos en materiales y

requerimientos ascendieron a Q 1,425.88, lo que resulta en una consolidación de costos totales de Q 90,156.70. Al momento de hacer la evaluación de riesgos el subestimar el tiempo de desarrollo de la plataforma resultó con una probabilidad Alta a suceder y con un impacto mayor. Las métricas de tiempo que se establecieron permitieron la visualización del avance en el desarrollo de cada módulo del proyecto.

# I. INTRODUCCIÓN

El proyecto Emotrix es una plataforma que permitirá el estudio, identificación y manipulación de emociones humanas. La plataforma se podrá utilizar para fines educativos, médicos o recreacionales, entre otros. A la fecha, existen distintos dispositivos que permiten imitar inteligencia emocional, los cuales tienen precios elevados. Es por esto que la plataforma tiene un enfoque open-source para permitir el desarrollo de forma abierta y evitar cualquier tipo de restricciones a los usuarios. El ámbito al que pertenece el proyecto es a la interacción humano – computador (HCI).

Como es de esperarse, para poder determinar el estado emocional de una persona se necesita algún tipo de interfaz humano-máquina que permita realizar una toma de datos. En algunos estudios, se utiliza la tecnología computer vision que permite al computador comprender lo que sucede en el mundo real. En este entorno se podría utilizar para observar micro expresiones faciales, movimiento ocular o cualquier otro patrón de comportamiento que ayude a determinar el estado de la persona. Otra opción es por medio de electrodos posicionados en la cabeza para realizar un electroencefalograma.

Existen varios factores que pueden ayudar a determinar el estado de ánimo de una persona, como su presión arterial, ritmo cardíaco, sudoración, alcalinidad y conductividad eléctrica de la piel, entre otros. En este estudio se utilizarán dos dispositivos físicos para la obtención de datos, los cuales serán un casco para lecturas electroencefalográficas y una pulsera para diversas señales, entre ellas, pulso cardíaco, gestos de la mano y posición angular del brazo que se ingresarán al software. Ambos dispositivos son no invasivos y como se mencionó anteriormente, con un enfoque open-source.

## II. OBJETIVOS

### A. Objetivos generales

Proveer al mundo una plataforma abierta e integrable con otras tecnologías, para uso y estimulación de emociones.

### B. Objetivos específicos por módulo

1. Módulo de procesamiento y clasificación de señales para interfaz brazo computadora. Realizar un brazalete que sea capaz de recolectar distintas señales (EMG, ECG y Sensores acelerómetro y giroscopio), filtrarlas, analizarlas y enviar la información relevante al módulo de recepción.

2. Módulo procesamiento de señales encefalográficas para interfaz cerebro computador. Realizar un casco que contenga electrodos, para la recepción de señales encefalográficas, para luego estas mismas señales poderlas filtrar digitalmente y enviarlas a la computadora.

3. Módulo comunicación con casco y pulsera para recepción y almacenamiento de datos. Implementar la interfaz de intercomunicación serial entre los dispositivos de entrada y el API de Emotrix. Proveer a la comunidad una API que permita la comunicación con cualquier dispositivo que implemente el protocolo de comunicación de Emotrix. Proveer un manual de usuario donde se detalle el protocolo de comunicación serial utilizado por Emotrix. Almacenar la data proveniente de los dispositivos para que posteriormente el módulo de machine learning pueda consultarlo para realizar el análisis correspondiente.

4. Módulo desarrollo de SDK enfocado en el procesamiento de señales EEG para la clasificación y reconocimiento de emociones. Desarrollar las funciones del API enfocadas en el procesamiento de señales para el reconocimiento de dos emociones básicas: alegría y tristeza por medio de la implementación de un algoritmo de machine Learning (Aprendizaje de Máquinas) para el reconocimiento de patrones, en señales electroencefalográficas (EEG) usando las salidas obtenidas con el dispositivo Emotiv y crear documentación relacionada a las funciones del API enfocadas a la detección de emociones.

5. Módulo diseño e implementación de la plataforma Figura para Emotrix. Creación y validación del diseño de la interfaz Figura de Emotrix utilizando la metodología de trabajo Interacción Humano Computador para la elaboración del diseño y desarrollo de la interfaz Figura.

6. Módulo administrar los recursos en el desarrollo de la plataforma de tecnología Brain Computer Interface, registrar sistemáticamente con métricas el avance del proyecto y proporcionar la información mediante un análisis de costos para fundamentar la toma de decisiones en las siguientes fases de Emotrix. Administrar los recursos en el desarrollo de la plataforma de tecnología Brain Computer Interface, proveer registros sistemáticos del avance del proyecto y proporcionar la información necesaria mediante un análisis e costos para tener un control de los procesos dentro del proyecto y fundamentar la toma de decisiones en las siguientes fases de Emotrix.

### III. JUSTIFICACIÓN

El proyecto Emotrix es una plataforma que permite el estudio e identificación de las emociones humanas. Esta versión inicial del proyecto proporciona y unifica un conjunto de herramientas y facilidades que plataformas cerebro-computador (BCI) ya ofrecen, pero Emotrix es en forma *open-source*. Los dispositivos existentes que permiten emular inteligencia emocional tienen precios elevados. El casco de Emotiv EPOC de 5 canales tiene un precio aproximado de \$ 300.00 y para tener acceso a la *data* se debe hacer una suscripción mensual o anual en la cual el precio varía según la cantidad de registros que consulte, puede llegar a prepagar hasta \$ 1,499.00 al año. El costo elevado limita el número de desarrolladores que pueden experimentar con dicha tecnología.

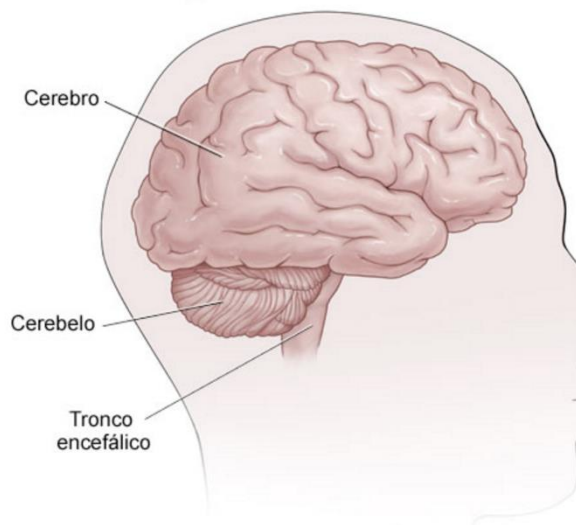
Emotrix busca en sus siguientes fases ser utilizada para fines educativos, médicos o recreacionales entre otros. En el ámbito recreacional, se puede incursionar a videojuegos para que estos se personalicen al estado emocional del jugador y se estimulen emociones para una mejor experiencia de juego. Según Newzoo sólo en el 2014 se registraron ingresos de más de \$ 75.000.00 millones en la industria de los videojuegos, muy cerca de la industria del cine \$ 90,000.00 millones y los mercados de América Latina y Asia crecen 11% al año. En el ámbito clínico, el proyecto recibió retroalimentación con el Departamento de Psicología de la Universidad del Valle sobre la aplicación en psicología clínica, específicamente por el potencial que tiene Emotrix en el diagnóstico e intervención de pacientes con Trastorno del Espectro Autista (TEA), debido a que los niños con TEA no responden a las señales emocionales en las interacciones humanas. Según una investigación en el 2008 del Centro para el Control y la Prevención de Enfermedades de Estados Unidos, 1 de cada 88 niños padece autismo en el mundo y no es posible diagnosticarse hasta los 4 o 5 años.

# IV. MÓDULO 1: PROCESAMIENTO DE SEÑALES ENCEFALOGRAFICAS PARA INTERFAZ CEREBRO-COMPUTADOR

## A. Marco teórico

1. Anatomía del encéfalo. El encéfalo está contenido en el cráneo, es la parte más voluminosa del sistema nervioso central (SNC), este se extiende en la médula espinal, contenida en la columna vertebral, y en los nervios sensitivos y motores que conducen, información sensorial al encéfalo y el control de la musculatura del esqueleto. El encéfalo se divide en las siguientes partes: tallo cerebral, cerebelo y cerebro, la forma en que esta dada la división para cada sección se puede observar en la Figura 1. (Barea, 2002)

Figura 1. Partes principales del encéfalo.



(University of Utah)

a. **Tallo cerebral.** Esta es la parte más antiguamente evolucionada del encéfalo, sirve como interconexión entre el córtex cerebral, la médula espinal y el cerebelo. Tiene como funciones principales controlar el ritmo cardíaco y respiratorio, también es el centro de diversos reflejos motores. (Barea, 2002)

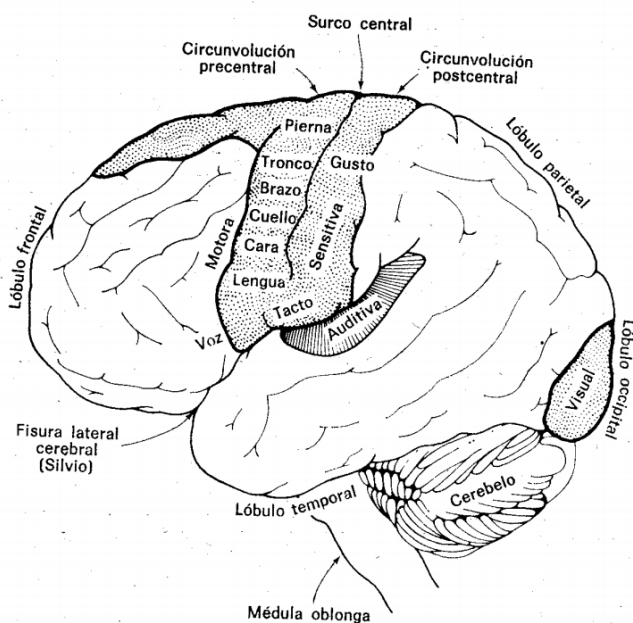
b. **Cerebelo.** Es el encargado de coordinar los movimientos voluntarios, y de controlar el equilibrio del cuerpo. Es el responsable de que los movimientos del cuerpo sean controlados y suavizados. (Barea, 2002)

c. **El cerebro.** Esta es la parte más evolucionada del encéfalo, acá se contienen todas las funciones consientes del sistema nervioso. Está dividido por dos partes llamadas hemisferio izquierdo y derecho. A toda la periferia del cerebro se le denomina corteza cerebral, es una capa fina compuesta por pliegues para

poder aumentar el área superficial. Esta es una de las subsecciones más importante del encéfalo, ya que en esta área se sitúan 9 de los 12 billones de neuronas que se tienen en el cerebro. (Barea, 2002)

A la corteza cerebral llegan todas las entradas sensoriales, como también se tiene el control de ciertas habilidades motoras, ciertas áreas parecen estar directamente relacionadas a ciertas modalidades de información, esto se puede observar en la Figura 2, donde se denota como hay áreas específicas para ciertas funciones, pero hay otras que son más generales. (Barea, 2002)

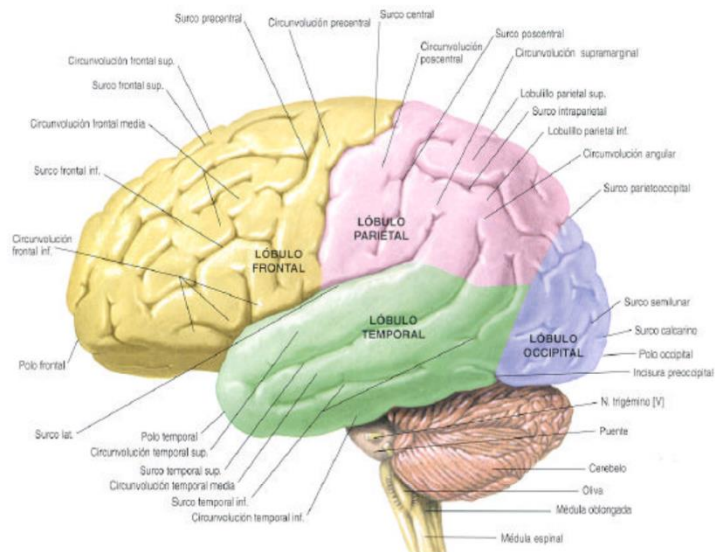
Figura 2. La corteza cerebral



(Barea, 2002)

2. Secciones corteza cerebral. La corteza cerebral está dividida en cuatro secciones principales, las cuales se pueden observar en la Figura 3, estas son: lóbulo frontal, lóbulo occipital, lóbulo parietal y lóbulo temporal. Cada una de ellas tienen ciertas funciones a cargo.

Figura 3. División de corteza cerebral.



(University of Utah, s.f.)

a. **Lóbulo frontal.** En esta parte se encuentran las neuronas relacionadas a las funciones motoras. A la sección delantera de este lóbulo, se le denomina en ciertos casos lóbulo prefrontal, ya que en esta sección se encuentran las neuronas relacionadas a ciertas funciones motoras especiales, como el control de los ojos. **(Barea, 2002)**

b. **Lóbulo occipital.** Este lóbulo se encuentra en la parte trasera del cerebro. En esta parte se encuentra la corteza visual, lo cual significa que esta es el área en donde se interpreta la información recopilada por la retina, es decir las imágenes. **(Barea, 2002)**

c. **Lóbulo parietal.** Esta parte es la encargada de procesar todas las percepciones sensoriales del cuerpo: presión, tacto, temperatura y dolor. También está encargada de procesar toda la información espacial, realizar todos los cálculos matemáticos y de la comprensión del lenguaje. **(Barea, 2002)**

d. **Lóbulo temporal.** En esta parte se encuentra la entrada sensitiva auditiva, esta sección se encuentra por encima de los oídos. En esta sección se encuentran las neuronas encargadas de interpretar las distintas frecuencias sonoras. **(Barea, 2002).**

3. **Electroencefalografía.** Una de las funciones del tejido nervioso es producir potenciales eléctricos, ya que estos son la base para la excitabilidad del organismo. Aunque todo el sistema nervioso contiene esta característica, la electroencefalografía se basa en estudiar solo los relacionados con la corteza cerebral y las regiones directamente relacionadas con ella.

La electroencefalografía se encarga de registrar y evaluar los potenciales eléctricos generados en la corteza cerebral, generados por la actividad cerebral. Esto lo realiza por medio de electrodos.

Lo que se detecta en un área de superficie subcortical es actividad eléctrica espontánea. Esto es un lapso de tiempo de ondas lentas, superpuestas con ritmos rápidos de pequeña amplitud. En los momentos donde no se registra actividad eléctrica, se detecta un potencial eléctrico igual a cero o despreciable. (Barea, 2002)

Los potenciales anteriormente descritos, se generan cuando un conjunto grande de neuronas se sincroniza, porque se activan por alguna reacción del cerebro, para así todas generar un potencial eléctrico, para que el conjunto de todos los potenciales generados, logre conseguir que se detecte una diferencia de potenciales sobre la corteza cerebral. Por esta razón es que se detecta ondas de frecuencia lenta, que son el resultado de la combinación de los potenciales de un gran grupo de neuronas sincronizadas. Y pequeñas señales con ritmos más rápidos, que son el resultado de pequeños subgrupos de neuronas interactuando. Esto lleva a al cuestionamiento de qué tipo de ondas son las que se detectan y que información aportan, esto se detalla en la siguiente sección. (Barea, 2002)

La actividad electroencefalográfica puede captarse de distintas formas: Por encima del cuero cabelludo, en la base del cráneo, en cerebro expuesto y en localizaciones cerebrales profundas. Dependiendo de qué forma se utilice, así deben ser los electrodos que se utilicen. Dependiendo también de qué forma se capten las señales, así también será la integridad de la señal recopilada, ya que entre más adentro del cerebro se encuentre el electrodo, mejor será la señal recopilada, siendo así que cuando se obtenga la señal estando por encima del cuero cabelludo, será la forma en que se tendrá menor integridad e intensidad de la señal electroencefalográfica, por la distancia que se tiene al punto en donde se origina la señal. (Barea, 2002)

a. **Electrodos.** Un electrodo es un dispositivo que lleva las señales eléctricas de los músculos, el cerebro, el corazón, la piel u otras partes del cuerpo hasta los aparatos de registro para ayudar a diagnosticar ciertas afecciones. Tienen forma de una placa o aguja delgada, y deben ser de un material que sea conductor de electricidad. (Instituto nacional del cáncer, s.f.)

Para la electroencefalografía existen diferentes tipos de electrodos, ya que varían, como se mencionó anteriormente, por la cercanía que tengan al cerebro. Los tipos de electrodos que hay son:

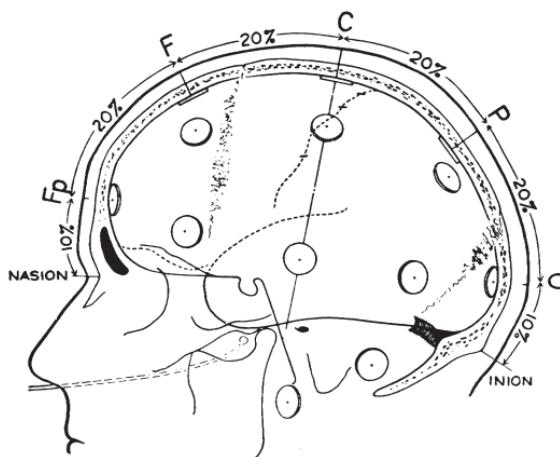
- 1) **Electrodos superficiales:** Se aplican sobre el cuero cabelludo.
- 2) **Electrodos basales:** Se aplican en la base del cráneo, sin necesidad de un procedimiento quirúrgico.
- 3) **Electrodos quirúrgicos:** Para su aplicación es necesario realizar una cirugía. (Barea, 2002)

Los electrodos de superficie pueden ser de dos tipos, de adherencia o de contacto. Los electrodos de contacto son normalmente pequeños discos metálicos con un radio aproximado de 5mm, estos electrodos son normalmente de plata-cloruro de plata, ya que este material tiene una menor tensión galvánica, esto se refiere a que aporta una menor componente DC a la señal recopilada. Este tipo de electrodos para disminuir su impedancia entre la superficie craneal y ellos, deben utilizar una solución salina, ya que esto ayuda a que haya una mejor conducción de electricidad en el medio. (EEGinfo, s.f.)

b. **Sistema de posicionamiento de electrodos 10-20.** Es un sistema que surgió por la necesidad de estandarizar resultados, ya que si se imponía un estándar para saber dónde se debía colocar los electrodos sobre el cráneo, se podrían generalizar resultados, y hacer comparaciones directas, ya que el punto de referencia sería el mismo para los experimentos que se realizaran. (Klem & Elger, 1999)

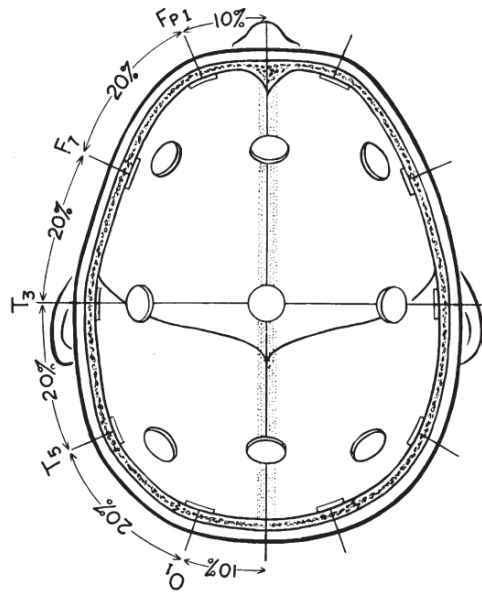
Este estándar se basa en el porcentaje en el que se distancia cada electrodo del total del espacio disponible. Como lo indica el nombre, se colocan cada electrodo con una separación 10% o 20% de la distancia total que se tiene disponible para colocarlos. Tres diferentes ejemplos de cómo es que se posicionan los electrodos según el sistema 10-20, se pueden observar en las figuras 4, 5, y 6. (Barea, 2002)

Figura 4. Ejemplo de posicionamiento de electrodos según el sistema 10-20, colocados de forma vertical de inion a nasion



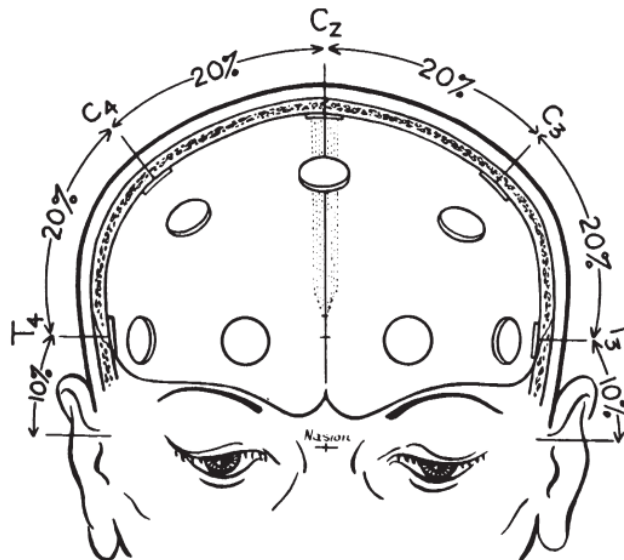
(Klem & Elger, 1999)

Figura 5. Ejemplo de posicionamiento de electrodos según el sistema 10-20, colocados de forma horizontal de inion a nasion.



(Klem & Elger, 1999)

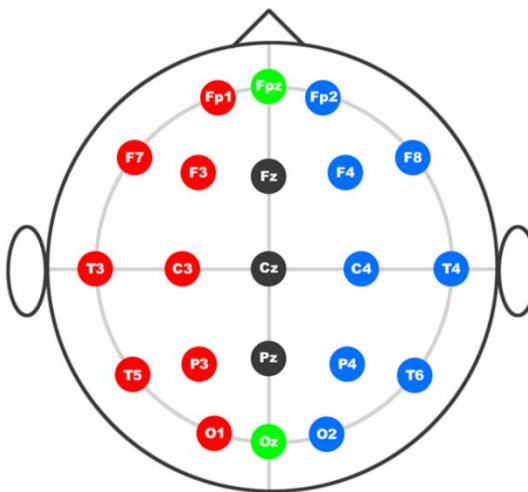
Figura 6. Ejemplo de posicionamiento de electrodos según el sistema 10-20, colocados de forma vertical de oreja a oreja.



(Klem & Elger, 1999)

Los electrodos reciben una letra inicial dependiendo en la posición en que se encuentren, ya que dicha letra está directamente relacionada con la posición de la corteza cerebral sobre la cual se encuentren. Luego tienen un numero relacionado también. Dependiendo el lado en que se encuentren también se les asigna un color, si es que se utiliza código de colores. Las letras que se pueden colocar a cada electrodo son: F(Frontal), PF(Prefrontal), O(Occipital), P(Parietal), C(Central) y T(Temporal). La Figura 5 muestra todos los electrodos que constituyen el sistema de posicionamiento 10-20. (Oostenveld, 2002)

Figura 7. Sistema de posicionamiento 10-20 completo.



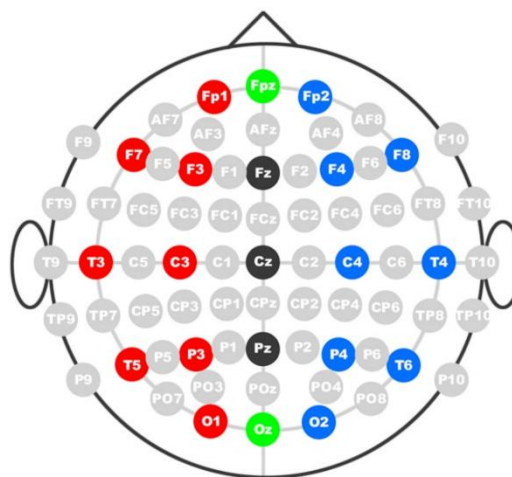
(Trans Cranial Technologies , 2012)

c. **Sistema de posicionamiento de electrodos 10-10.** Este sistema es una versión extendida del estándar 10-20, ya que en este sistema se colocan los electrodos cada 10% de la distancia todos, generando esto que haya electrodos en medio de todos de los establecidos en el estándar 10-10. Para colocarlos se sigue el mismo proceso utilizado en el estándar 10-20. (Jurcak & Tsuzuki, 2005)

Este sistema surgió por la necesidad de tener más exactitud en las mediciones de las señales. Esto también es debido a las posibilidades que brinda la evolución de la tecnología, ya que, con la posibilidad de poder procesar más canales, era imperativo poder generar otro estándar que satisficiera dicha necesidad. Este sistema extendió de 21 electrodos del estándar 10-20 a un total de 81 electrodos. El total de los electrodos y sus posiciones se puede observar en la Figura 8. (Jurcak & Tsuzuki, 2005)

Este sistema no ha sido reconocido como una estándar internación, debido a que no ha sido probada su efectividad, aunque ya bastantes instituciones alrededor del mundo ha lo han aceptado como su sistema determinado. (Jurcak & Tsuzuki, 2005)

Figura 8. Sistema de posicionamiento 10-10 completo.



(Trans Cranial Technologies , 2012)

4. **Clasificación señales EEG.** Las señales electroencefalografías pueden clasificarse en cuatro tipos principales, cada una de ellas se caracteriza principalmente por tener un rango de frecuencia asociado y una sección del cerebro donde su presencia es mayor. La detección de cada una de ellas puede ser utilizada para indicar el tipo de proceso que está realizando el cerebro, por ejemplo: Si está en un estado de concentración elevado, o en un estado de relajamiento.

a. **Ondas alpha.** Estas señales se encuentran en un rango de frecuencias de 7-12Hz, su aparición habitual es en el área occipital, en esta área estas señales pueden alcanzar una amplitud de hasta 120 microVoltios, pero pueden ser encontradas desde los 50 microVoltios, aunque estas señales se caracterizan por tener una amplitud de onda bastante variada. Las ondas alpha aparecen normalmente en las personas despiertas, recostadas y con los ojos cerrados. Si la persona abre los ojos estas ondas se atenuarán o bloquearán. (Morillo, s.f.)

b. **Ondas beta.** Estas señales se encuentran en un rango de 14 a 40 Hz, estas señales se presentan cuando el individuo está en un estado de alerta, o está realizando algún razonamiento lógico. (Kotsos, 2016). Tienen una amplitud pequeña, su amplitud tiende a no ser mayor que 15 microVoltios. Estas señales se presentan principalmente en el área frontal y frontal superior. Cuando estas señales aparecen a ritmos altos, normalmente indican algún tipo de daño cerebral, por lo general deben ir a menos de 25 Hz. (Morillo, s.f.)

c. **Ondas delta.** Esta es la señal con la frecuencia más baja, se encuentra en el rango de 0.3 a 3 Hz. Esta señal se da principalmente en estado de sueño profundo, en el momento donde el cuerpo se está

regenerando. En la mayoría de los casos cuando se presentan estas señales, no se está teniendo sueños, debido al nivel de inconciencia que se tiene. (BrainWave Laboratories, s.f.)

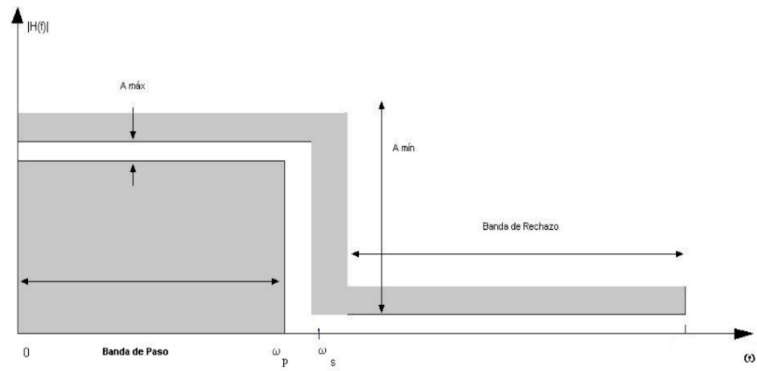
d. **Ondas theta.** Estas señales están en el rango de 3-7 Hz, aparecen habitualmente en menores a 15 años y en adultos somnolientos. Estas ondas aparecen normalmente en la mayoría de las áreas del cerebro. Su mayor amplitud se da cuando se está pasando de vigilia a sueño, su amplitud es menor a 100 microVoltios. (Morillo, s.f.)

5. **Filtrado de señales eléctricas.** Normalmente en las aplicaciones de día a día, es de interés poder obtener o entender una señal eléctrica específica, lo cual es imposible en la realidad, ya que al no estar en un ambiente controlado a la señal siempre se le agregan señales no deseadas, que se les llama ruido. También hay casos de donde ya se saben que hay un conjunto de señales y solo se desea obtener una de ellas, para esto es necesario un dispositivo que pueda ayudar a aislar una señal a una frecuencia específica, a esto se le llama un filtro. (Mendoza, 2004) Un filtro electrónico es un dispositivo que contiene una entrada y una salida, el cual se maneja en el dominio de frecuencia, y puede rechazar o permitir el paso a señales con frecuencias específicas. Existen filtros digitales y analógicos. Todos los filtros están compuestos por tres componentes principales, la banda de paso, banda de transición y banda de rechazo. (Mendoza, 2004)

En la banda de paso se encuentran todas las frecuencias que el filtro permitirá el paso, en la banda de rechazo se encontraran las frecuencias a las que el filtro atenuara o eliminara al pasar a través de él. La banda de transición es la sección en la que se traslada de la banda de paso a la de eliminación, o viceversa, esta banda es considerada debido a que los filtros no son ideales, ya que idealmente los filtros solo tendrían banda de paso y rechazo, pero en la realidad no es así, en medio de ellas. (Mendoza, 2004)

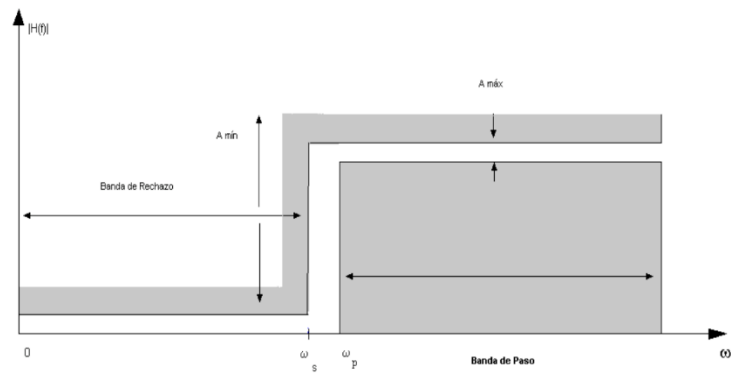
Hay diferentes tipos de filtros y son llamados según su función. Están los filtros pasa baja, que permiten el paso solo de frecuencias bajas y eliminan las altas; los filtros pasa alta, que permiten el paso de las frecuencias altas, y eliminan las frecuencias bajas; los filtros pasa banda, que permiten el paso de un rango de frecuencias establecidas, y todas las frecuencias fuera de este rango serán eliminadas; por último se encuentra el filtro rechaza banda, que es el inverso a el filtro pasa banda, ya que en este se eliminan las frecuencias que estén en un rango determinado, y todas las demás pasan. La forma de cada uno de estos tipos de filtros se puede observar en las figuras 9 a 10, en donde se muestra todas las figuras de mérito de cada uno. (Mendoza, 2004)

Figura 9. Respuesta en magnitud ideal para un filtro pasa baja.



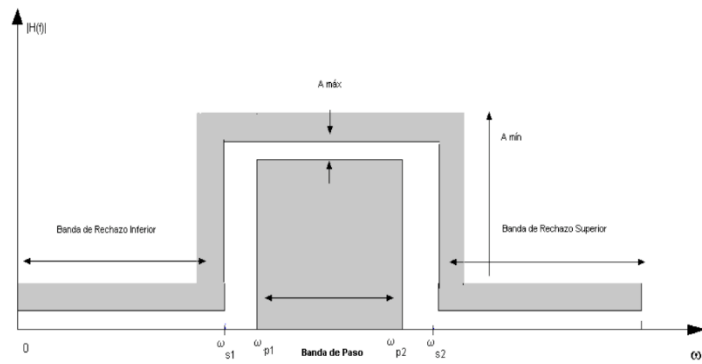
(Mendoza, 2004)

Figura 10. Respuesta en magnitud ideal para un filtro pasa alta.



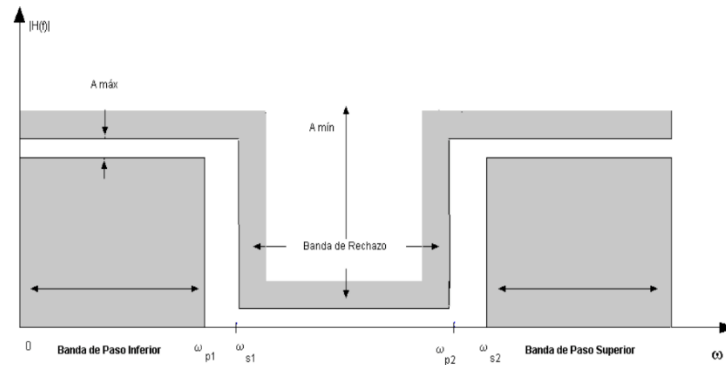
(Mendoza, 2004)

Figura 11. Respuesta en magnitud ideal para un filtro pasa banda.



(Mendoza, 2004)

Figura 12. Respuesta en magnitud ideal para un filtro rechaza banda.



(Mendoza, 2004)

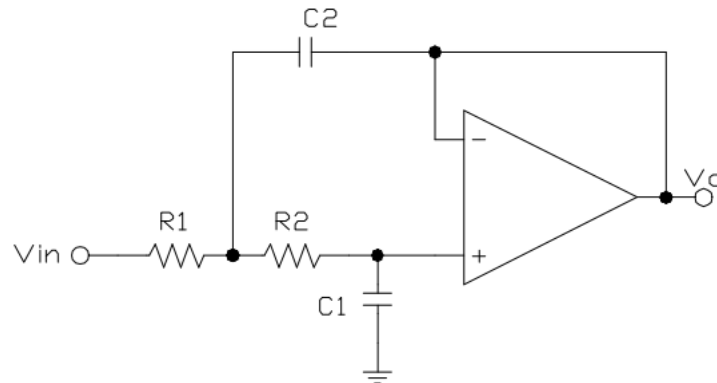
6. **Filtros analógicos.** Como se mencionó en la sección anterior, una de las dos divisiones de filtros electrónicos que hay, son los filtros analógicos, estos son realizados con dispositivos electrónicos, y según cuales sean estos dispositivos, los filtros pueden ser activos o pasivo. (Mendoza, 2004) Un filtro pasivo utiliza solo resistencias, capacitores e inductores, y no necesita otra fuente de alimentación externa, este funciona solo con el paso de la señal por los componentes. Un filtro activo se construye con amplificadores operacionales, y los componentes mencionados para filtros pasivos. Este tipo de filtro si requiere una fuente de alimentación externa. (Suárez, 2005)

Existen diferentes prototipos de filtros analógicos, cada uno con características diferentes, los cuales pueden ser utilizados según la aplicación que se desee implementar, por ejemplo, si es indispensable que no haya oscilaciones en la banda de paso, o si es necesario una banda de transición pequeña, según dichos requerimientos se debe escoger el prototipo.

a. **Filtros Butterworth.** Este filtro tiene como ventaja principal que en su banda de paso tiene una ganancia constante, lo cual no agrega distorsión a la señal filtrada, su desventaja es que su banda de transición es ancha, esto significa que o puede que atenúe frecuencias que no se desea atenuar, o que no atenúe frecuencias que deberían de estar por completo en la banda de rechazo. Para hacer más angosta la banda de transición, se debe utilizar un filtro de orden mayor, lo cual infiere en circuitos más grandes y costosos. (Suárez, 2005) En la Figura 14 se puede observar la ganancia en frecuencia del filtro Butterworth pasa baja de orden 10. (Mendoza, 2004)

b. **Filtros Sallen-Key.** Este es filtro Butterworth de orden 2, el cual tiene una topología bastante simple. Su forma mas simple, es cuando tiene ganancia unitaria. La estructura de este filtro en su configuración de pasa baja con ganancia unitaria, se puede observar en la Figura 13. (Universidad del país Vasco, 2016)

Figura 13. Topología filtro Sallen-Key pasa baja.



(Universidad del país Vasco, 2016)

La función de transferencia de este filtro igualando el valor de R1 y R2, está dada en la ecuación 1. Para obtener la frecuencia de corte de este filtro y su factor de calidad, se comparan los coeficientes de esta función de transferencia con los de la función de transferencia de un filtro pasa baja de orden 2, esta se puede observar en la ecuación 2.

$$H(s) = \frac{1}{(R^2 * C1 * C2)s^2 + (2 * R * C2)s + 1} \quad (\text{Universidad del país Vasco, 2016}) \quad (1)$$

$$H(s) = \frac{1}{(s^2 / \omega_c^2) + s / (\omega_c * Q) + 1} \quad (\text{Silva, 2008}) \quad (2)$$

Por lo tanto, igualando los coeficientes del polinomio del denominador de la ecuación 1 y la ecuación 2, se obtiene la frecuencia de corte  $\omega_c$  y el factor Q, en términos de los componentes del filtro.  $\omega_c$  y Q, se puede observar en las ecuaciones 3 y 4, respectivamente.

$$\omega_c = \frac{1}{\sqrt{R^2 C1 C2}} \quad (3)$$

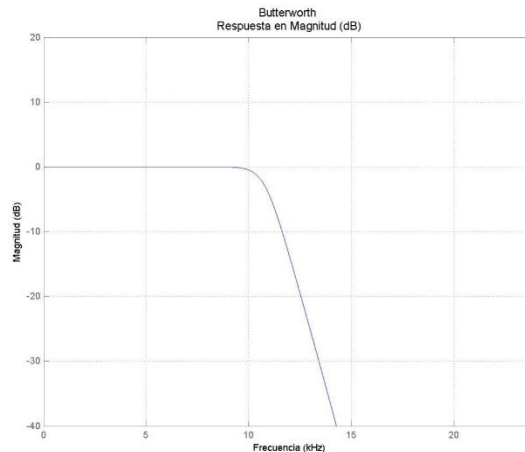
$$Q = \frac{\sqrt{R^2 C1 C2}}{2 * R * C2} \quad (4)$$

La variación con la configuración pasa alta de este filtro, es la posición de los capacitores y las resistencias, ya que en donde van las resistencias, ahora estarán los capacitores, y viceversa. La frecuencia de corte del filtro pasa alta, está dada también por la ecuación 3. (Jurican, s.f.)

c. **Filtro Chebyshev.** La principal ventaja de este filtro es que su banda de transición es angosta, aunque no es una pendiente infinita, tiene una pendiente bastante grande, lo cual hace que el rango de frecuencias que se encuentran en esta sección sea menor, es ideal si lo que interesa es tener una diferencia marcada entre las frecuencias que pasan y las que no. Uno de los problemas es tiene oscilaciones en su banda

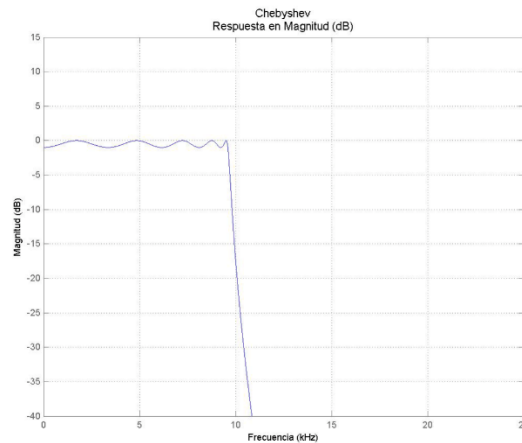
de paso, lo cual agrega distorsión a las señales que se dejan pasar, esto podría ser crucial si es de interés ver las amplitudes de las frecuencias que pasan. (Suárez, 2005) En la Figura 15 se puede observar la ganancia en frecuencia de un filtro Chebyshev pasa baja de orden 10. (Mendoza, 2004)

Figura 14. Ganancia en frecuencia de filtro Butterworth pasa baja de orden 10.



(Mendoza, 2004)

Figura 15. Ganancia en frecuencia de filtro Chebyshev pasa baja de orden 10.

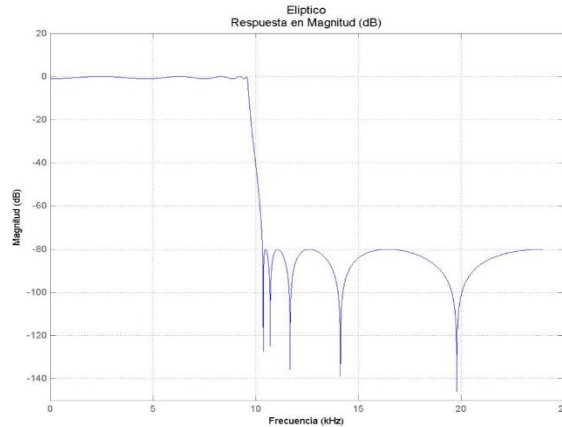


(Mendoza, 2004)

d. **Filtro elíptico.** Este filtro si tiene una banda de transición bastante angosta, su transición es bastante abrupta, estos filtros son útiles al igual que el tipo Chebyshev para definir mejor las frecuencias que pasan y las que no. Este filtro tiene oscilaciones en la banda de paso y la de rechazo. Las oscilaciones en la banda de rechazo no son un problema, si se dan a atenuaciones altas, ya que no importa si oscila, aunque así siempre atenúa las señales en la banda de rechazo. (Suárez, 2005) Mas, si las oscilaciones ya afectan que algunas frecuencias en la banda de rechazo no se atenúen lo suficiente, esto si podría implicar un problema,

para asegurar esto, el orden del filtro debe ser mayor, aunque esto también afectaría la banda de paso. En la Figura 16 se puede observar la ganancia en frecuencia de un filtro elíptico de orden 10. (Mendoza, 2004)

Figura 16. Ganancia en frecuencia de filtro elíptico pasa baja de orden 10.



(Suárez, 2005)

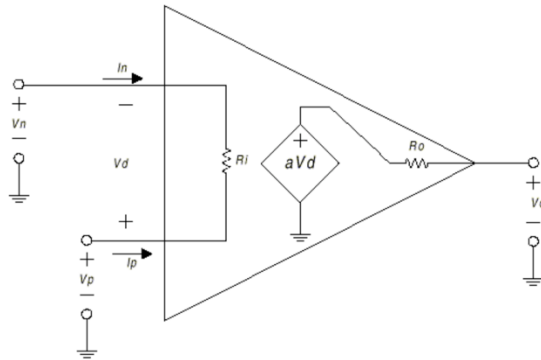
e. **Amplificador operacional.** El amplificador operacional es un dispositivo electrónico que idealmente tiene las siguientes características:

- 1) Ganancia infinita
- 2) Resistencia de entrada infinita
- 3) Resistencia de salida cero

Aunque en la realidad, realmente no alcanza estas condiciones ideales, son valores que según la aplicación podrían ser aproximados a dichas cantidades idealidades.

El esquema de un amplificador operacional, se puede observar en la Figura 17. En esta imagen se puede observar, a que se refiere cada una de las características ya descritas. La ganancia infinita, se refiere a que la salida es igual a la diferencia de voltaje entre sus dos terminales de entrada amplificada por una ganancia  $A$ , la cual tiende a infinito. Y de que entre sus dos terminales de entrada no hay paso de corriente. (Universidad del país Vasco, 2005)

Figura 17. Esquema del amplificador operacional.



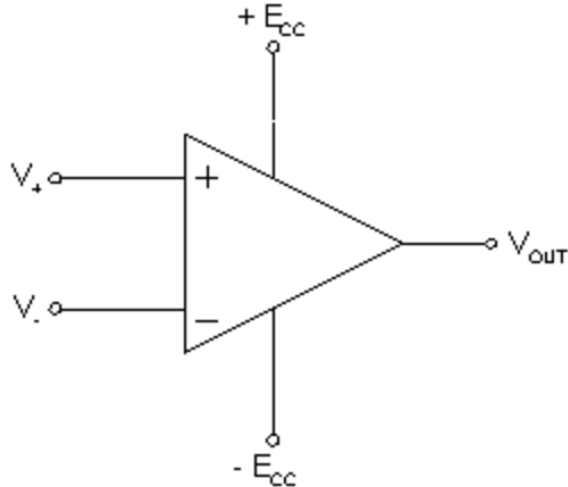
(Universidad del país Vasco, 2016)

Aunque el rendimiento de la amplificación de este dispositivo tiene sus limitaciones, ya que este en ningún momento puede alcanzar un voltaje infinito en su salida, ya que para este tiene una alimentación negativa y positiva, las cuales son el límite para el voltaje que puede brindar en su salida, ya sea negativo o positivo. Si el voltaje que debería tener en la salida es mayor a sus voltajes de alimentación, el amplificador operacional se saturara, y simplemente brindara su voltaje de alimentación. Esto se debe evitar que suceda, ya que, si se tiene una señal analógica que sobre pase el valor de saturación, no se podrá observar los cambios de voltaje que sean mayores al voltaje de referencia, ya que en la salida solo se mantendrá este voltaje.

En la Figura 18, se puede observar el esquemático se utiliza para representar al amplificador operacional en circuitos eléctricos. En donde a la entrada  $V+$  se le conoce como “Entrada no invertida”, a la entrada  $V-$  como “Entrada diferencial invertida”, a  $+Ecc$  como “Voltaje de referencia positivo”, a  $-Ecc$  como “Voltaje de referencia negativo”. (Universidad del país Vasco, 2005)

Las características de este dispositivo han sido utilizadas para generar diferentes circuitos con diferentes aplicaciones, entre sus principales aplicaciones se encuentra operaciones aritméticas entre señales, filtrado de señales, derivación e integración de señales. (Universidad del país Vasco, 2005)

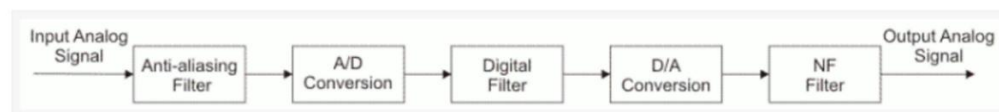
Figura 18. Símbolo esquemático del amplificador operacional.



(Ruiz, 2001)

7. **Filtro digitales.** Esta clasificación de filtros a diferencia de los analógicos, no se realiza con diferentes componentes electrónicos, para poder obtener el filtro deseado, aquí se utilizan operaciones aritméticas con precisión limitada para poder manejar la distribución de las frecuencias de una señal. Pero estas operaciones aritméticas no se le aplican a la señal analógica, si no a muestras de esta señal, las cuales son representadas con valores digitales, numero de cierto tamaño de bits. La estructura general para poder implementar un filtro digital se muestra en la Figura 19, en este esquema se muestra los componentes necesarios para poder regresar una señal analógica después de haberla filtrado. (Mendoza, 2004)

Figura 19. Componentes para implementación de filtro digital.



(Milvojevic, 2009)

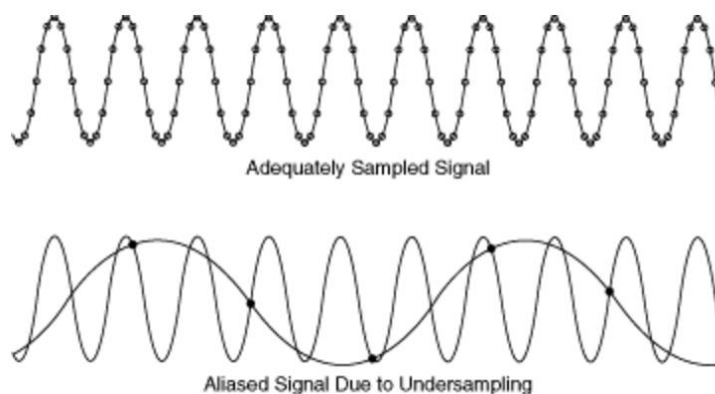
a. **Teorema del muestreo de Nyquist.** El teorema del muestreo de Nyquist es un principio fundamental para el filtrado digital de señales, ya que este define la frecuencia a la que se debe muestrear las señales a filtrar, esta definición se puede observar en la ecuación 1. Donde  $F_s$  es la frecuencia de muestreo, y  $F_{max}$  es la máxima frecuencia que alcanza la señal de entrada. En la ecuación 1 se establece que la frecuencia de muestreo debe ser al menos 2 veces mayor a la frecuencia máxima de la señal de entrada, esto se hace para evitar distorsión, y no se de el fenómeno conocido como Aliasing, que es que una señal sea interpretada como otra, por la acción de muestreo, esto se puede observar en la Figura 20, en donde en la parte baja de la imagen se muestra lo que sucede si se muestrea una señal por arriba de la frecuencia de

muestreo, en donde se puede observar que la señal que es reconocida por las muestras es una con menor frecuencia, la cual sería un Alias de la señal original, esta frecuencia estaría en términos de la frecuencia de muestreo. (Milvojevic, 2009) (González V. , 2015).

$$2F_s \ll F_{max} \text{ (González V. , 2015)} \quad (5)$$

Con base en esto, se define la importancia del primer bloque en la Figura 19, ya que este bloque es un filtro pasa baja analógico, que ayuda a banda limitar la señal de entrada, esto significa asegurar que la frecuencia máxima este definida, ya que sin este filtro se tendrán frecuencias mayores a las esperadas, ya que como se mencionó, toda señal siempre va acompañada de ruido en distintas frecuencias, y este ruido si no se tiene este filtro, puede ser interpretado como una de las señales de interés, y de este modo generar error en las aplicaciones en la que se utilice las señales de salida del filtro. (Milvojevic, 2009)

Figura 20. Ejemplo de una señal muestreada de forma correcta (Parte superior), y una con el efecto de Aliasing (Parte inferior).



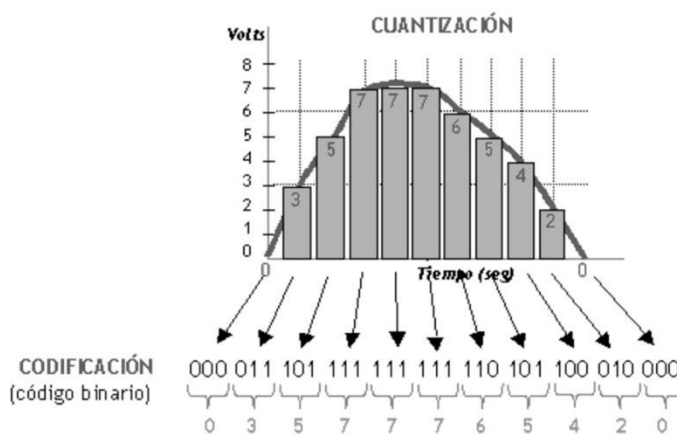
(National Instrument, s.f.)

**b. Digitalización.** El segundo bloque en la Figura 16, corresponde al convertido de análogo a digital. La digitalización de una señal analógica consiste en tomar muestras (tomar valores de la señal análoga en momentos específicos de tiempo) de una señal continua, donde las muestras se toman a una frecuencia  $F_s$ , que debe cumplir con el teorema de Nyquist, y cada una de estas muestras es representada con cierto número de bits, el cual está definido por el convertidor análogo a digital utilizado. Esto se hace con la intención de poder entender las señales analógicas en un dispositivo digital, aunque con esta acción se pierde cierta información de la señal original, es un costo que vale la pena pagar, para poder almacenar y procesar todas las señales del mundo real. Este error que se introduce por digitalizar la señal, proviene de dos fuentes, una es que no se tome suficientes muestras de la señal, para poder interpretarla ya muestreada. (Universidad de Navarra, s.f.) El otro error es llamado “Error de cuantización”, el cual sucede al no representar de forma exacta el valor real de la señal analógica, ya que el valor con el que se represente depende de la precisión que se pueda obtener por la cantidad de bits que se tenga para cuantizar la señal dentro del dispositivo digital. Dependiendo el rango en el que se deba cuantificar, y la cantidad de bits que se tenga, este error será menor,

ya que la exactitud de la digitalización es directamente proporcional al número de bits de la conversión, e inversamente proporcional al rango de valores de la conversión. Por lo tanto, después de esto se puede decir que la señal digitalizada es discreta en sus valores e instantes en el tiempo. (Eveliux, s.f.)

Las señales al ser digitalizadas, no mantienen el valor que tenían en su forma análoga, ya que reciben una representación binaria del nivel al que se encuentran, se puede decir que son codificados, ya que no pueden recibir un valor decimal, como lo hacían en su forma análoga, por ejemplo, al tomar una foto, lo que se almacena no es el color percibido por cada pixel, por ejemplo “turquesa”, “rosado”, etc. Si no la intensidad de rojo, la intensidad de verde y la intensidad de azul, utilizando  $n$  bits, para representar la intensidad de cada uno de los tres colores, pudiendo tomar  $2^n$  valores diferentes de intensidad cada color. Este proceso sucede así para cualquier señal análoga, que se desee digitalizar. Un ejemplo del resultado de una digitalización comparado con la señal real, se puede observar en la Figura 21. (Eveliux, s.f.)

Figura 21. Ejemplo de señal digitalizada.



(Eveliux, s.f.)

c. **Filtros de respuesta finita impulsional.** Existen dos tipos de filtros digitales que pueden ser utilizados, uno de ellos son los filtros que tienen una respuesta finita impulsional (FIR, por sus siglas en inglés “Finite Impulsive Response”), esto se refiere a que, si la entrada al filtro es un pulso de duración finita, la salida será un pulso con duración finita también. Esto se da porque este filtro para obtener la salida, solo utiliza los valores de la entrada. La ventaja que tienen estos filtros es que pueden ser diseñados para tener una fase lineal, lo cual los hace ideales para las aplicaciones donde se requiera tener una fase lineal. Estos filtros también por su estructura tienden a ser estables. (Milvojevic, 2009)

La estructura general de un filtro digital, se observa en la Figura 22, en donde  $y$  es el vector de salida del filtro,  $x$  es el vector de muestras de entrada.  $a_n$  y  $b_m$  son los coeficientes del filtro, para un filtro FIR, los coeficientes  $a_n$ , son cero excepto para  $n$  igual cero. La forma de un filtro FIR se muestra en la Figura 23, donde  $h$  es un vector que contiene los coeficientes  $b_m$  mostrados en la Figura 22,  $N$  es el orden del filtro. El diagrama de bloques de un filtro FIR se muestra en la Figura 24. (Mendoza, 2004) (Milvojevic, 2009)

Figura 22. Estructura general de filtro digital.

$$\sum_{n=0}^N a_n y(k-n) = \sum_{m=0}^M b_m x(k-m)$$

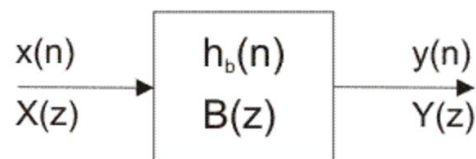
(Milvojevic, 2009)

Figura 23. Estructura general filtro digital FIR.

$$y[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n-k]$$

(Milvojevic, 2009)

Figura 24. Diagrama de bloques de filtro FIR.



(Milvojevic, 2009)

Existen diferentes formas de diseñar filtros FIR, pero la más utilizada es la denominada método de ventanas. En este método se utilizan los modelos ideales de los filtros, en estos modelos se definen las características que se desea que tenga el filtro final, como la ganancia y frecuencia de corte. Luego se utiliza la respuesta en el tiempo del filtro, esta respuesta se muestrea a la frecuencia de muestreo del filtro. Luego para delimitar el filtro a un orden específico, se utiliza una ventana, que hace que solo se utilicen ciertas muestras de la respuesta transitoria del filtro ideal, el número de muestras que se utilicen es igual al orden del filtro. Esta ventana puede modificar o no la ganancia de cada muestra utilizada, ya que se multiplica la

ganancia de cada muestra de la ventana, con su muestra respectiva del filtro. El resultado de la multiplicación de las muestras de la ventana con las del filtro, son los coeficientes del filtro FIR. Entre más ancha sea la ventana, el filtro tendrá un mejor comportamiento, ya que será de un orden mayor, pero entre más grande sea el orden del filtro, se necesitan más recursos computacionales para poder realizarlo. (Milvojevic, 2009)

Otro factor importante para obtener un orden con mejor respuesta en frecuencia, es la ventana utilizada, ya que existen diferentes tipos de ventana, y cada una puede ser utilizada según la aplicación, ya que, si se necesita una banda de transición angosta, es necesario utilizar una ventana que pueda lograr dicho objetivo. Entre mejores resultados se dese, se debe utilizar una ventana con coeficientes más complejos, los cuales requieren también capacidades computacionales mayores. (Milvojevic, 2009)

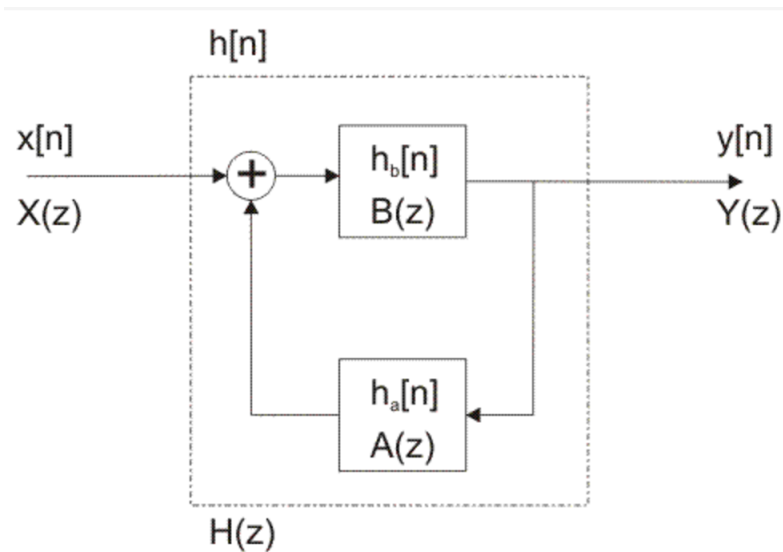
d. **Filtros de respuesta infinita impulsional.** Los filtros de respuesta infinita impulsional (IIR, por sus siglas en inglés “Infinite Impulse Response”) son el otro tipo de filtros digitales que hay, recibe este nombre debido a que la salida del filtro no es de respuesta finita, aunque la entrada sea un pulso de duración finita, esto sucede porque la salida no solo depende de los valores de la entrada, si no de los valores de la salida pasados también. Para los filtros IIR los coeficientes han de la estructura general de un filtro digital mostrada en la Figura 22, no son cero. La estructura de un filtro IIR se muestra en la Figura 25, el diagrama de bloques de los filtros IIR se muestra en la 26. Estos filtros se caracterizan por ser de menor orden que un filtro FIR, para poder cumplir con los mismos requerimientos. Aunque en estos filtros no se puede asegurar tener una fase lineal. También pueden tender a ser inestables, si el orden del filtro es grande. (Milvojevic, 2009)

Figura 25. Estructura de filtro IIR.

$$y[n] = \sum_{k=0}^N b[k] \cdot x[n - k] - \sum_{k=1}^N a[k] \cdot y[n - k]$$

(Milvojevic, 2009)

Figura 26. Diagrama de bloques de filtro IIR.



(Milvojevic, 2009)

Para diseñar un filtro IIR, se utilizan los prototipos de filtros analógicos mencionados anteriormente, ya que a partir de estos se hacen transformaciones, para pasar del tiempo continuo al discreto. Estas transformaciones se realizan por diferentes aproximaciones que existen para transformar una función de transferencia del dominio de  $s$ , al dominio  $z$ , donde se propone una equivalencia entre  $s$  y  $Z$ . Una de las aproximaciones más usadas es la bilineal, la cual se muestra en la Figura 27. Al ya tener la aproximación que se utilizara, es necesario decidir a partir de que prototipo analógico se obtendrá el filtro, ya sea el prototipo Butterword, Chebyshev o algún otro, el prototipo se escoge según las necesidades de la aplicación, según las características de cada uno descritas. Al ya tener el prototipo, se utiliza su función de transferencia, y se hace la sustitución de  $s$  por su aproximación dada de  $Z$ . Al ya tener una función de transferencia en el dominio  $Z$ , se anti transforma dicha función, para obtener la ecuación de diferencias que implementara este filtro digital.

Figura 27. Aproximación bilineal.

$$s = \frac{1 - z^{-1}}{1 + z^{-1}}$$

(Milvojevic, 2009)

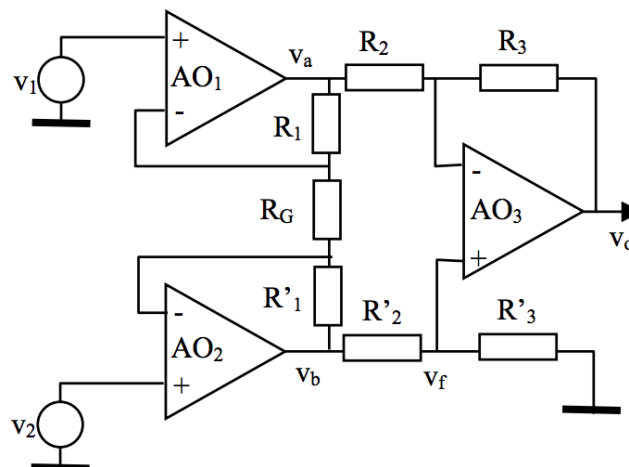
8. **Impresoras 3D.** Una impresora 3D es una máquina que por medio de calor derrite un material plástico, para generar un filamento delgado, ya que por medio de filamentos va formando capas de material, ya que al salir el filamento caliente y unirse con otros, se funden al enfriarse. Por medio de varias capas es que se construyen las figuras hacia arriba. Para poder colocar el filamento sobre toda la superficie de cada capa, se utilizan dos motores para que el extrusor que calienta el material, pueda moverse por todo el plano horizontal. Al terminar una capa, se pasa a la siguiente utilizando otro motor, que sube el extrusor en el eje z. Siguiendo este proceso iterativamente hasta completar todas las capas, es que se obtiene la Figura tridimensional. (Tecnología, s.f.)

El diseño de la Figura que se desea fabricar, se realiza como un sólido en un software de diseño de pizas, luego se utiliza otro software que convierta el sólido al formato de capas y filamentos, para que así ya pueda ser fabricado por la impresora 3D.

9. **Amplificador instrumental.** Este es un amplificador construido con tres amplificadores operacionales, el cual tiene como principales características, poder obtener grandes amplificaciones de la diferencia de dos señales, esto es sumamente importante para señales pequeñas, ya que de este modo se puede eliminar el ruido común que haya entre estas dos señales, y solo amplificar el valor de interés, esto significa que este amplificador tiene un buen radio de eliminación del ruido común. La topología de este amplificador se muestra en la Figura 28. La ganancia que se puede obtener con este amplificador está dada en la ecuación 6, esta ecuación es válida si el circuito es simétrico. (Drake, 2005)

$$G = \frac{R_3}{R_2} \left( 1 + \frac{2R_1}{R} \right) \quad (\text{Drake, 2005}) \quad (6)$$

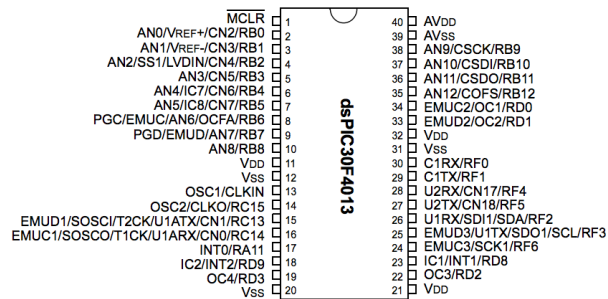
Figura 28. Topología amplificador instrumental.



(Drake, 2005)

10. **DSPIC30F4013.** El DSPIC30F4013 es un microcontrolador para realizar procesamiento de señales con una arquitectura de 16bits, es fabricado por la empresa Microchip. Está diseñado para el procesamiento de señales por las características de Hardware que contiene, ya que este microcontrolador puede alcanzar frecuencias de operación con su oscilador interno de hasta 120MHz, lo cual hace que pueda hacer operaciones a una velocidad bastante rápida. También trae dentro de su arquitectura un bloque para poder realizar operaciones aritméticas como multiplicación y división en un solo ciclo de instrucción. Puede manejar números en punto flotante en el formato Q15. Tiene Registros acumuladores de 40 bits. Contiene 13 canales para conversiones ADC, utilizando 12 bits para la conversión. En la Figura 29 se puede observar el diagrama de pines de este microcontrolador en el empaquetado PDIP. (Microchip, 2004)

Figura 29. Diagrama de pines de DSPIC30F4013.



(Microchip, 2004)

## B. Marco metodológico

1. **Amplificación de señales EEG.** Como primer paso para esta etapa, se necesitaba saber cuáles serían las amplitudes de las señales a amplificar, las cuales van en un rango de 15 a 100 microVoltios. Por lo tanto, la amplificación que se requiere es bastante grande, ya que el objetivo de la amplificación de estas señales, es que, al muestrearlas con el microcontrolador, este pueda reconocer los cambios de dicha señal, y este reconocerá señales en el rango de los 0 Voltios a 5 voltios, con precisión de 1.2 miliVoltios. Aunque se debe tener bastante cuidado de la cantidad que se amplifica, ya que en la señal que se amplificara se tendrá ruido, el cual no se puede remover hasta después de amplificar, ya que, por el hecho de filtrar, se atenúa la señal, lo cual significa que, si se hiciera el filtrado antes de amplificar, la señal de interés se eliminaría, ya que la señal es bastante pequeña por naturaleza. Para poder reducir la cantidad de ruido con la que termina la señal amplificada, se utilizó amplificadores instrumentales, ya que estos ayudan a eliminar el ruido común, y obtener la diferencia de potencial entre dos señales con la mejor relación señal-ruido. Por la forma en que se generan las señales EEG, que son impulsos que parten de un punto a otro, por las estructuras del cerebro, es que se utiliza este tipo de amplificadores que obtienen una diferencia entre dos señales. Por lo tanto, las entradas al amplificador instrumental son el electrodo que se desea leer, que es el voltaje V2, y los electrodos que se utilizan como referencia, que están en los lóbulos de las orejas, serían la otra entrada al amplificador, el voltaje V1. La ecuación 6, muestra la ganancia que se obtiene con el amplificador instrumental mostrado en la Figura 28, esta ganancia es recomendable llegarla hasta 10,000, ya que esto hace que el amplificador mantenga su radio de eliminación de ruido común, y si solo se modifica la resistencia R1, y se utilizan valores iguales de R2 y R3, esto ayuda aún más a mantener esta relación. (Drake, 2005)

Como el filtrado de la señal para obtener la información relevante se hará de forma digital, se debe colocar otros componentes antes de ingresar la señal al microcontrolador, se debe como paso crucial para el filtrado digital, se debe colocar un filtro antialias, el cual debe estar diseñado para atenuar todas las frecuencias mayores a la frecuencia de muestreo, ya que así se evade el efecto de aliasing. También se debe colocar un filtro pasa alta, para eliminar el offset DC, que la señal contiene, estos filtros a la hora de diseñarlos se deben tener en cuenta que no afecten las señales de interés. Por último, se requiere una etapa de acondicionamiento de la señal, ya que como se mencionó anteriormente, el microcontrolador solo puede detectar voltajes entre 0 y 5 Voltios, por lo tanto, se debe llevar la señal a este rango, esto por medio de un circuito que agregue un offset DC para centrar la señal en 2.5 Voltios, para que este en el centro del rango a detectar, y una etapa de amplificación, para tener la señal con un voltaje pico-pico de 5 Voltios máximo.

El diseño de las etapas mencionadas irá con base en las frecuencias y amplitudes de las señales EEG, y se utilizará el software Altium Designer, para poder realizar las simulaciones de los diseños propuestos, y ver su rendimiento. Al ya tener un circuito esquemático definido, se realizará un diseño de PCB, haciendo uso de la herramienta para crear PCBs de Altium.

Antes de realizar la PCB, se probará que los circuitos establecidos funcionen de forma correcta en un protoboard, ya que, aunque el protoboard añade más ruido que una placa PCB, es más factible realizar cambios al hacer pruebas de esta forma, que al soldar componentes a una placa. Para de este modo al momento de soldarlos, se tenga total certeza que son los valores adecuados para las señales reales, que tienen diferencia con las simuladas

El amplificador operacional que se utilizó para construir los amplificadores y filtros analógicos, fue el integrado TL082, el cual contiene en un empaquetado PDIP-8, 2 amplificadores operacionales. Se escogieron estos amplificadores operacionales debido a la cantidad de ruido que aportan a la señal, ya que el ruido que aportan este alrededor de los 16 nanoVoltios, lo cual no afecta las señales de interés, porque el ruido es un orden de magnitud más pequeño que la señal más pequeña que se manejara.

2. **Construcción de casco.** Para poder construir el casco, como primera parte se diseñó como sería el mismo, el diseño se planteó de forma bastante simple, ya que se debía pensar en la forma que se fabricaría, que sería utilizando la impresora 3D, en donde cada pieza que imprima debe tener un soporte, es decir, no puede imprimir sobre nada. El soporte debe ser ya sea la camilla de la impresora, o la misma estructura de la pieza que se está imprimiendo. Por lo tanto, esto es un impedimento para hacer un casco completo como tal, ya que este debería ser una media esfera hueca, la cual no podría ser fabricada de ninguna forma en la impresora 3D. También se debió considerar el tamaño de piezas que puede fabricar la impresora, ya que, si la pieza es demasiado grande, no cabra en el área en que puede imprimir, y no podrá ser fabricada. Por lo tanto, se diseñó un casco que cumpliera con las funciones deseadas, que colocara los electrodos en las posiciones correctas, que guardara todas las PCBs dentro de él, y que se pudiera adaptar a diferentes tamaños de cabeza.

Como primer paso se realizó el diseño de la pieza en el software Inventor de AutoCAD, en donde para poder colocar medidas a las piezas realizadas, se midió las dimensiones de la cabeza del escritor de este trabajo, en donde se asumieron formas básicas para estas medidas, se asumió que la parte trasera de la cabeza era tenía la forma de medio círculo, y con una cinta se midió su circunferencia, y con este dato se obtuvo su radio. Luego se tomó como dos partes totalmente rectas a las secciones laterales de la cabeza, y se midió la distancia entre el fin de la parte circular trasera, hasta donde termina la cabeza, esto de forma recta, no siguiendo la forma del cráneo. Ya con estos datos, se tenía dimensiones para poder diseñar las piezas.

Las piezas debían ser diseñadas para poder adaptarse a cualquier otra cabeza, ya que cada individuo tiene un tamaño diferente, aunque la forma siempre tiende a ser la misma, en términos generales. Por lo tanto, el diseño se realizó con el objetivo que pudiera ser ajustable, para que no importase si la cabeza fuese más pequeña o más grande, el cazo podría ser usado, esto se realizó dejando una parte flexible, y haciendo el cazo un poco más pequeño que las medidas que fueron tomadas en principio.

Para finalmente fabricarlo una vez realizado el diseño, se tuvo que dividir las piezas hechas de tal forma que pudieran ser unidas de nuevo al fabricarlas, ya que la pieza completa no cabría por completo en la

impresora. Para que pudieran ser unidas de nuevo, simplemente se les agrego una parte de material extra, para que pudieran ser pegadas de mejor forma. Para pegarlas una vez fabricadas, se usaría una mezcla de pegamento instantáneo y bicarbonato, para asegurar una buena adherencia de las piezas.

Antes de imprimir todas las piezas, se imprimió piezas clave, para poder observar si el dimensionamiento que se le había dado a las piezas estaba bien, es decir, si encajaba de forma correcta en la cabeza, para así poder hacer las correcciones necesarias para tener las medidas correctas, al momento de fabricar por completo el casco. Otra razón, para hacer pruebas de ciertas piezas, era para ver qué tan flexibles eran según el diseño realizado y la constitución del material, ya que para que el cazo fuera adaptable a otros tamaños de cabeza, era requerido que ciertas piezas fueran flexibles.

3. **Filtrado digital de las señales EEG.** Para poder realizar el filtrado digital se utilizará el microcontrolador para procesamiento de señales, DSPIC30F4013. Para realizar el programa, se utilizará el programa MikroC, el cual es de la empresa fabricante de esta línea de microcontroladores. Se implementará un filtro IIR pasa baja, con frecuencia de corte 59Hz, para asegurar la eliminación del ruido generado por la alimentación eléctrica. Se buscará que el filtro implementado sea del orden menor posible y que sea estable, para esto si se requiere un orden grande, se harán secciones de filtros de orden pequeño. Para diseñar los filtros se utilizó la herramienta de MikroC para el diseño de filtros digitales, la cual brinda la facilidad de calcular el tiempo del temporizador interno del microcontrolador para tomar las muestras a la frecuencia especificada; también brinda la facilidad de dar los numero en punto flotante ya en la forma que el micro controlador los maneja.

La información ya filtrada será enviada a la computadora por medio del módulo UART (Transmisión y recepción asíncrona universal), el cual lo reconoce la computadora como un elemento Serial, aunque este se conecta al computador por medio del puerto USB.

Se utilizará un canal analógico diferente para cada uno de los cuatro electrodos, la frecuencia a la que se muestreará cada electrodo será 1KHz. Como las muestras de cada canal no se pueden tomar en paralelo, si no se debe de hacer en serie, el tomar las 4 muestras debe tomar 1 milisegundo, para que así cada electrodo pueda ser muestreado a 1KHz, para que esto pueda ser posible, se debe tener separación entre la muestra de un electrodo y otro, un cuarto de milisegundo, lo cual equivaldría a que se estuviera tomando muestras a 4KHz.

## C. Resultados

### 1. Amplificación y acondicionamiento de señales EEG

a. **Amplificación y eliminación de ruido común.** El amplificador instrumental, como ya se mencionó, su ganancia está dada por la ecuación 6, en un principio se planteó que este tuviera una ganancia bastante grande, para lograr llegar las señales EEG a un valor de aproximadamente 1 Voltio. Según la teoría, las señales EEG tienen una amplitud entre 15-100 microVoltios, lo cual significa que para llegar a 1 V el

valor medio de este rango que estaría alrededor de los 50 microVoltios, se necesitaría aproximadamente una ganancia de 18000. Por lo tanto, la ganancia que se propuso para el amplificador fue de 15810, para que estuviera cercana a este al valor propuesto, y pudiera ser realizados con valores de resistencias comerciales. La ecuación 5, sustituyendo los valores de las resistencias escogidas, quedaría como se muestra en la ecuación 7.

$$G = \left(1 + 2 * \frac{100K\Omega}{220\Omega}\right) * \frac{47K\Omega}{2.7K\Omega} = 15810 \quad (7)$$

El esquemático del amplificador operacional realizado en Altium Designer, se muestra en la Figura 30. Este amplificador si amplificaba una señal en el rango mencionado, a un voltaje alrededor de 1 V. Esto se comprobó utilizando el análisis transiente de Altium, en donde se ingresó una señal de 50 microVoltios a 20Hz, diferenciándola con tierra. La señal de salida fue de aproximadamente 80 miliVoltios, esto es equivalente a una amplificación de 16000. Este resultado se puede observar en la Figura 31.

Figura 30. Esquemático de amplificador Instrumental con ganancia 15810.

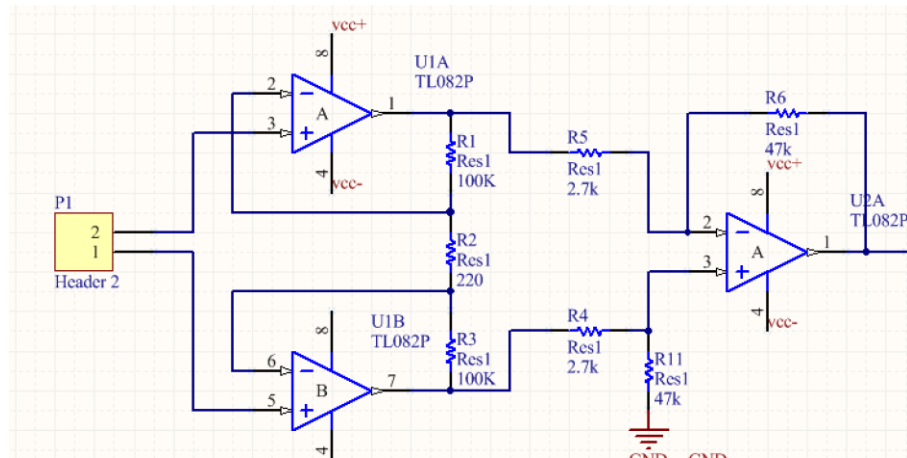
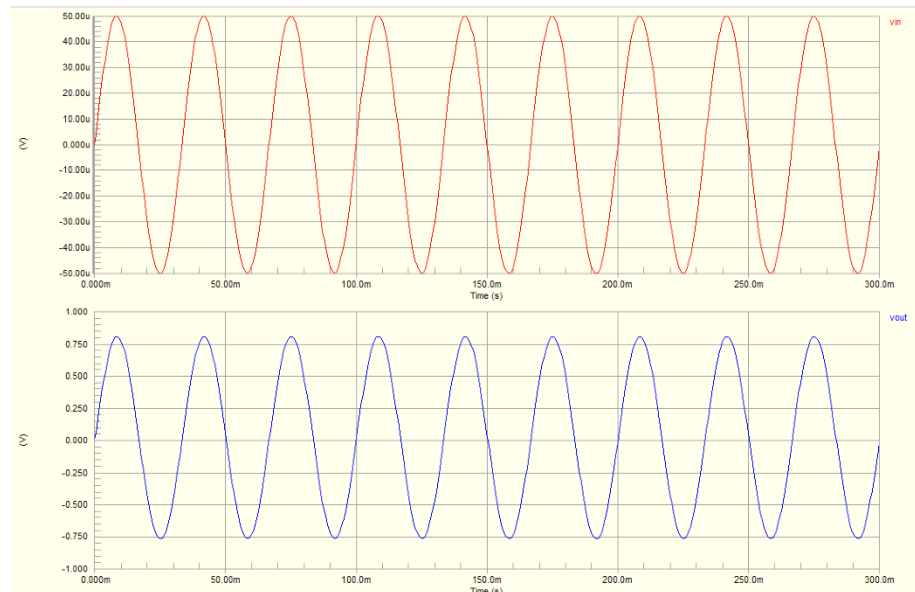


Figura 31. Señal de entrada al amplificador instrumental (parte superior), Señal amplificada por el amplificador instrumental con ganancia 15810(parte inferior).



Aunque este planteamiento se tomó de una forma demasiado ideal, ya que como se ha mencionado, estas señales están acompañadas de ruido, el cual es mucho mayor a la amplitud de las mismas, y este ruido también es amplificado, y si este ruido están en el rango de los miliVoltios, esto significa que con una amplificación de este magnitud, aun el ruido fuera de 1 miliVoltio, el voltaje al cual se amplificaría con una ganancia de 16000 sería de 16 Voltios, y si los Opamps son alimentados con +/- 9 Voltios, esto los saturara, y esto no permitirá observar ninguna señal en la salida, más que flancos positivo y negativo. Por lo tanto, al ver esto al probarlos en la realidad, se replanteo la amplificación de esta etapa.

Se disminuyó en gran cantidad la ganancia, ya que ahora solo se utilizó  $R_g$  para amplificar,  $R_2$  y  $R_3$  se dejaron con un valor igual entre ellas, la ganancia de la nueva amplificación estada dada por la ecuación 8. Se puede observar que la disminución de la ganancia es bastante considerable, ya que a menos del 5% del valor original que se había planteado. Pero con esta amplificación ya es menos probable que el opamp se sature, ya que el ruido debería ser mayor a 15 miliVoltios, para que este se saturara. El que tan grande sea el ruido dependerá bastante de cómo se coloquen los electrodos, ya que esto también afecta la componente DC de la señal, la cual también puede causar saturación en cualquiera de los voltajes del amplificador operacional.

$$G = \left( 1 + 2 * \frac{100K\Omega}{330\Omega} \right) * \frac{47K\Omega}{47K\Omega} = 606 \quad (8)$$

Aunque con esta amplificación se sigue teniendo la señal de interés con una amplitud pequeña, esta amplitud se incrementó en la última etapa, que es la de acondicionamiento de la señal, ya que acá hay otra

etapa de amplificación, esta es un simple amplificador inversor. La ventaja es que en este punto la señal ya paso por el filtro antialias (pasa baja), que elimina ya bastantes frecuencias de ruido, y por el filtro pasa alta que elimina la componente DC.

b. **Filtros Antialias.** En esta etapa el objetivo principal es eliminar las frecuencias mayores a la mitad de la frecuencia de muestreo, siguiendo el teorema de Nyquist, para evitar el Aliasing. Siendo la frecuencia de muestreo 1KHz, se deben eliminar las frecuencias mayores a 500 Hz. Para realizar esta acción, se utiliza un filtro pasa baja. Para esto se utilizó un filtro Sallen-Key en su configuración pasa baja. Aunque por las características de la señal de interés, que se encontraba en las frecuencias más bajas del rango de muestreo, este filtro no solo se diseñó para que eliminara el efecto de alias, si no que funcionara como una etapa de filtrado analógico de la señal, ya que, de este modo, desde este punto se comenzaría a eliminar las frecuencias mayores a el rango de interés, teniendo siempre como principal objetivo atenuar la señales a 60 Hz. Para diseñar el filtro se utilizaron las ecuaciones 3 y 4, ya que con estas se puede definir la frecuencia de corte del filtro y el facto Q. Seleccionando una frecuencia de corte de 33Hz, e imponiendo un valor a las resistencias de 2.2 K $\Omega$ ., Se encontró una relación de capacitores que fueran comerciales, que obtuvieran la frecuencia de corte propuesta, y aumentara el factor Q, para así tener una elevación en la banda de paso y que disminuyera la banda de transición. Los valores que se encontró para los capacitores fue C1 = 1 microfaradio y C2 = 4.7 microfaradios. El circuito esquemático de este filtro se muestra en la Figura 32. La respuesta en frecuencia del filtro se obtuvo utilizando el análisis de señales AC de Altium designer, esta se puede observar en la Figura 33.

Figura 32. Circuito esquemático de filtro Antialias.

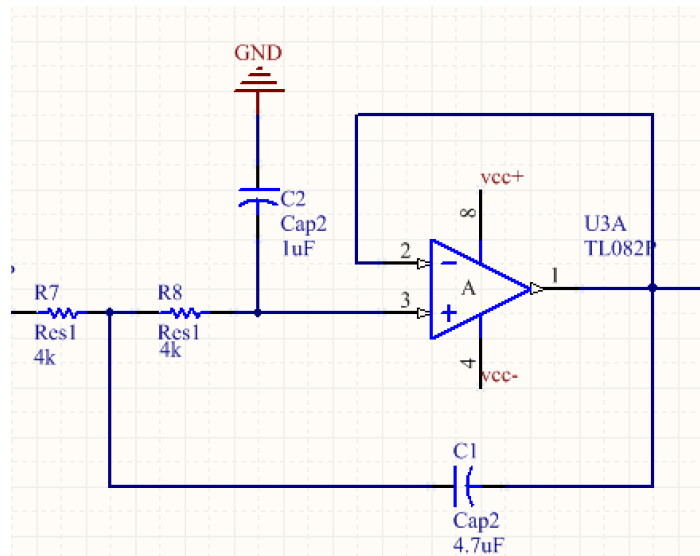
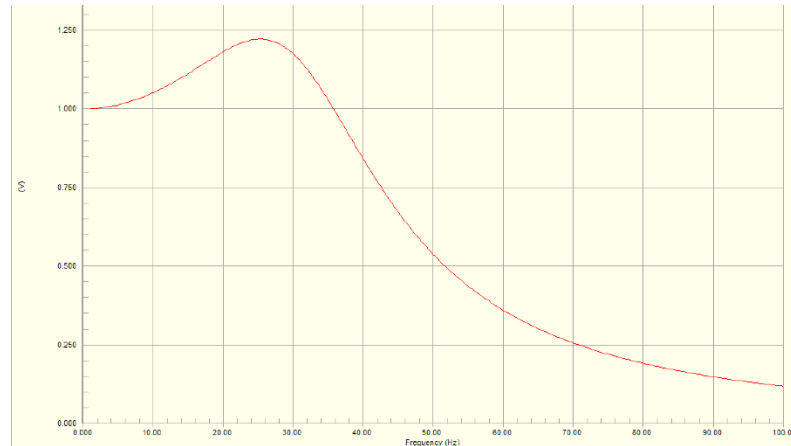


Figura 33. Respuesta en frecuencia del filtro Antialias.



c. **Eliminación de componentes DC.** En esta etapa lo que se busca es eliminar la componente DC que trae la señal de interés, esto se realiza utilizando un filtro pasa alta, ya que lo que se desea es eliminar la frecuencia igual a cero. Para esto se utilizó un filtro Sallen-Key en su configuración pasa alta. Se diseñó utilizando solamente utilizando la ecuación 3. Para este filtro se debía colocar una frecuencia de corte bastante pequeña, para que no interviniera con la señal de interés, que comienza a aparecer desde frecuencias de 0.3 Hz. Por lo tanto, se diseñó con una frecuencia de corte de 0.33 Hz. Los valores de los componentes para cumplir con esta frecuencia de corte fueron  $C1=C2=10$  microfaradios,  $R1=R2=47K\Omega$ . El circuito esquemático de este filtro se muestra en la Figura 34. Su respuesta en frecuencia se muestra en la Figura 35.

Figura 34. Circuito esquemático de filtro pasa alta.

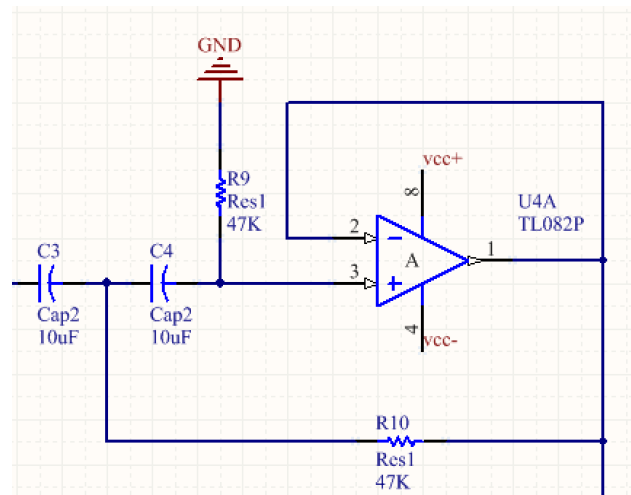
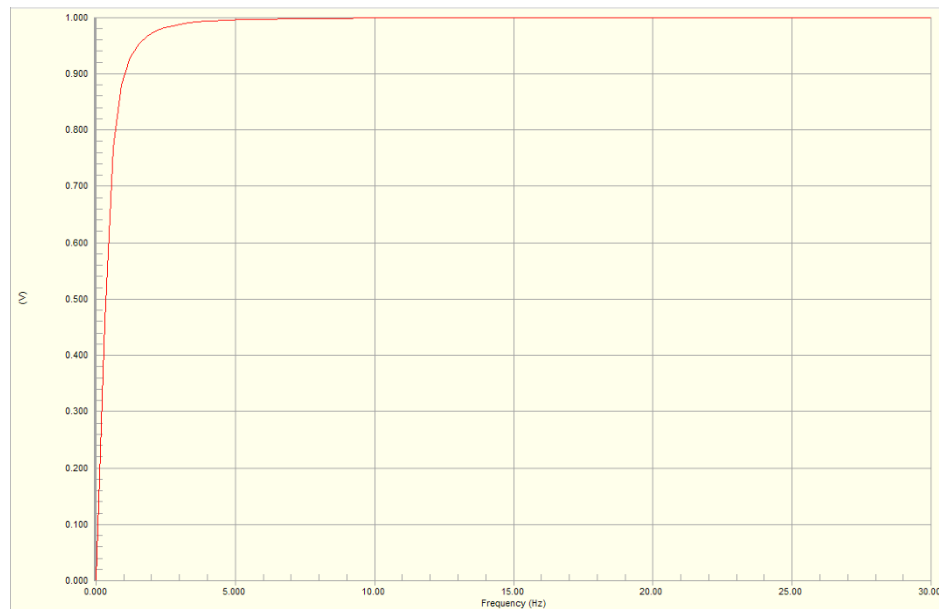


Figura 35. Respuesta en frecuencia de filtro pasa alta con frecuencia de corte de 0.33Hz.



d. **Acondicionamiento de señal.** En esta etapa la tarea que se realiza es amplificar la señal más, y colocarla en el rango de detección del microcontrolador. La amplificación se hace por medio de un circuito amplificador inversor, con una ganancia máxima de 50. Esta ganancia es variable, porque se realiza por medio de un potenciómetro, esto para poder controlarla, según el estado en el que llegue la señal a este punto. Este Circuito se muestra en la Figura 36. Al salir ya amplificada de este punto, se le agrega un offset DC, para poder colocarla en el centro del rango de detección del microcontrolador, esto se realiza con un divisor del voltaje, para obtener un voltaje de -2.5 V, ya que luego se utilizará un sumador inversor, y luego para aislar la impedancia de este divisor, se coloca un seguidor de voltaje. Por último, se utiliza un sumador inversor, para agregar el offset a la señal, y regresarla a su estado original, no el invertido que venía de la última amplificación. Esta etapa completa se puede observar en la Figura 37. La salida de esta etapa es la entrada al ADC del microcontrolador. (EEGinfo, s.f.)

Figura 36. Circuito esquemático de amplificador inversor con ganancia máxima de 50.

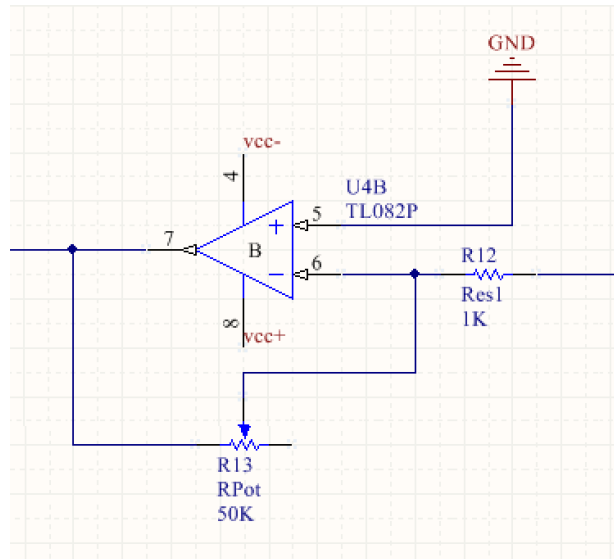
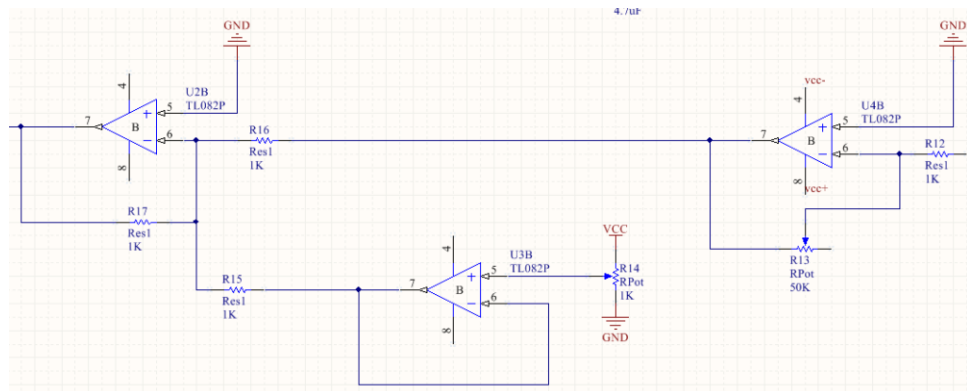
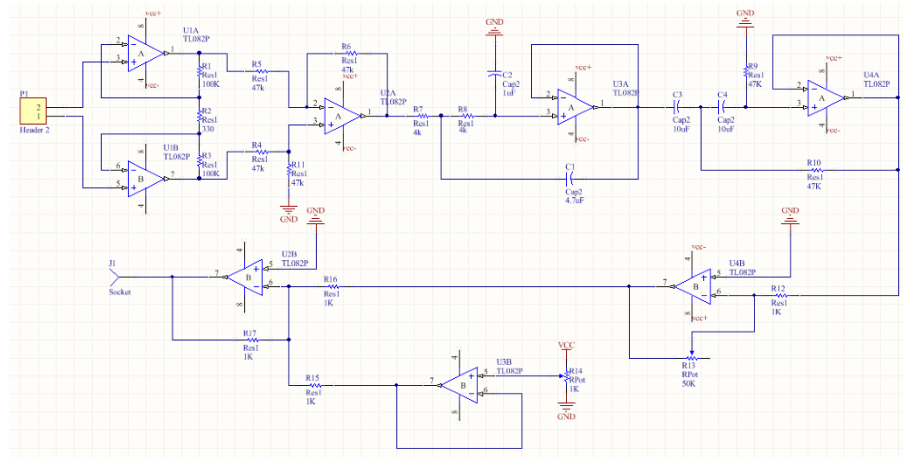


Figura 37. Etapa completa de acondicionamiento.



e. **Diseño de PCB.** El esquemático total de toda la etapa de amplificación y acondicionamiento de señales EEG, se puede observar en la Figura 38. Desde la construcción del Esquemático se comenzó a organizar los componentes de una forma de que se adaptaran mejor en el PCB. El mejor modo que se encontró de utilizar los amplificadores de cada integrado, fue conectar primero todos los del lado derecho de arriba hacia abajo, y luego al llegar al final de ellos, seguir con los del lado izquierdo, de abajo hacia arriba. De este modo sería más fácil seguir el camino de la señal con los tracks. Aunque el primer integrado sí se conectó directo entre él, ya que este se utilizó para la primera etapa del amplificador instrumental.

Figura 38. Esquemático completo de circuito de amplificación y acondicionamiento de las señales EEG.

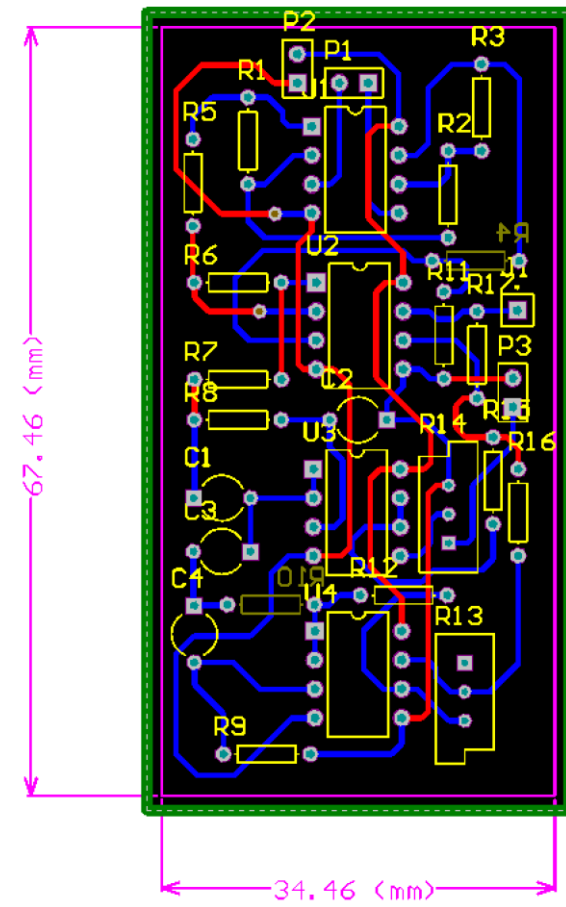


Las restricciones que se utilizaron para el PCB son:

- 1) Ancho mínimo de pista: 0.5mm
- 2) Ancho mínimo entre pistas: 0.5mm
- 3) Tamaño mínimo de pad: 0.5 mm
- 4) Tamaño mínimo entre pads: 0.25 mm
- 5) Tamaño de agujeros: 0.5 mm

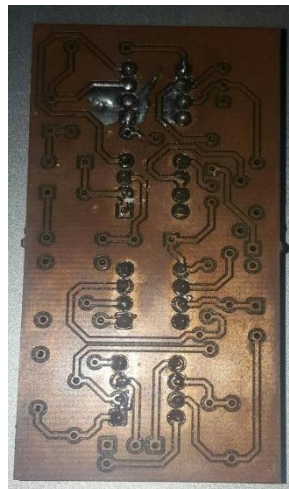
Se utilizaron estas restricciones, ya que eran las que tenía la fresadora con la que se iba a realizar la placa. El ruteo del PCB se realizó utilizando los dos lados de la placa. El diseño final se puede observar en la Figura 39, termino siendo de 67.46mm de alto y 34.46mm de ancho. Las rutas en rojo están en la capa superior, al igual que los componentes. Las rutas en azul, están en la capa inferior.

Figura 39. Diseño final PCB.



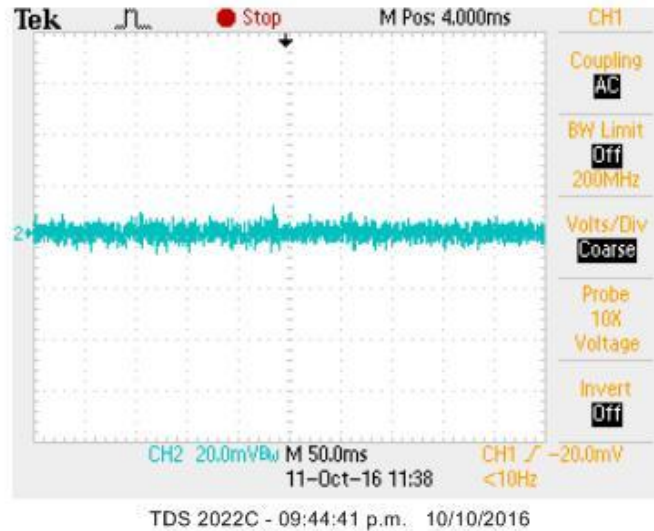
En la Figura 40 se puede observar la placa ya fabricada.

Figura 40. Placa de amplificación de señales EEG.



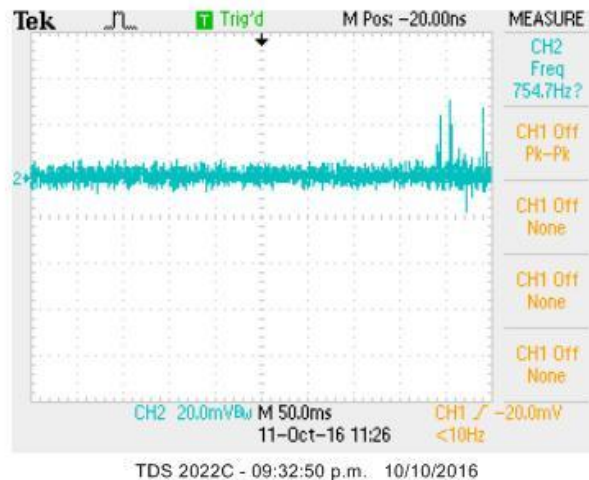
f. **Señales reales.** En la Figura 41 se puede observar la salida del amplificador instrumental al amplificar la salida del electrodo F4. Se puede notar que en esta imagen no se puede diferenciar realmente cual es la forma de la señal, esto debido a que en esta etapa tiene la mayor cantidad de ruido y la componente DC.

Figura 41. Señal de electrodo F4, después de ser amplificada por el amplificador instrumental.



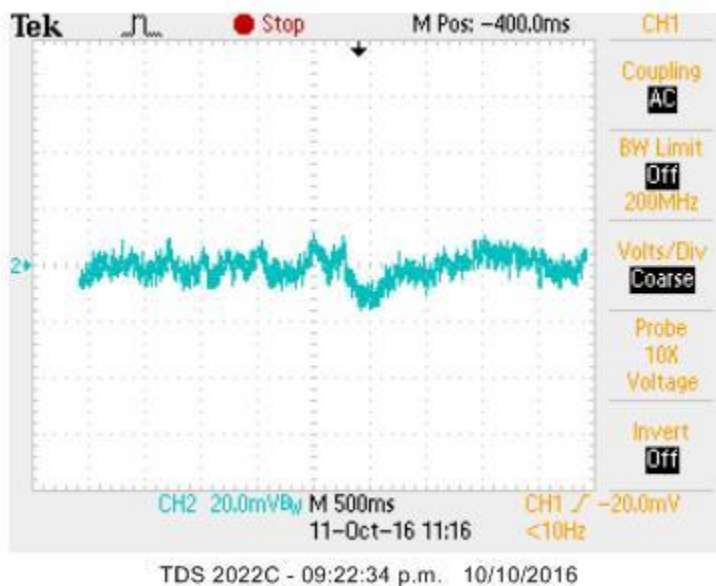
En la Figura 42, se puede observar la salida de la señal después de pasar por el filtro pasa baja, en este punto, pareciera ser igual a la del amplificador instrumental, pero es por la amplitud de las señales y la ganancia DC que tienen, la que no permite una mejor visualización la forma de la señal. Para que la señal pudiera ser vista de forma mejor en las figuras 41 y 42, la señal se colocó en acople AC, para que mostrara la señal sin su componente DC.

Figura 42. Señal amplificada luego del filtro pasa baja.



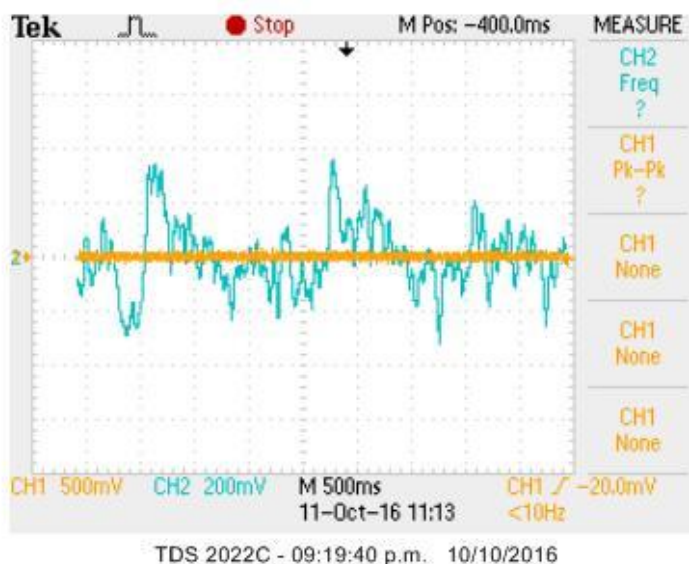
En la Figura 43 se puede observar que la señal ya no tiene componente DC, pero en este caso porque esta es la verdadera forma de la señal, ya que el filtro pasa alto removi6 dicha componente, no porque tenga un acople AC. La forma de esta se1al se puede observar de mejor manera, aunque siempre es de amplitud peque1a.

Figura 43. Se1al luego del filtro pasa alta.



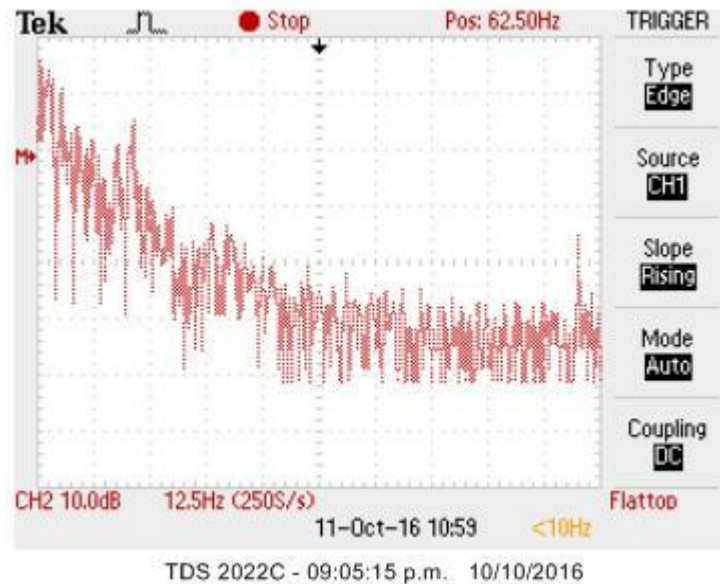
Por ultimo en la Figura 44 se puede observar la se1al ya amplificada por el amplificador inversor. Se puede observar como ya esta se1al, aunque es peque1a se encuentra en un rango de voltaje de cientos de miliVoltios.

Figura 44. Se1al despu6 de amplificarla en la 6ltima etapa.



En la Figura 45 se puede observar las frecuencias que componen la señal final, esto utilizando la herramienta de la transformada rápida de Fourier del osciloscopio. Se puede observar que, aunque si hay ruido de otras frecuencias, las frecuencias que tienen mayor magnitud en este punto son las que están en el rango de interés. Mientras se tomaba las muestras para obtener la transformada de Fourier, se mantuvieron cerrados los ojos, para que la magnitud de las señales fuera mayor.

Figura 45. Frecuencias que componen la señal amplificada final.



## 2. Diseño y fabricación del casco

a. **Diseño del casco.** Para poder diseñar una pieza en Inventor, como primer paso es necesario definir un Sketch base, de donde se partirá para la creación del sólido 3D. El Sketch utilizado para la pieza donde se sostendría toda la estructura, se puede observar en la Figura 46. La pieza resultante de este Sketch es la que contendría todos los circuitos, y de donde partiría la diadema para colocar los electrodos. Las medidas que contiene este Sketch son las que se tomaron para tener una aproximación de las dimensiones de la cabeza. Este Sketch en principio fue realizado para formar una sola pieza, pero por las limitaciones de fabricación, tuvo que ser dividido, para que a partir de él se realizaran tres piezas, para esto tuvo que dividirse con las líneas perpendiculares al contorno. Al salir las tres piezas del mismo Sketch, aseguraba que a la hora de unir las casarían de forma correcta. Aunque para las dos piezas de los extremos que surgían de este Sketch, se tuvo que realizar un cambio al Sketch, ya que a la hora de imprimir la forma de las piezas, para probar que tan bien se adaptaban al cráneo, quedó demasiado distanciado la parte de los lados, esto debido a que se supuso que la parte trasera era todo el semicírculo, mientras solo era una parte de él, por lo tanto, para solucionar este problema, se le realizó unos cambios al Sketch original, este se puede observar en la Figura 47.

Figura 46. Sketch original de diadema horizontal.

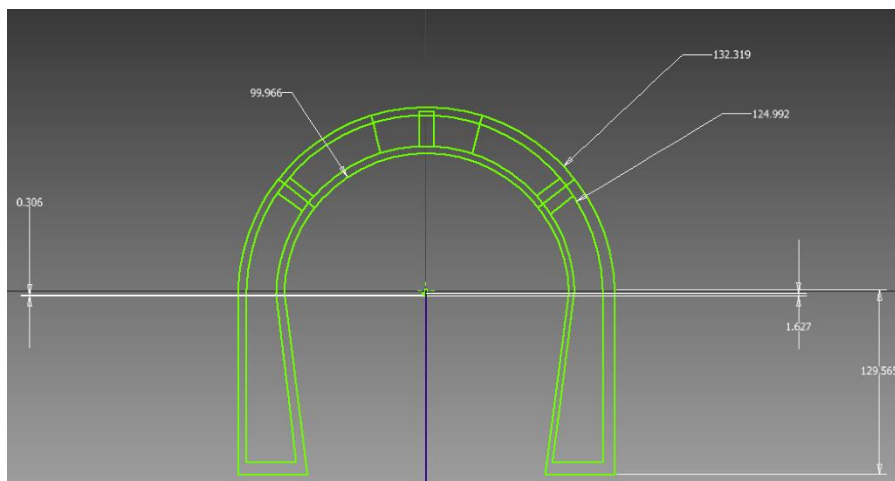
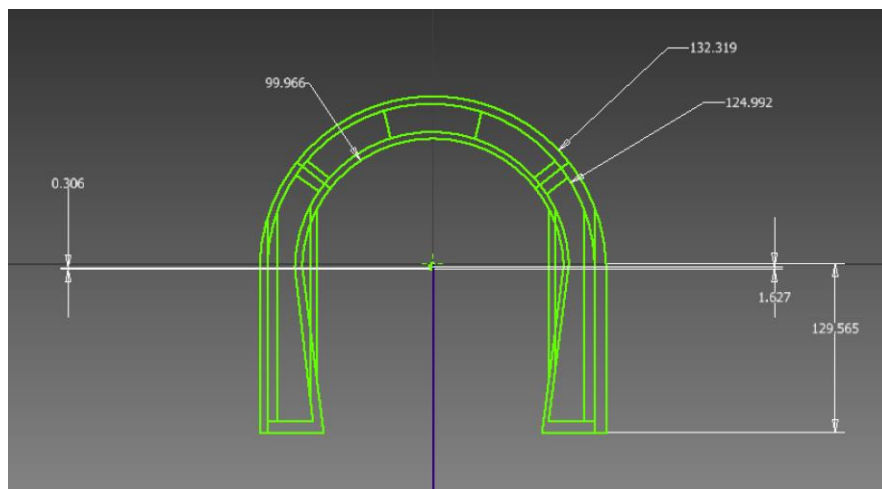


Figura 47. Sketch de diadema horizontal modificado para encajar mejor con la cabeza.



Para realizar la pieza 3D a partir del Sketch, simplemente se utiliza la herramienta de inventor Extrusión, que solo necesita un plano, para levantar un sólido partir de él, con la forma del mismo y una altura determinada. La altura del contorno de la pieza fue de 35mm, esto con la intención de que las placas pudieran caber dentro de él, y de la parte interna de 3mm.

En las figuras 48 a 53, se pueden observar dos vistas diferentes para cada una de las piezas resultantes del Sketch de la Figura 47. La pieza mostrada en las figuras 48 y 49, tiene ese espacio cortado en la parte trasera, para poder hacerla flexible, y que esta sea la parte donde el casco tome su flexibilidad. Esta parte se realizó de esta forma, ya que, al hacerla completo y probar su flexibilidad fue nula. Por lo tanto, se rediseñó de esa forma.

Figura 48. Vista frontal lateral izquierda desde arriba de pieza central.

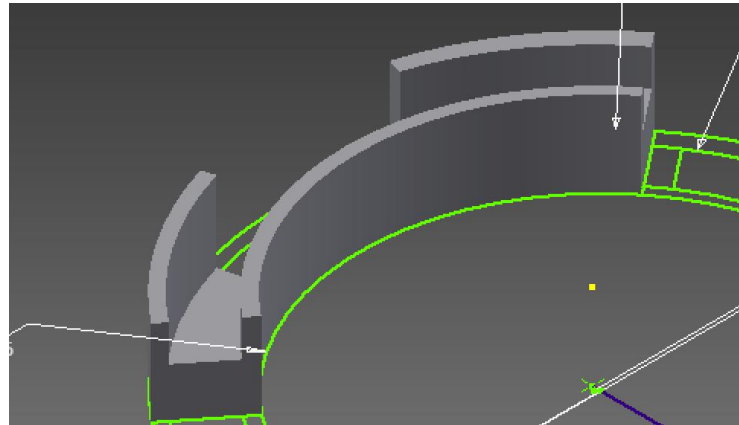


Figura 49. Vista trasera lateral de pieza central.

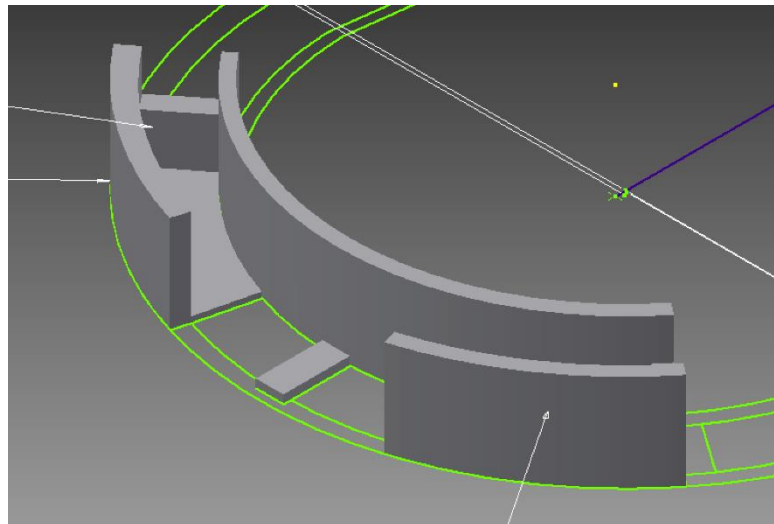


Figura 50. Vista frontal lateral derecha desde arriba de pieza del lado derecho.

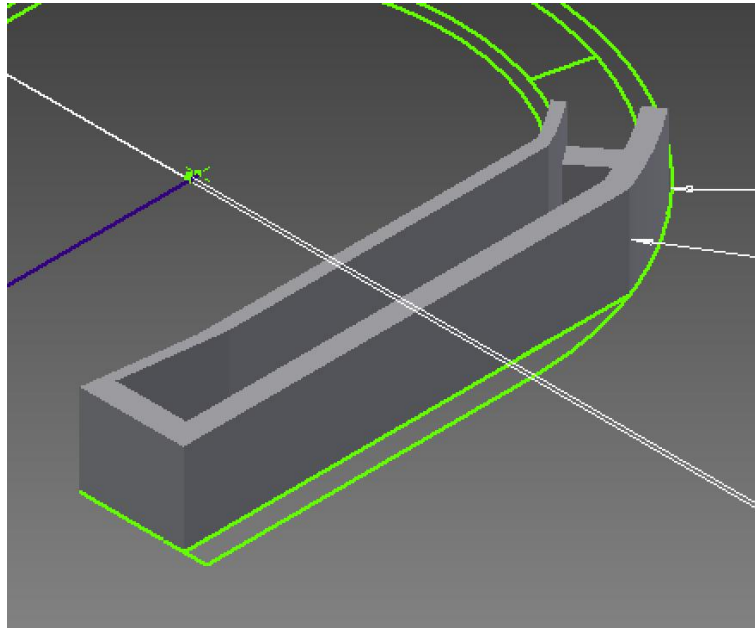


Figura 51. Vista trasera lateral izquierda desde arriba de pieza del lado derecho.

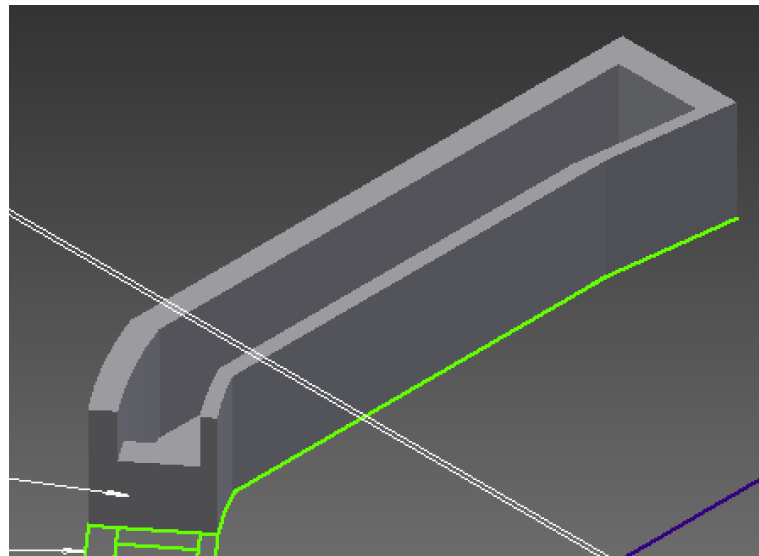


Figura 52. Vista frontal lateral izquierda desde arriba de pieza del lado izquierdo.

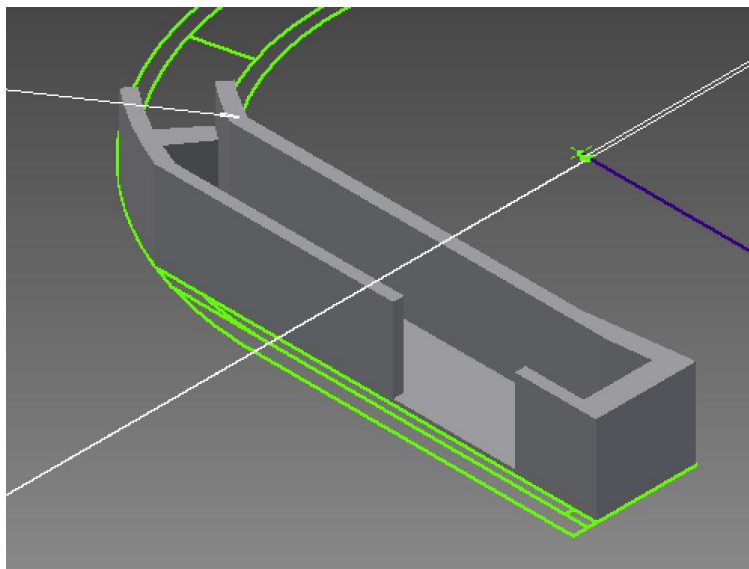
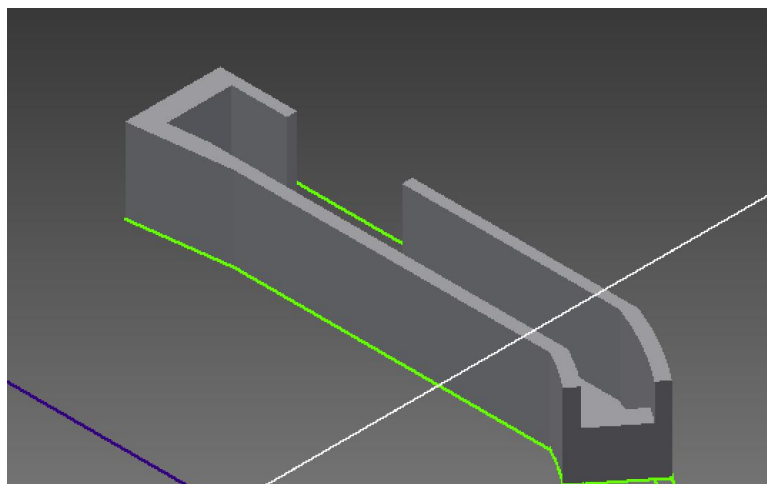


Figura 53. Vista trasera lateral derecha desde arriba de pieza del lado izquierdo.



Para poder colocar los electrodos F3 y F4, se diseñó otra diadema, solo que estaría de forma vertical, con la intención de atravesar la cabeza en la línea donde se encuentran los electrodos deseados, esta diadema partiría de la diadema horizontal. Esta diadema es mucho más simple, ya que su única función es sostener los electrodos, los cuales son bastante pequeños. Para realizar esta pieza se tomó como base el radio interior del Sketch mostrado en la Figura 47, ya que se había notado que esta también se acomodaba de forma correcta a esta parte de la cabeza. Esta pieza se puede observar en la Figura 54. Esta pieza tiene una altura de 1.15 cm.

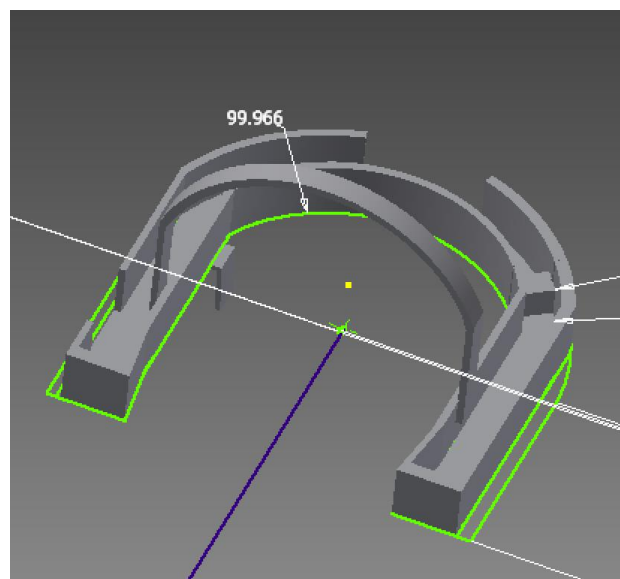
Figura 54. Vista frontal lateral derecha de la diadema para colocar los electrodos F3 y F4.



Por último, se realizó una pieza en forma de escuadra, la cual se colocaría del lado derecho y del lado izquierdo, en la parte inferior de la diadema horizontal. Esta pieza sirve para colocar los electrodos de referencia.

El ensamble de todas las piezas utilizando la opción de inventor para ensamblar piezas, se puede observar en la Figura 55.

Figura 55. Ensamble de piezas de casco.



b. **Fabricación del casco.** Para fabricar el diseño anterior, se utilizó la impresora 3D, para poder imprimir cada pieza. La ya tener todas las piezas impresas, se unieron utilizando pegamento instantáneo y bicarbonato de sodio, ya que, al pegar las piezas de esta forma, queda totalmente solida la unión. Para poder colocar la diadema vertical en su posición correcta se utilizó el casco Emotiv EPOC como referencia para saber en dónde deberían ir los electrodos. Se hizo diferentes pruebas con diferentes personas antes de pegar esta pieza, para asegurar que los electrodos se mantuvieran en el lugar correcto dependiendo la cabeza. En la Figura 56 se puede observar la diadema horizontal completa. En la Figura 57 se puede observar el casco completo, y en la Figura 58 se puede observar el casco colocado sobre una persona.

Figura 56. Diadema horizontal.

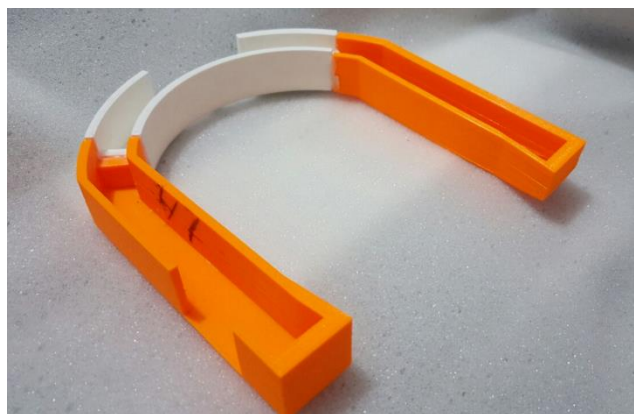


Figura 57. casco completo

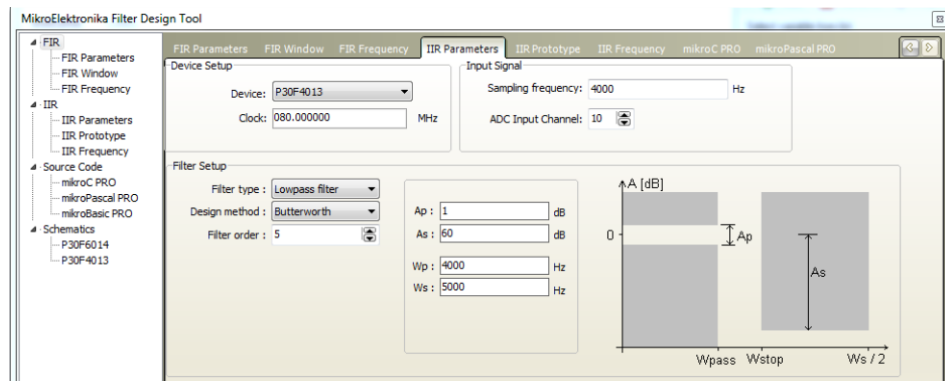


Figura 58. casco colocado sobre persona



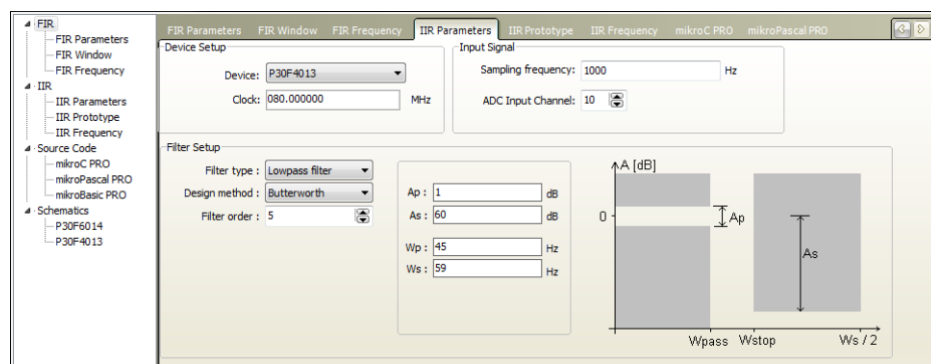
3. Filtrado de señales. Para diseñar el filtro que se utilizaría, se utilizó la herramienta de diseño de filtros de MikroC para dspic. En esta herramienta se pueden diseñar ya sean filtros FIR o IIR, al final de diseñarlos, esta entrega un código con la implementación del mismo. Desde un principio se decidió que se implementaría un filtro IIR, para poder obtener un filtro de orden menor. En la Figura 59 se muestra la interfaz de esta herramienta para diseñar un filtro IIR. En la ventana mostrada en la Figura 59, se muestra todos los parámetros que solicita para poder diseñar el filtro. Ya que genera el código del filtro el microcontrolador sobre el cual se implementará, y con qué frecuencia de reloj estará corriendo. También solicita el canal ADC que se utilizara para obtener la señal, la frecuencia de muestreo que se desea utilizar. Para el diseño del filtro, solicita que tipo filtro se desea implementar, pasa baja, pasa alta, pasa banda o rechaza banda. Luego se ingresa a partir de que prototipo analógico se desea generar el filtro digital, solo contiene tres opciones, Butterworth, Chevyshev 1 y Chevyshev 2. Por último, si es un pasa baja o pasa alta solicita ingresar las frecuencias de corte y de paso, y la atenuación que se desea tener en la frecuencia de corte.

Figura 59. Herramienta de MikroC para diseño de filtros IIR.



Para implementar el filtro se utilizó el microcontrolador DSPIC30F4013, a una frecuencia de operación de 80MHz. La señal se obtuvo del canal 10, que es el pin 10 del puerto B. El prototipo escogido fue el Butterworth, ya que este no tiene oscilaciones en su banda de paso. El orden del filtro fue 5, esta decisión se tomó debido a que, si el filtro era de un orden menor, atenúa cada vez más las frecuencias de interés, y aunque el mayor propósito de este filtro es eliminar de 60Hz, se desea mantener la magnitud de las frecuencias de interés, que están por debajo de los 35Hz. Y aunque un filtro de orden alto pareciera aportar una banda de transición más angosta, un filtro IIR de orden alto tiende a ser inestable. Y la banda de transición se puede mejorar al colocar varios filtros en cascada. La frecuencia de paso que se colocó fue 45Hz, para que no se atenuaran las frecuencias de interés, se coloca más alto, porque para el diseño del filtro esta herramienta se enfoca más en cumplir con la frecuencia de corte que con la de paso, aunque tampoco es que cumpla con la atenuación e 60dB a la frecuencia de corte especificada. Los parámetros que se utilizaron para diseñar el filtro, se muestra en la Figura 60.

Figura 60. Parámetros acogidos para filtro IIR a implementar.



En la pestaña “IIR Prototype”, se puede observar la función de transferencia del filtro que implementara, también la función de transferencia del prototipo analógico que se utilizó, en la Figura 61 se puede observar el resultado de esta ventana para el filtro que se implementó. La pestaña “IIR Frequency” muestra la respuesta en frecuencia del filtro que se implementara, en la Figura 62 se puede observar la respuesta en frecuencia del filtro que se implementara. Con el cursor se puede mover a través de este

diagrama, y observar cual es la ganancia que se tiene a cierta frecuencia. Al colocarse en la frecuencia de 60 Hz, este filtro la atenúa a un 22% de su magnitud y en 35Hz se encuentra a un 98% de su magnitud. Por último, en la pestaña “MikroC PRO” se encuentra el código que implementa este filtro. Del código que se otorga en esta ventana lo más importante son los coeficientes del filtro y como los implementa. Ya que para implementar los coeficientes del filtro mostrados en la Figura 61, necesita escalarlos, para que todos estén en el rango que puede representar el microcontrolador, ya que por el formato que utiliza el microcontrolador para número de punto flotante, no se puede representar cualquier número. Por lo tanto, el código otorga un vector con los coeficientes ya en la notación de punto flotante del microcontrolador, y el factor de escalamiento que se utilizó para cada conjunto de coeficientes, la forma en que se da estos valores se muestra en la Figura 63. En la Figura 64 se muestra la forma en que se implementa el filtro IIR, utilizando la función de MikroC para filtros IIR, en esta imagen se puede observar cómo es que realiza el escalamiento de los coeficientes. La función que implementa el filtro IIR y los parámetros que recibe, se muestran en la Figura 65. La función “IIR\_Radix”, recibe los coeficientes y su factor de escalamiento, el orden del filtro, los vectores de salida y entrada, y el tamaño de ellos, esta función devuelve el valor actual de la salida.

Figura 61. Función de transferencia de filtro a implementar.

The screenshot shows the following transfer functions:

Analog prototype filter :  

$$H_a(s) = \frac{B_a(s)}{A_a(s)} = 1.0000 * \frac{1}{(s+0.3090 -j*0.9511)*(s+0.8090 -j*0.5878)*(s+1.0000)*(s+0.8090+j*0.5878)*(s+0.3090+j*0.9511)}$$

Analog filter :  

$$H(s) = \frac{B(s)}{A(s)} = 0.0001 * \frac{1}{(s+0.0440 -j*0.1354)*(s+0.1151 -j*0.0837)*(s+0.1423)*(s+0.1151+j*0.0837)*(s+0.0440+j*0.1354)}$$

Digital filter :  

$$H(z) = \frac{B(z)}{A(z)} = \frac{0.0002*z^1+0.0004*z^2+0.0004*z^3+0.0002*z^4}{1.0000*z^0-4.0859*z^1+6.7476*z^2-5.6213*z^3+2.3598*z^4-0.3990*z^5}$$

Figura 62. Respuesta en frecuencia de filtro a implementar.

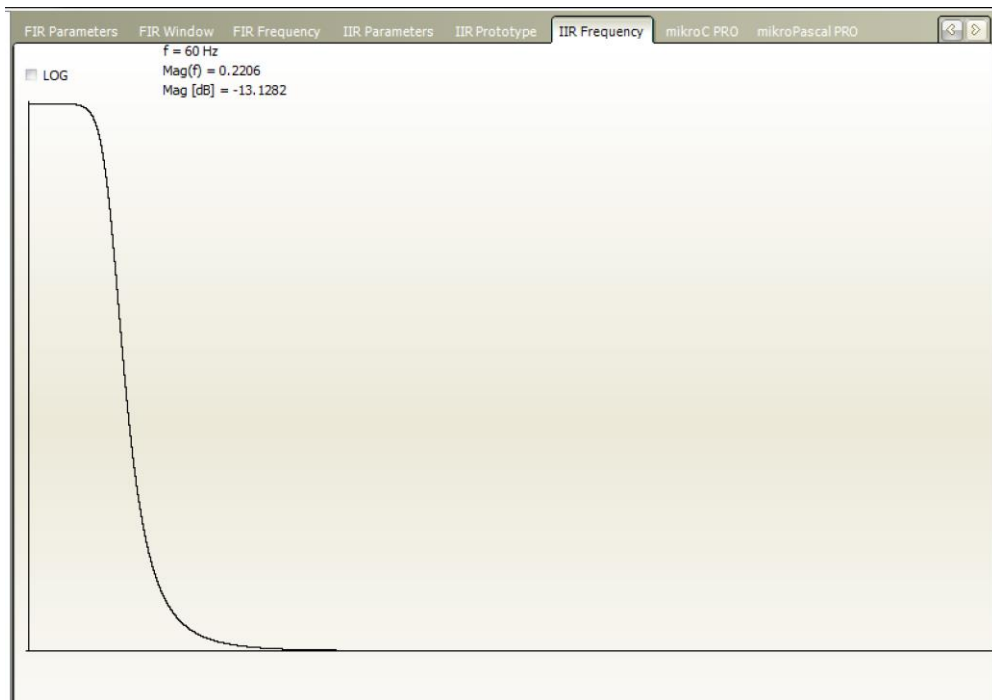


Figura 63. Coeficientes de filtro escalados y su factor de escalamiento.

```

const unsigned int
  COEFF_B[FILTER_ORDER+1] = {0x09AB, 0x3058, 0x60B0, 0x60B0, 0x3058, 0x09AB};
const unsigned int
  COEFF_A[FILTER_ORDER+1] = {0x1000, 0xBEA0, 0x6BF6, 0xA60F, 0x25C2, 0xF99E};

const unsigned int
  SCALE_B      = 11;
const unsigned int
  SCALE_A      = -3;

```

Figura 64. Ecuación de diferencias implementada por la función "IIR\_Radix" de MikroC.

$$y[n] = \frac{1}{2^{Bscale}} * \sum_{k=0}^N (Bcoef[k] * x[n-k]) - \frac{1}{2^{Ascale}} * \sum_{k=1}^M (Acoef[n] * y[n-k])$$

Figura 65. Función que implementa filtro IIR, y sus parámetros de entrada.

```

CurrentValue = IIR_Radix( SCALE_B,
                          SCALE_A,
                          COEFF_B,
                          COEFF_A,
                          FILTER_ORDER+1,
                          input,
                          BUFFER_SIZE,
                          output,
                          inext);

```

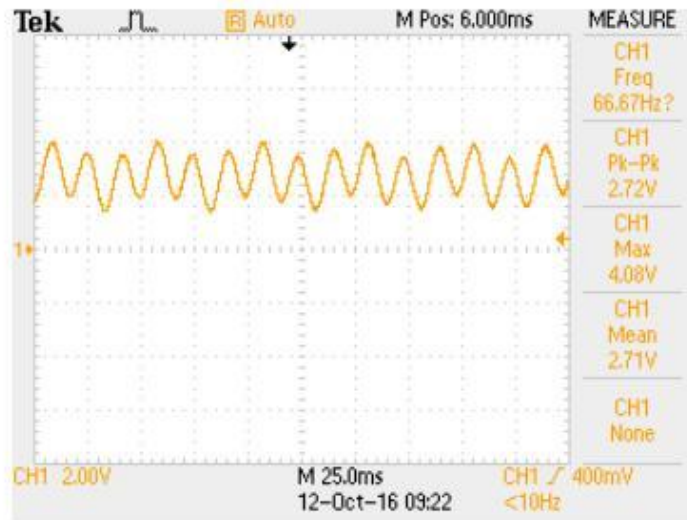
Al código generado por MikroC, se le debió remover fragmentos, ya que este código asume que la salida será enviada a un DAC, por medio del protocolo SPI, lo cual es innecesario, ya que el objetivo es enviarlo a la computadora. Por lo tanto, en lugar de esto, se agregó el envío del valor actual a la computadora utilizando el protocolo UART. Para enviar cada dato era necesario hacerlo en dos partes, ya que cada valor generado era un entero en 16bits, y el protocolo serial, solo envía de Byte en Byte. Por lo tanto, primero se envió el Byte que contenía los bits más significativos y luego el Byte con los menos significativos.

Para poder saber el tiempo de ejecución del filtro, se colocó que se encendiera el puerto D antes de filtrar y generar el nuevo valor, y se apaga al ya tener el nuevo valor. Esto es de utilidad para saber es lo suficientemente rápido para poder realizar la operación en el periodo de muestreo.

Para colocar los filtros en cascada, simplemente el vector de salida de uno es el de entradas del siguiente, ya que los coeficientes son los A y B son los mismos. La ventaja de colocar los filtros en cascada es que se puede generar un filtro de orden mayor, manteniendo la característica de uno de orden pequeño.

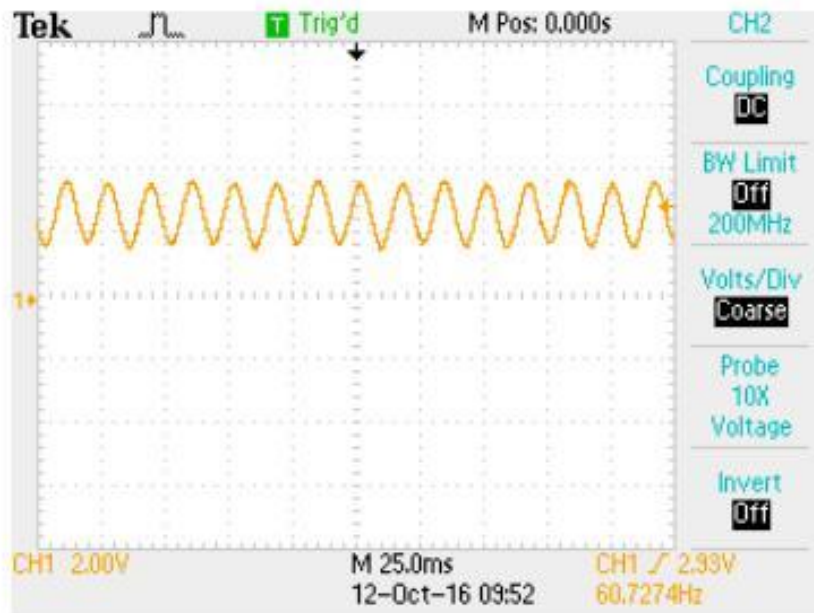
Para poder verificar que filtro usar, se implementó el filtro con los coeficientes mostrados en la Figura 63, y a partir de este se realizó otro de orden 25, colocando 3 en cascada y un de 25, colocando 5 en cascada. El rendimiento de estos filtros fue probado con tres señales diferentes, la primera fue una señal que sumaba una señal de 60Hz de 1 Voltio de amplitud y una señal de 22Hz de 500 miliVoltios, esta se puede observar en la Figura 66. La segunda señal fue era la suma de una de 60 Hz de 1 Voltio de amplitud y una de señal de 22 Hz de 100 miliVoltios de amplitud, esta se puede observar en la Figura 67. La última señal de prueba era la suma de una señal de 60Hz de 1 Voltio de amplitud y una señal de 22 Hz de 1 Voltio de amplitud, esta se puede observar en la Figura 68. Todas las señales estaban centradas en 2.71 Voltios.

Figura 66. Señal de 60 Hz de 1 Voltio de amplitud sumada a señal de 22 Hz de 0.5 Voltios de amplitud.



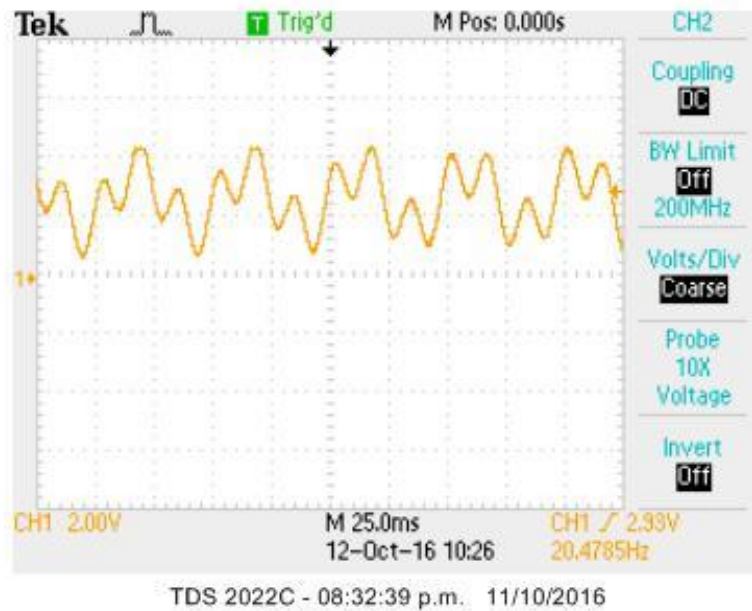
TDS 2022C - 07:28:44 p.m. 11/10/2016

Figura 67. Señal de 60 Hz de 1 Voltio de amplitud sumada a señal de 22 Hz de 0.1 Voltios de amplitud.



TDS 2022C - 07:58:10 p.m. 11/10/2016

Figura 68. Señal de 60 Hz de 1 Voltio de amplitud sumada a señal de 22 Hz de 1 Voltio de amplitud.



Los tres filtros propuestos fueron probados con cada una de las señales mostradas en las figuras 66 a 68. En las figuras 69 a 71, se puede observar la respuesta de los tres filtros a la señal mostrada en la Figura 66. En las figuras 72 a 74, se puede observar la respuesta de los tres filtros a la señal mostrada en la Figura 67.

Figura 69. Respuesta del filtro de orden 5 a la señal de prueba 1.

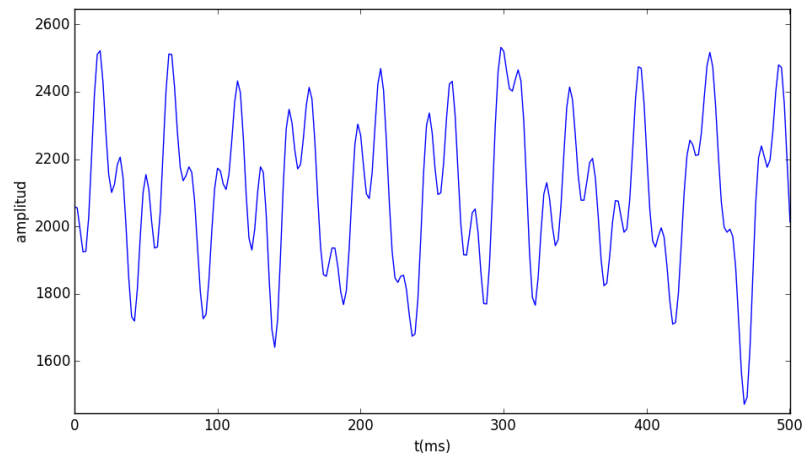


Figura 70. Respuesta del filtro de orden 15 a la señal de prueba 1.

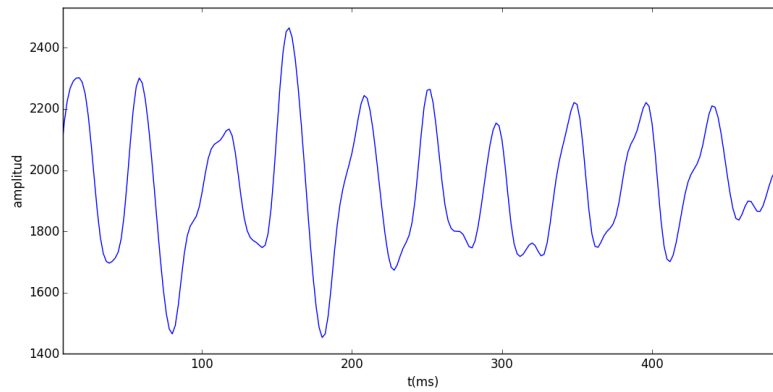
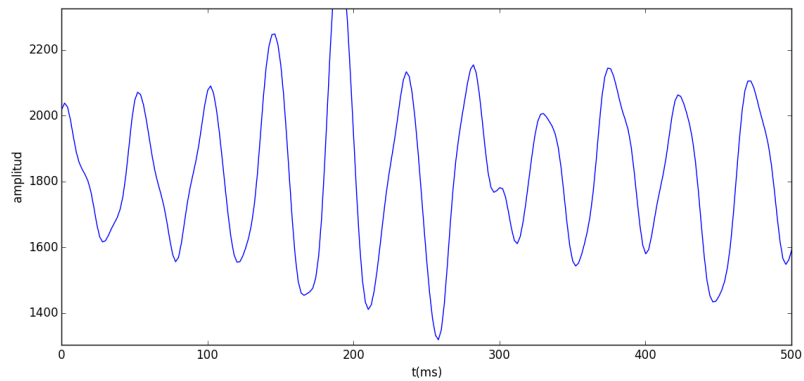


Figura 71. Respuesta del filtro de orden 25 a la señal de prueba 1.



En las figuras 72 a 74, se puede observar la respuesta de los 3 filtros a la señal mostrada en la Figura 67.

Figura 72. Respuesta del filtro de orden 5 a la señal de prueba 2.

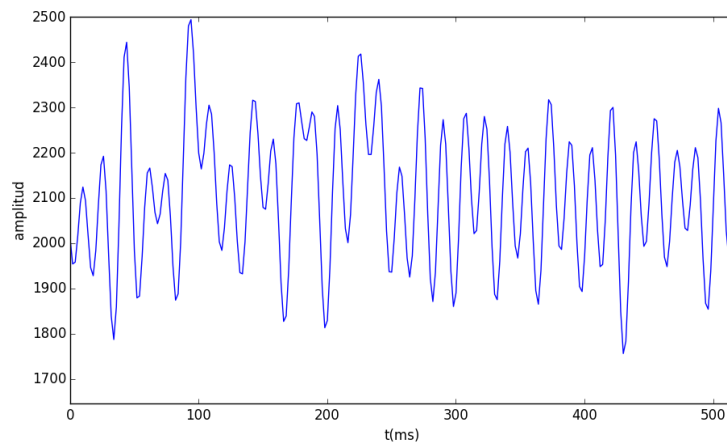


Figura 73. Respuesta del filtro de orden 15 a la señal de prueba 2.

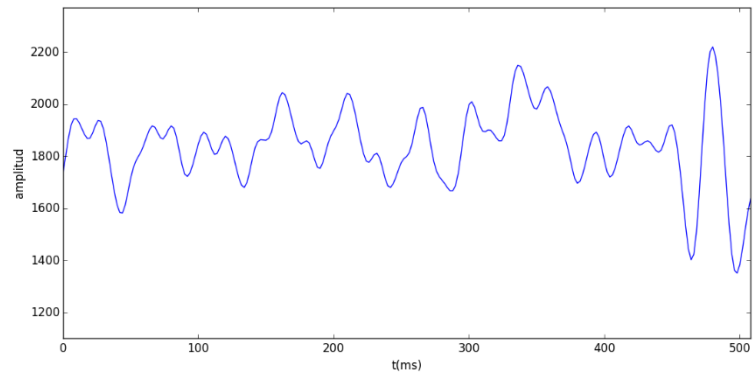
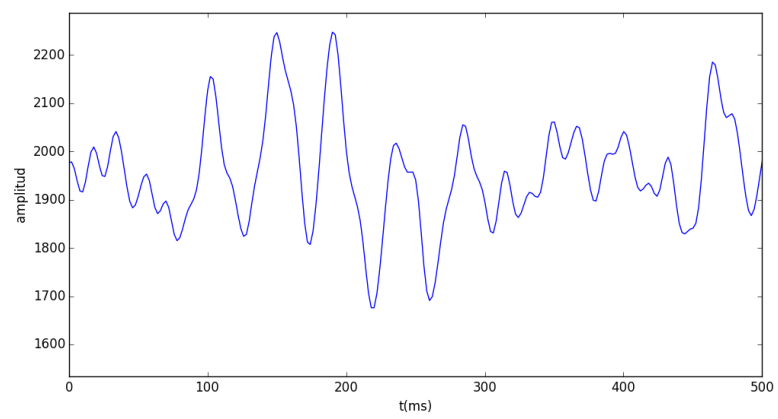


Figura 74. Respuesta del filtro de orden 25 a la señal de prueba 2.



En las figuras 75 a 77, se puede observar la respuesta de los tres filtros a la señal mostrada en la Figura 69.

Figura 75. Respuesta del filtro de orden 5 a la señal de prueba 3.

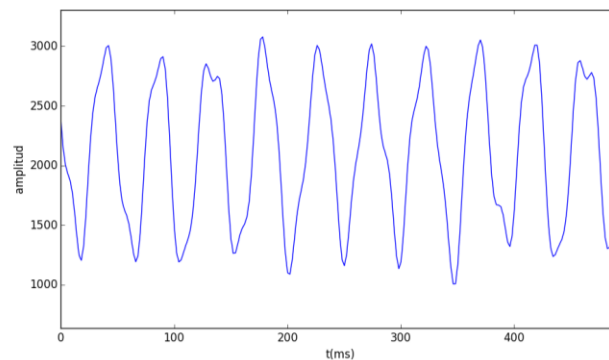


Figura 76. Respuesta del filtro de orden 15 a la señal de prueba 3.

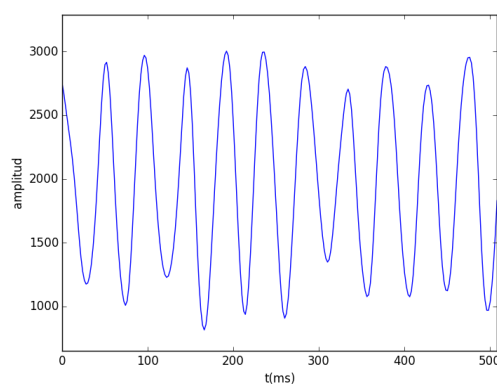
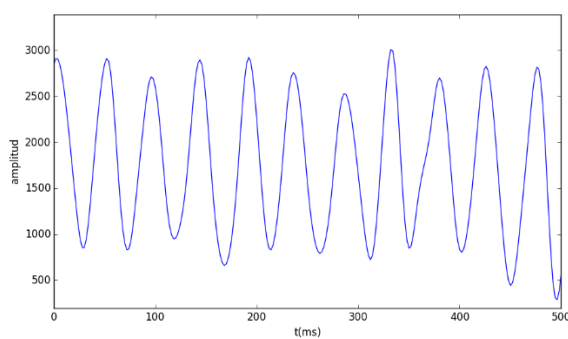
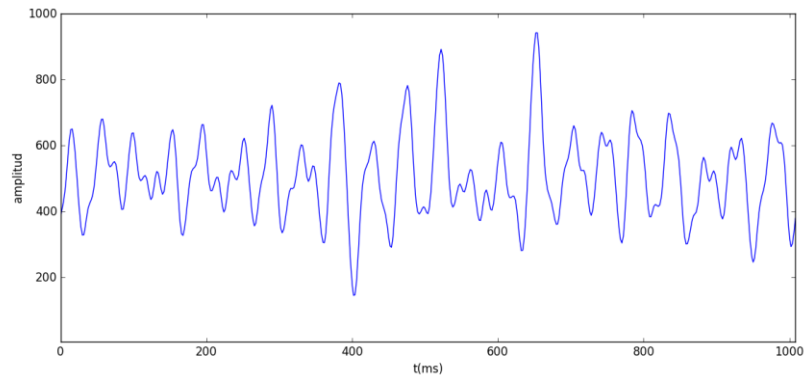


Figura 77. Respuesta del filtro de orden 25 a la señal de prueba 3.



El tiempo del filtro de orden 5 es de 6.8 microsegundos, el del filtro de orden 15 es de 18 microsegundos y el del filtro de orden 25 es de 27 microsegundos. El tiempo de ejecución se obtuvo midiendo el ancho del pulso que se daba en el puerto D cada vez que se obtenía una nueva salida. El filtro escogido fue el de orden 25, con este se recopiló la señal EEG proveniente de la etapa de amplificación, en la Figura 78 se muestra un fragmento de la recopilación que se realizó de la señal EEG.

Figura 78. Señal EEG proveniente de electrodo F4, digitalizada y filtrada.



## D. Discusión.

Los resultados obtenidos en la sección de amplificación y acondicionamiento de la señal tienen una gran relación con los realizados en la siguiente sección, ya que como se mencionó, la calidad de la señal está altamente relacionada con la calidad de contacto de los electrodos, ya que el no colocar de la forma más óptima los electrodos, genera que entre el electrodo de referencia y el activo, no haya el mismo ruido común, y es por eso que aunque el amplificador instrumental debería solo amplificar la señal de interés, termina amplificando la señal de ruido, y es lo que termina limitando la cantidad de ganancia que se obtiene con este circuito, ya que si el ruido común que no se elimina tiene una magnitud mayor a la de la señal de interés, al amplificarlo, se puede saturar los amplificadores operacionales, tal como se mencionó. Es por esto que al momento de colocar el casco en la persona que lo utilizara se debe verificar que los electrodos activos y de referencia estén teniendo buen contacto, y se debe humectar con solución salina a la esponja que recubre al electrodo, para mejorar la conducción eléctrica del medio. Esta mejora de la conductividad también ayuda a disminuir la componente DC con la que viene la señal, ya que esta depende bastante de la forma en que hace contacto el electrodo con la cabeza, a esta componente DC se le denomina tensión galvánica, la cual es importante disminuir lo más posible, ya que esto ayuda a prevenir que el amplificador operación se sature por esta componente.

Otro punto bastante importante para la obtención de una buena señal, es siempre mantener la referencia a tierra de todas las fuentes de voltaje conectada, en este caso el cuerpo es una fuente de voltaje, el cual necesita estar unido a la tierra del circuito, para que todas las señales tengan la misma referencia eléctrica.

Para la etapa del filtro antialias, se obtuvo un resultado bastante bueno, ya que como se puede observar en la Figura 33, en este filtro se comienza a atenuar la frecuencia de 60Hz, ya que esta se atenúa en un 33% de la amplitud con la que entra. Otro factor importante que se puede observar en la Figura 33, es que en la banda de paso se encuentra una elevación, que amplifica las frecuencias de interés. Este comportamiento se obtuvo al buscar tener un factor Q mayor, ya que este los cambios que realiza en la respuesta frecuencial del filtro, es que disminuye la longitud de la banda de transición, a cambio de introducir una elevación antes de

llegar a la frecuencia de corte. Esta Elevación en la banda de paso, muchas veces no es deseada, ya que introduce una distorsión en magnitud a las señales en la banda de paso. Pero en este caso se aprovechó este comportamiento totalmente, ya que esta elevación en la parte final de la banda de paso, amplifica las frecuencias que tienen una menor magnitud, que son las ondas Beta, y como la elevación se da en la mayoría de la banda de paso, excepto en las frecuencias más pequeñas, que son las que tienen mayor magnitud. Aunque se tuvo el suficiente cuidado, para escoger este factor, ya que se podría pensar que elevar muchísimo más este factor sería lo mejor, pero este tipo de filtros utilizados no son aptos para tener un factor de calidad muy alto, ya que con un factor Q muy alto introducen oscilaciones en la banda de paso, pero ya son bastante considerables, lo cual si introduciría distorsión en la señal de interés, y podría afectar las aplicaciones en las que se use.

Al observar las figuras las figuras 41 y 44, se puede observar cual fue el rendimiento real del circuito con las señales EEG. Al observar la Figura 44, se puede notar que la señal resultante después de todas las etapas, es bastante buena, ya que el ruido que contiene es pequeño comparado con la magnitud de la señal de interés. Esto se pudo lograr al colocar los electrodos de una forma correcta, y agregarles solución salina, para que hicieran un mejor contacto eléctrico. Se puede saber que fue esto lo que ayudo, porque al colocar de forma correcta los electrodos, en la primera etapa, la del amplificador instrumental, el ruido común de 60Hz fue eliminado en gran medida, de otro modo en la salida de esta etapa, se hubiera tenido una señal con amplitud mayor y con una predominancia de la frecuencia de 60Hz. Esto fue un punto crítico para poder llegar al resultado mostrado en la Figura 44. En la Figura 45 se puede observar cuales son las frecuencias que componen a las señales amplificadas finales, en donde se puede notar que las que tienen mayor amplitud están por debajo de los 37Hz, esto se sabe porque la escala en frecuencia es de 12.5Hz, lo cual significa que cada cuadrado contiene 12.5Hz.

El casco final mostrado en la Figura 57, coloca los electrodos en el lugar donde se espera, aunque si es adaptable a diferentes cabezas, si introduce cierto error en la posición de los electrodos según la variación que haya entre los tamaños de cabeza. Ya que como se mencionó estos siguen el sistema de posicionamiento 10-10, el cual establece que los electrodos se colocan a una distancia normalizada por las dimensiones de la cabeza donde se colocan. Al dejarlos fijos, no se está siguiendo con exactitud esto, ya que, para colocarlos en el lugar correcto, las cabezas deben ser del mismo tamaño. Aunque este error para este caso sería notorio, si se compara la cabeza de un adulto con la de un niño, pero para diferentes adultos, el error introducido será mínimo, y el error introducido no será de gran impacto en la aplicación, ya que lo que se busca es obtener las señales electroencefalográficas generadas en la parte central del área frontal y prefrontal, y los electrodos siempre se mantienen en estas posiciones.

En el primer escenario de las señales de prueba para los filtros, la señal que se desea sea eliminada, tiene el doble de magnitud que la señal de interés, en este caso como se puede observar en la Figura 69, el rendimiento del filtro de orden 5 no es de gran calidad, ya que la señal después de ser filtrada, aún tiene gran cantidad de ruido. El resultado del filtro de orden 15 mostrado en la Figura 70, tiene un mejor rendimiento,

ya que la señal se puede notar más integra, más hay puntos donde se nota cierta distorsión. El resultado del filtro de orden 25 mostrado en la Figura 71, que comparado con el filtro de orden 15, se tienen menos distorsiones. Por lo tanto, para la primera señal la diferencia entre un filtro de orden 25 y el de orden 5 es grande, aunque entre el de orden 25 y 15 no es mucha, pero sí se obtiene una mejor señal con el filtro de orden 25.

Para el segundo escenario, se tiene el peor de los casos probados, ya que en este caso el tamaño de la señal que se desea eliminar es diez veces mayor a la señal de interés. Para este caso el rendimiento del filtro de orden 5 mostrado en la Figura 72, fue bastante malo, ya que como se puede ver la señal de 60 Hz es la más notoria en la señal recuperada, lo cual significa que esta fue atenuada muy poco, por lo tanto, acá la relación entre la señal y el ruido, está muy baja, ya que predomina el ruido. Para los filtros de orden 15 y 25, como se puede observar en las figuras 73 y 74, no es tan bueno, ya que la señal recopilada sí tiene bastantes perturbaciones, pero la señal de interés es la que predomina. Aunque la señal aun con el filtro de orden 25 no fue obtenida con la mejor calidad, ya que el ruido restante sí atribuye bastantes perturbaciones, el resultado no se puede tomar como malo, ya que la señal de entrada tiene una señal de ruido que es 10 veces mayor a la señal de interés, lo cual es una relación señal-ruido bastante pequeña, y la señal resultante terminó al menos con una relación señal ruido mayor a uno, indica que el filtro sí tuvo gran impacto en la señal de entrada.

Para el último escenario, se tiene el mejor de los casos probados, ya que en este caso las señales de ruido tienen la misma magnitud que la señal de interés. La respuesta del filtro de orden 5 para esta señal se muestra en la Figura 75, la cual se puede observar que, aunque no es perfecta la señal, las perturbaciones que tienen la señal filtrada son mínimas. En la respuesta del filtro de orden 15 mostrada en la Figura 76, se puede observar que la señal pareciera no tener ruido alguno, solo algunas elevaciones del voltaje en ciertos puntos. Para la respuesta del filtro de orden 25 mostrada en la Figura 77, se puede notar que al igual que la respuesta del filtro de orden 15, la señal pareciera no tener ruido, aunque en esta respuesta, la señal se mantiene con una amplitud más constante.

Después de realizar las pruebas de los filtros propuestos, se escogió el filtro de orden 25, ya que fue el que tuvo el mejor rendimiento en los tres escenarios, y aunque si toma más tiempo de procesamiento, no es un tiempo crítico, ya que la diferencia entre el de orden 5 y el de orden 25 son aproximadamente 20 microsegundos, lo cual para esta aplicación no termina siendo de gran relevancia, ya que para poder cumplir con la frecuencia de 4KHz, se tiene un tiempo de 250 microsegundos. Y aun en el peor de los casos, se tarda aproximadamente un décimo del tiempo disponible.

No se probó con filtros de orden mayor a 25, porque se notó que la diferencia en los resultados del filtro de orden 15 y 25 no son tan notorios, lo cual indica que aunque se suba más el orden del filtro, los resultados no tendrán una gran mejora, y el subirlo no solo involucra el tiempo de procesamiento, sino también los recursos del microcontrolador consumidos por cada filtro, como lo es la memoria, y tomando en cuenta que no es solo un filtro el que se debe implementar, y que se debe dejar abierta la puerta a la implementación de más canales, es mejor mantener un buen margen de los límites. También hay que tomar en cuenta, que entre

mayor es el orden del filtro mayor es el desfase que le agrega a la señal, y ya que no hay una gran motivación por obtener mejores resultados, no valdría la pena agregar más fase a las señales.

## E. Conclusiones

1. Se amplificó y se acondicionó las señales Electroencefalográficas provenientes de un electrodo, para poder ser detectadas en el rango del microcontrolador DSPIC30F4013.
2. Se fabricó un casco adaptable a diferentes personas, en donde van montados los electrodos para colocarlos en las posiciones F3, F4, AF3 y AF4. Este casco también almacena todos los circuitos de amplificación y el microcontrolador.
3. Se muestrearon y filtraron señales provenientes de cuatro electrodos, a una frecuencia de muestreo de 1KHz para cada canal analógico.
4. Utilizando el microcontrolador DSPIC30F4013 se implementó un filtro digital IIR pasa baja de orden 25, con frecuencia de corte de 60 Hz.
5. Se removió el ruido que acompañaba a las señales electroencefalográficas, principalmente el ruido a 60Hz.
6. Se enviaron las señales electroencefalográficas filtradas por medio del protocolo serial a la computadora a un Baudrate de 425,000.

## F. Recomendaciones

1. Para poder obtener una señal electroencefalográfica de buena calidad, se recomienda humectar bien las esponjas que recubren los electrodos.
2. Para colocar los electrodos en la posición correcta, se recomienda siempre tomar como referencia los electrodos que van en la parte trasera de la oreja, y tener la parte de atrás del casco pegada a la cabeza.
3. Para trabajar con señales electroencefalográficas se recomienda trabajar los circuitos en un pcb, ya que este aporta menos ruido a la señal.
4. Se recomienda que, aunque se vaya a realizar filtrado digital, aprovechar las características de los filtros analógicos que se utilicen.
5. Para realizar filtrado digital se recomienda utilizar un microcontrolador para procesamiento de señales, por la capacidad y recursos de operación que tienen.

## **V. MÓDULO 2: PROCESAMIENTO Y CLASIFICACIÓN DE SEÑALES PARA INTERFAZ BRAZO COMPUTADORA**

### **A. Marco teórico**

1. **Electromiografía.** El cuerpo humano está compuesto por diversos sistemas incluyendo el sistema nervioso, el cual se encuentra directamente relacionado al aparato locomotor. Este sistema ayuda a percibir el entorno por medio de las terminaciones nerviosas; detectar temperaturas, texturas e incluso enviar instrucciones a los músculos por medio de transmisión de datos en ambas direcciones a costa de potenciales de acción producidos por flujos iónicos celulares. Estos cambios de potencial generan un campo eléctrico en el rango de los microvoltios ( $\mu\text{V}$ ) que se puede medir por medio de electrodos (Escobar, 2006).

La red neuronal, compuesta por encéfalo, médula espinal, nervios y sistema vegetativo, es la responsable de coordinar las funciones por medio de potenciales. Al momento de realizar la transmisión de una señal por medio de las fibras nerviosas (potenciales de acción), se puede observar un cambio de voltaje en dichas fibras. Dependiendo del tipo de electrodos utilizados para la detección de estas actividades eléctricas se pueden clasificar como invasivas o no invasivas (Escobar, 2006).

Los electrodos invasivos (intramusculares) requieren de asistencia médica para su inserción, adicional de un cuidado extremo y mayores costos de operación. Esta forma de medición permite observar los potenciales en el rango de los milivoltios ( $\text{mV}$ ), mientras que los electrodos no invasivos (superficiales) muestran estos cambios en una escala menor con las ventajas de precios más alcanzables y menor complejidad al momento de colocarlos en el paciente (Konrad, 2005).

Como es de esperarse, las señales atraviesan el cuerpo y se pueden medir desde distintas áreas. Es por eso que al momento de colocar un electrodo en el bíceps es posible detectar el movimiento de un dedo o el de todo el brazo. Tomando esto en consideración, es necesario que se ubiquen los electrodos en posiciones que favorezcan la lectura de las señales que se quieran medir. Adicionalmente se debe contemplar el ruido que causan otras señales, ya sean internas o externas al paciente (Luca., 2006).

En la mayoría de los casos se puede ajustar el voltaje de referencia para los canales que convierten señales analógicas a digitales. Sin embargo, no es común que se trabaje en el rango de los  $\text{mV}$  por la facilidad que tiene la señal de corromperse por el ruido y otras perturbaciones que pueda sufrir. Por esta razón es necesario que se amplifique la señal para que sea adquirida con una resolución adecuada por medio de un canal ADC (Konrad, 2005).

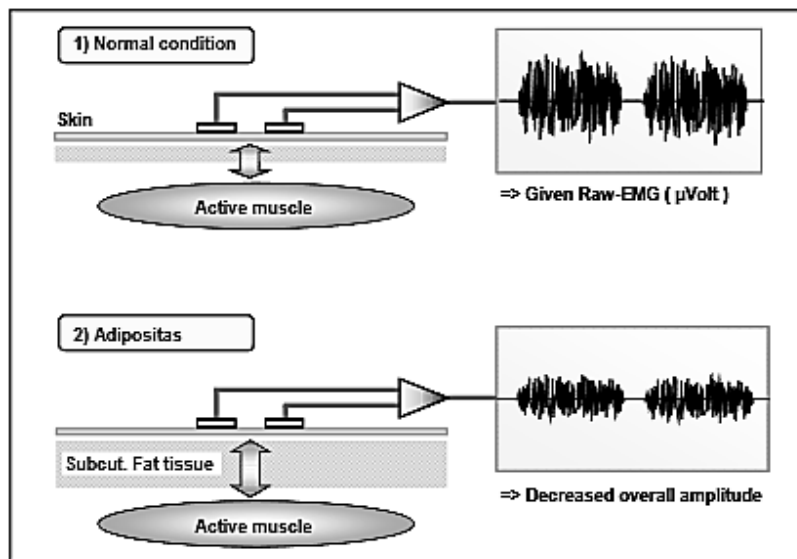
Al momento de amplificar una señal, se amplifica todo lo que se encuentre en el canal de entrada incluyendo el ruido que pueda contener. Es por esto que se debe implementar distintos tipos de filtros para ajustar la señal antes de ser convertida a digital. Entre estos cabe mencionar un filtro pasa baja que sirve como anti-alias, eliminando frecuencias superiores a la mitad de la frecuencia de muestreo. También es necesario un filtro pasa alta para eliminar la componente de voltaje DC de la señal adicional de eliminar el ruido común por medio de un amplificador de instrumentación (Konrad, 2005).

Habiendo fijado algunos principios a tomar en cuenta sobre la forma de tomar las mediciones, se puede establecer algunos puntos a considerar al momento de la toma de datos.

- Características del tejido.
- Intervención de señales.
- Cambio de geometría entre piel y electrodo.
- Ruido externo.
- Amplificadores.

En general, el cuerpo humano es un buen conductor, pero esto varía de tejido a tejido. Entre algunos factores se puede mencionar el grosor, cambios fisiológicos, temperatura y humedad. Para poder favorecer estas condiciones se puede untar solución salina, gel conductor o alguna otra sustancia conductora. También se puede rasurar el área de contacto con el electrodo para disminuir los ruidos (Parker, 2004).

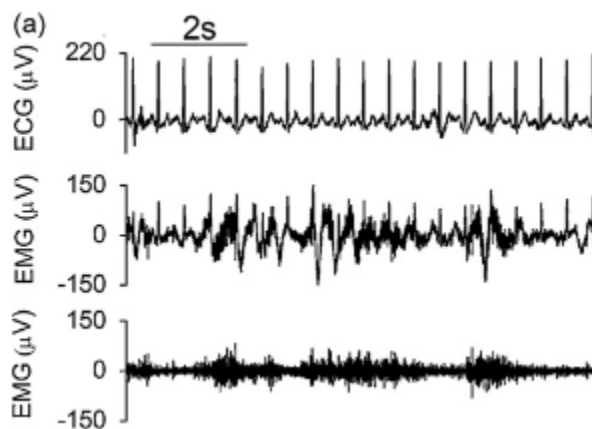
Figura 79. Reducción de señal por tejido adiposo



(Konrad, P.)

En la intervención de señales se cuenta con impulsos electromiográficos de músculos vecinos que sean significativos en las lecturas. Comúnmente esta intervención se encuentra entre el 10% y 15% de la señal deseada, sin embargo hay otros ruidos como el electrocardiograma que puede afectar gravemente dicha lectura (Hermes J., 2006). En estos casos se debe eliminar estas señales de ECG (electrocardiograma) para poder obtener un EMG (electromiograma) limpio. (Guohua L., 2009)

Figura 80. Señales ECG, EMG con y sin ruido ECG



(Guohua L.)

En la imagen anterior se observa tres señales distintas. En la primera se encuentra una lectura de electrocardiograma, la segunda es una señal de electromiograma con ruido de ECG y la tercera es la misma señal filtrada. El ruido por parte del electrocardiograma juega un gran papel en la lectura. De no ser filtrada, puede afectar las mediciones y limitar los resultados a no ser concluyentes. (Guohua L., 2009)

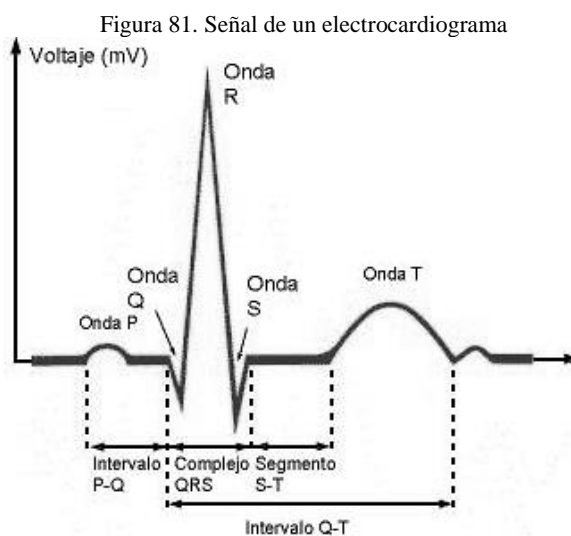
Luego se tiene el cambio de la geometría entre la piel y el electrodo. Al momento de contraer los músculos se cambia la geometría de la superficie cutánea, lo que afecta al contacto con el electrodo, variando el área de contacto o incluso separando por completo las superficies. Este falso contacto es perjudicial para el muestreo de la señal, por lo que se debe procurar que el electrodo se coloque en una posición que brinde la información requerida sin estar en un área que presente dificultad de muestreo por el cambio de geometría. De igual manera, aunque se busque que el electrodo no se mueva de su posición, también es recomendable utilizar adherentes sobre este para mantenerlo fijo. (Konrad, 2005)

En cuanto a la lectura de estas señales es necesario que se establezca las frecuencias de importancia, así como la frecuencia máxima, la frecuencia de muestreo del dispositivo y el rango de frecuencias de mayor interés. En el caso de la electromiografía se cuenta con un rango de señales entre los 10 y los 500 Hz. Sin embargo, el espectro más representativo se encuentra entre los 20 y 150 Hz. (Luca., 2006) Acorde con el criterio de Nyquist se debe tomar una frecuencia de muestreo de por lo menos el doble de la frecuencia

máxima. Tomando esto en consideración se debe de utilizar una frecuencia de por lo menos 1000 Hz en el dispositivo de muestreo.

2. **Electrocardiograma.** El corazón es el órgano muscular principal del aparato circulatorio. El hecho que sea un músculo que se contrae implica que cuenta con el mismo flujo iónico que en la electromiografía. Gracias a la lectura de estos impulsos se pueden detectar varias enfermedades del corazón, anomalías con el ritmo cardiaco (arritmia), bloqueos cardiacos entre muchas otras. Para este estudio se suele utilizar un total de diez electrodos colocados en distintas áreas del paciente (Yanowitz, 2006).

El constante estudio de esta señal ha proporcionado un estándar que se presenta en un ritmo cardíaco saludable. Los distintos picos de cada latido se seccionan en una onda P, un complejo QRS, una onda T y una última fase U (Yanowitz, 2006).



(Yanowitz, G.)

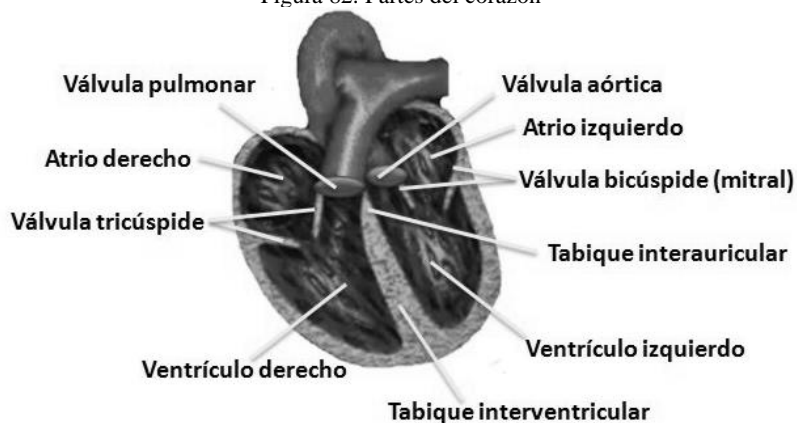
La onda P surge de la superposición de las despolarizaciones de las aurículas derecha e izquierda. Esta onda proporciona información sobre agrandamientos en los auriculares del corazón. Su duración se encuentra entre los 70 y los 110 ms dependiendo de la edad del paciente. Es esta onda es la que inicia el proceso de contracción y bombeo de la sangre (Yanowitz, 2006).

Luego se tiene el complejo QRS que se origina por la señal que causa la contracción de los ventrículos. El segmento PQ es en el cual la señal eléctrica viaja del nodo sinoauricular al nodo auriculoventricular. La onda negativa Q corresponde a la despolarización del tabique interventricular. La onda R corresponde a la despolarización de la masa principal de los ventrículos. Como se observa en la ilustración anterior hay una relación directa en lo masivos que son los ventrículos con la amplitud de la onda R. La onda S corresponde

a la última etapa de despolarización ventricular en la base del corazón. La repolarización auricular también ocurre en este momento, pero la señal se oculta por el tamaño del complejo QRS (Yanowitz, 2006).

La etapa ST refleja la etapa de equilibrio de los potenciales de acción miocárdicos, es en este segmento donde los ventrículos se contraen y bombean la sangre hacia el resto de sistema circulatorio. La onda T refleja la repolarización ventricular para luego llegar a la relajación ventricular. En esta última etapa es también donde entra la sangre a las aurículas derecha e izquierda para volver a repetir el proceso (Yanowitz, 2006).

Figura 82. Partes del corazón

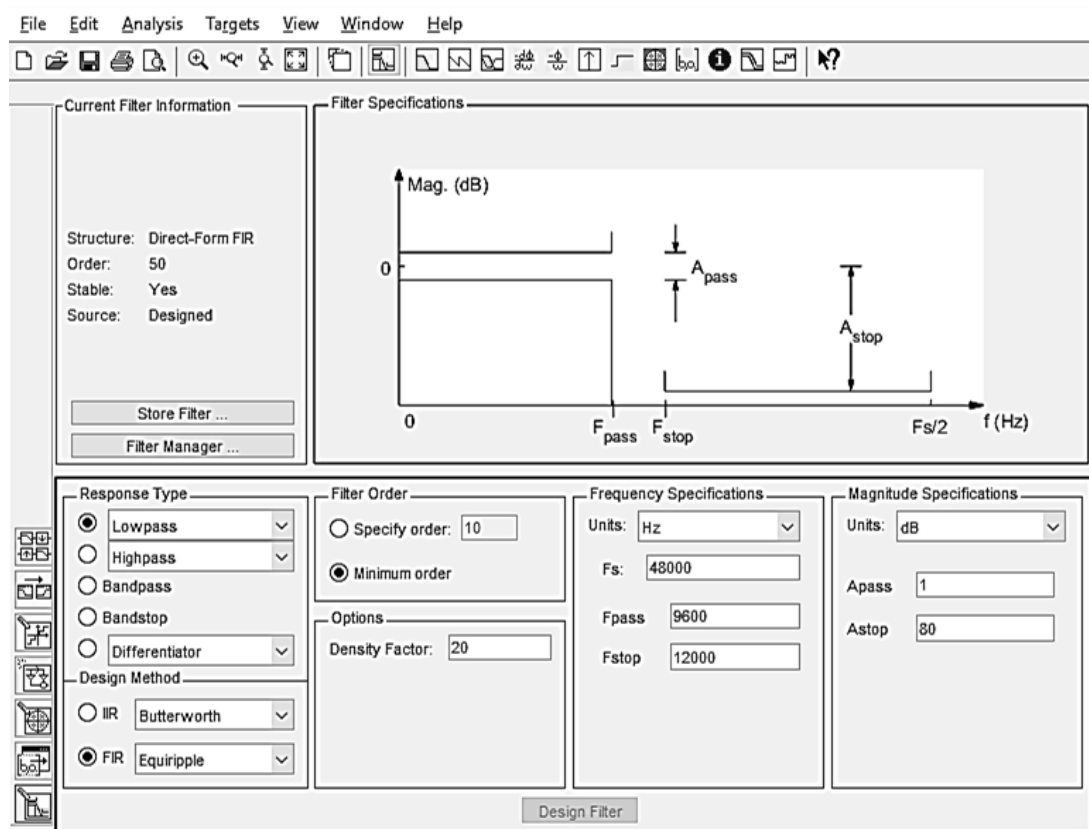


(Yanowitz, G.)

Un ritmo cardíaco saludable se encuentra entre 60 y 100 pulsaciones por minuto (ppm) para un adulto saludable en reposo. Un número alto de ppm corresponde a una dificultad mayor de bombear la sangre por todo el cuerpo. Tomando en cuenta que el ritmo cardíaco puede aumentar hasta las 230 ppm en actividad física intensa y un mínimo de 60 ppm en reposo, se puede concluir que la frecuencia cardiaca de la señal portadora se encuentra alrededor de 1 Hz y 4 Hz. Para la toma de estos datos por medio de electrodos, se debe de tomar los mismos cuidados anteriormente mencionados (Guohua L., 2009).

3. **Herramienta de análisis y diseño de filtros FDATool.** Para el diseño de filtros digitales existen diversas herramientas que permiten realizar análisis en el dominio del tiempo al igual que en el dominio de las frecuencias. En el software Matlab se encuentra una de estas herramientas que permite realizar el diseño de varios tipos de filtro con las características necesarias.

Figura 83. Herramienta de análisis y diseño de filtros



En esta herramienta permite realizar filtros analógicos con las siguientes características: Tipo de respuesta (pasa baja, pasa alta, pasa banda, elimina banda, multibanda, notch, entre otros), método de diseño para respuesta finita / infinita al impulso unitario, orden del filtro, frecuencias de corte, amplificación y atenuación.

Luego de crear el filtro, permite observar la respuesta que éste produce. Se observa la magnitud y la fase en respuesta a distintas frecuencias, el retardo que sufre por muestras a distintas frecuencias, la respuesta unitaria, al escalón, su diagrama de polos y ceros. Luego de diseñar el filtro adecuado, esta herramienta permite exportar el filtro como un objeto o como coeficientes. En el caso de filtrar señales por medio de Matlab es conveniente utilizar la opción de exportar como objeto a este filtro, ya que se puede utilizar como parámetro para el comando:

$$y = \text{filter}(\text{filtro\_diseñado}, \text{senal});$$

Si el análisis no se va a realizar por medio de Matlab, sino por medio de un dispositivo digital, se puede exportar la función de transferencia del filtro diseñado como coeficientes en el dominio z. Con base en esto se realiza el análisis matemático para obtener el filtro digital deseado para obtener una salida con base en

entradas y salidas anteriores. Donde la “entrada” es la señal obtenida por el dispositivo ADC y las salidas anteriores son los cálculos pasados por este filtro. En caso sean necesarias salidas pasadas, es necesario establecer condiciones iniciales.

Al final del diseño, se puede implementar el filtro como una operación matemática que depende de valores anteriores, asumiendo que no necesita de valores futuros de entrada o salida, de lo contrario se debe de realizar el procedimiento nuevamente para obtener un filtro causal (McClellan, 1998).

4. **Herramienta para aprendizaje de clasificación Classification Learner.** Las máquinas de aprendizaje cuentan con varias teorías y herramientas para poder realizar diversas tareas. En Matlab se cuenta con una herramienta de estas, la cual cuenta con varios algoritmos de clasificación que se entrenan por medio de una matriz que porta toda la información necesaria.

En esta herramienta se debe importar la matriz de datos, seleccionando el método de validación, escogiendo entre las opciones “Cross Validation”, “Holdout” y la última opción es sin validación alguna. Como los nombres lo indican, en la primera se hace una validación cruzada entre secciones y la segunda deja una parte de las muestras para la validación (Reitermanová, 2010).

Para la primera forma de validación es necesario ingresar una cantidad que define la cantidad de intervalos a validar entre ellos. La segunda recibe un parámetro que indica el porcentaje de muestras para realizar la validación, el resto se utiliza para entrenar el algoritmo seleccionado (Mathworks, 2016).

Luego de importar la información y seleccionar el método de validación, se debe escoger entre 20 distintos tipos de clasificadores, entre los cuales se encuentran árboles complejos, clasificadores por vecinos cercanos y máquinas de vectores de soporte, entre otros.

Las máquinas de vectores de soporte es una forma de clasificar la información por medio de hiperplanos. En primer lugar, se ubica la información en un espacio con las dimensiones necesarias para el problema. Usualmente se utilizan dos tipos de objetos a clasificar en un espacio bidimensional. En este plano cartesiano se plasman las figuras con un identificador, el cual puede ser el color y la geometría, entre otros. Luego por medio de optimización ubica el hiperplano divisor que logra clasificar a los componentes, manteniendo una distancia pareja hacia ambas partes. (Gunn, 1998)

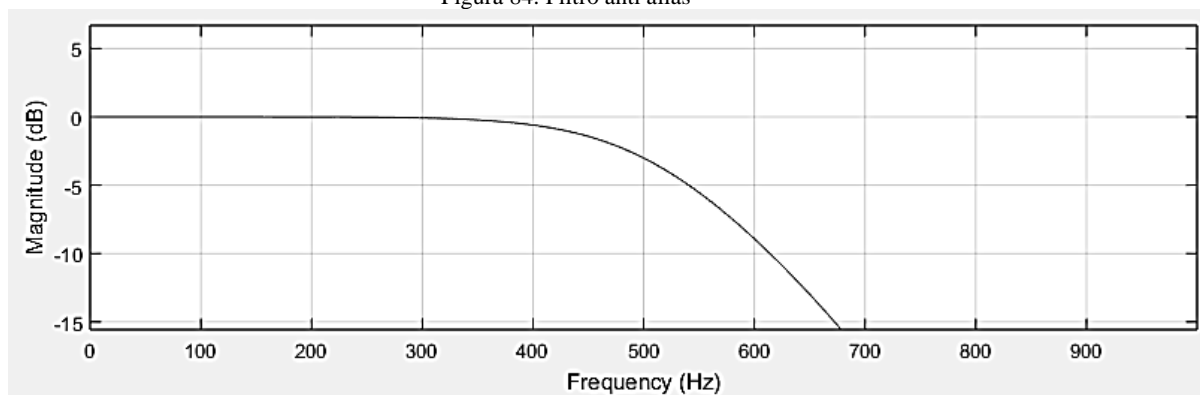
## B. Marco metodológico

1. **Obtención de señales EMG y EKG.** El propósito general del trabajo es la clasificación de las señales electromiográficas y electrocardiográficas, por consiguiente, la obtención de datos y los filtros se trataron como cajas negras con comportamientos ideales.

Ya se mencionó anteriormente que las señales tienen ciertas frecuencias deseadas y que la amplitud de estas no es favorable para realizar un muestreo directo de la señal sin realizar algún tipo de ajuste. Tomando esto en cuenta se diseñaron cajas negras que puedan condicionar la señal a voltajes legibles (en términos de conversión ADC) y en frecuencias de interés.

La ventana de la señal se extiende hasta los 500 Hz, por lo que se aplicó un filtro que limitara las frecuencias hasta esta frecuencia máxima. Tomando esto en cuenta, se eliminó todas las señales con frecuencias mayores a la indicada, ya que de no hacerlo, estos componentes de alta frecuencia mayor a la máxima deseada van a ser mapeados a bajas frecuencias dentro del sistema (Johnson, 2011).

Figura 84. Filtro anti alias



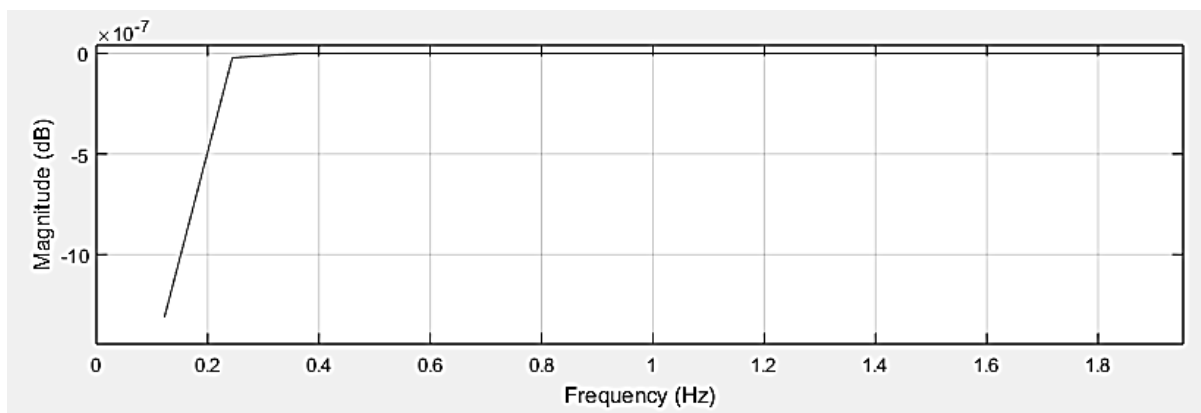
Luego de limitar la ventana de observación en el dominio de las frecuencias con una frecuencia de corte de 500 Hz, se eliminó la componente DC del sistema (que corresponde a 0Hz), esto con el fin de poder manipular la señal de una forma más eficiente. Esta componente de voltaje DC es una señal que afecta la medición, ya que no es un valor fijo ni uno conocido, por lo tanto, es necesario eliminar dicha característica.

De la misma manera que se empleó un filtro pasa baja para eliminar las frecuencias mayores al espectro deseado, se utilizó un filtro pasa alta para eliminar la componente DC. Como se puede apreciar en la Figura 6, los filtros tienen una banda de frecuencia para la cual la señal se atenúa. Esto quiere decir que no elimina completamente las frecuencias a partir de la señal de corte, debido a la llamada banda de transición del filtro.

Tomando esto en consideración hay que recordar que la frecuencia cardíaca se encuentra desde los 60 latidos por minuto, hasta un agitado 240 latidos por minuto. Empleando un factor de seguridad, se sobredimensionó este rango, utilizando 30 latidos por minuto como la mínima y 300 latidos por minutos para la máxima. Trasladando estos datos a términos de frecuencia, se cuenta con un rango entre 0.5 y 5 Hz.

Por consiguiente, se tuvo que utilizar un filtro pasa alta, que permitió eliminar la señal DC sin afectar la lectura electrocardiográfica. En la Figura 85 se observa como el filtro fue capaz de atenuar las frecuencias cercanas a 0 Hz, sin tener algún efecto sobre las frecuencias involucradas en el electrocardiograma.

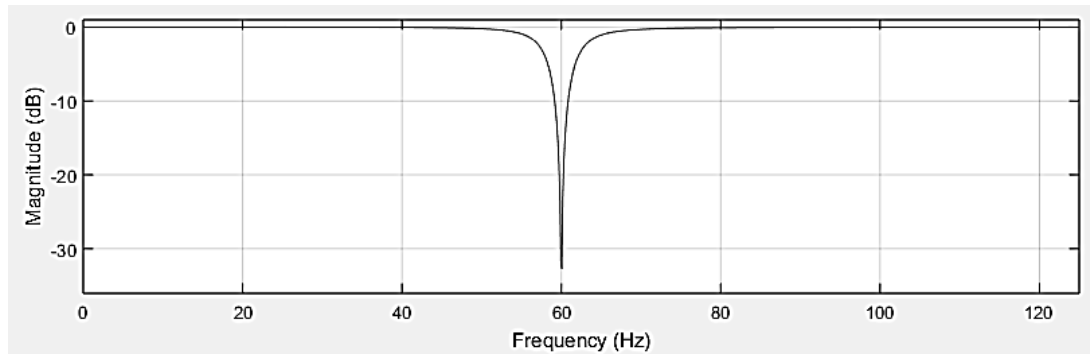
Figura 85. Filtro pasa alta  
Filtrado de componente DC



Adicionalmente a este proceso se debe realizar un filtro para eliminar el ruido común que se encuentra alrededor de 60Hz, el cual se origina por la alimentación AC del sistema, siempre tomando en cuenta esta misma atenuación de las señales en los distintos rangos. Este proceso se puede implementar tanto en un circuito analógico como mediante un filtro digital. Para efectos de las cajas negras se colocó en esta sección, sin embargo, el filtrado de esta señal de forma digital se encuentra en la sección de procesamiento de señales EMG.

En cualquiera de los dos enfoques (filtrado analógico o digital) lo que se desea realizar puede efectuarse con un filtro del tipo notch, cuyo funcionamiento es el de atenuar la señal a una frecuencia específica, tratando al mismo tiempo de alterar las frecuencias vecinas lo menos posible. El comportamiento de este filtro en el dominio de frecuencial se puede observar en la Figura 8. Como se puede observar, este filtro tuvo una única muesca en los 60 Hz.

Figura 86. Filtro tipo "notch" a 60 Hz



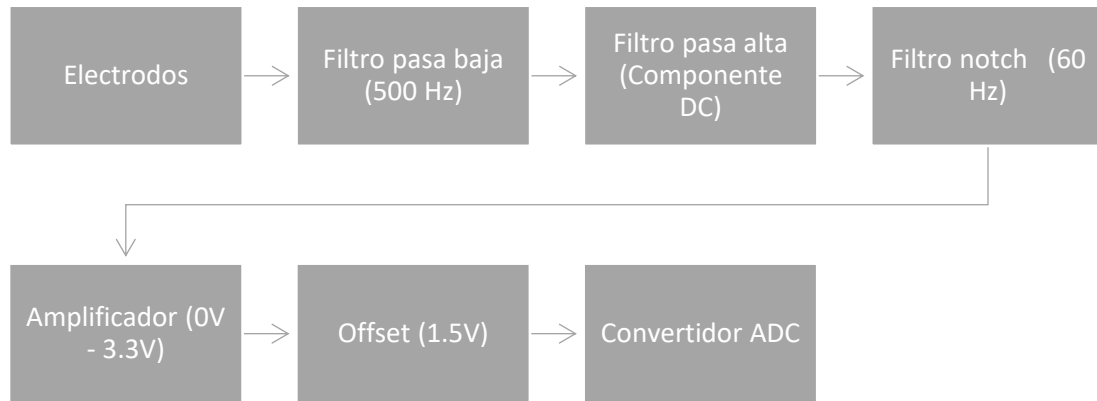
Luego de aplicar el filtro se obtuvo una señal que se encontraba en el rango de frecuencias necesarias, sin embargo, no con una amplitud que permitió muestrear por medio de canales analógicos. Por lo tanto, se tuvo que amplificar la señal con un factor suficiente para obtener un diferencial de voltaje de 5V pico – pico a partir de las señales en el rango de mV.

Los convertidores analógicos a digitales utilizan un método de reconstrucción por medio de sujetador de orden cero (Baraniuk, 2003), el cual mantiene el valor medido durante un periodo completo de muestreo para poder realizar la conversión. El convertidor analógico-digital que se utilizó no cuenta con valores negativos y su voltaje de referencia es de 3.3V (se puede ajustar a un voltaje de entrada u otro voltaje interno, sin embargo 3.3V es lo que se va a considerar para el muestreo), por lo que tenemos que ajustar la señal a este rango.

Para esto se realizó un desfase con un voltaje DC, sumándose a la señal recién procesada. De forma arbitraria se tomó el valor de 1.5 V para tener un rango de 1.5 V en ambas direcciones de resolución. De esta forma se centró la señal en este voltaje permitiendo monitorear su comportamiento en el rango de valores positivos y negativos. Teniendo una resolución de 16 bits para cada muestra, se puede llegar al diferencial de voltaje que se obtiene en cada muestreo.

El sistema de cajas negras que filtran y amplifican esta señal se puede observar en la Figura 87, donde cada etapa contribuyó a la obtención de dichos datos para luego poder transmitirlos a etapas posteriores en los módulos de procesamiento por computadora.

Figura 87. Proceso de filtrado de señal para muestreo analógico-digital



Sabiendo que el voltaje varía de  $0V - 3.3V$  y que se cuenta con 16 bits de información booleana para indicar el voltaje de la señal, se concluyó que se dispone de  $2^{16}$  diferentes combinaciones para describir el potencial muestreado. Como se observa en la Ecuación 1, esto significa que el valor más pequeño posible de representar con un bit se encuentra en el rango de los microvoltios ( $\mu V$ ) para una referencia de  $3.3V$ .

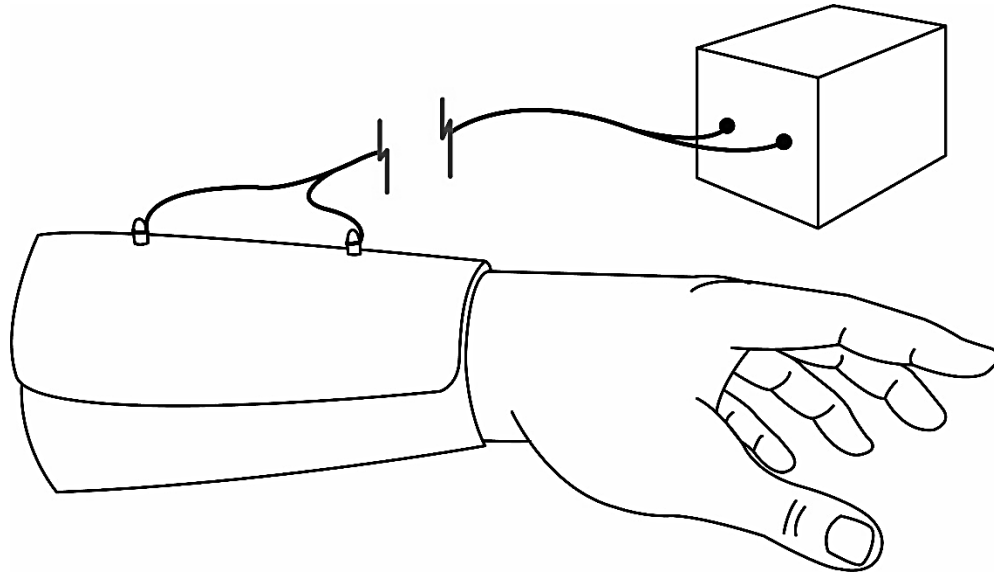
$$3.3V = 2^{16} \text{ bits}$$

$$1\text{bit} = \frac{3.3V}{2^{16}} \cong 50.35\mu V \quad (1)$$

En la obtención de señales se debe tomar en cuenta la colocación de estos electrodos para poder obtener una buena recepción de esta. En la siguiente imagen se muestra el diseño conceptual de la pulsera, el cual sujeta los electrodos en posición para poder realizar los gestos y mover la mano sin afectar el posicionamiento de estos.

Este diseño ilustra claramente el proceso de recepción de la señal para luego ser procesado por las cajas negras anteriormente mencionadas, utilizando el diagrama de flujo de la Figura 9.

Figura 88. Diseño conceptual de pulsera



Se utilizó este diseño para poder permitir el libre posicionamiento de los electrodos dentro de la pulsera, de tal manera que se pueda encontrar la posición óptima para la obtención de los datos utilizando las recomendaciones del SENIAM. Como se puede observar en la Figura anterior, el diseño se conformó por una tela envolvente que permite a los electrodos registrar la actividad cutánea y atravesar esta tela para trasladar por medio de cables a los módulos de filtrado y análisis.

2. **Procesamiento de señales EMG.** Para el procesamiento de la señal EMG, se aplicó como primera etapa el filtro para eliminar el ruido común, ubicado en 60 Hz (Lindström LH, 1977). Utilizando la herramienta de diseño de filtros, se encontró la función de transferencia para obtener un comportamiento como el que se muestra en la Figura 86.

Esta función de transferencia se despejó para obtener la ecuación de diferencias en el dominio discreto de tiempo. Esta ecuación de diferencias fue la que se implementó por medio de código en el microprocesador utilizado para convertir la señal de analógica a digital. Para este filtro se tomó en cuenta la frecuencia de muestreo, la frecuencia para la muesca (notch), el ancho de banda y la magnitud de las señales no filtradas.

Sabiendo que la función de transferencia en el dominio  $z$  equivale a la salida del sistema  $y[n]$  usar herramienta de ecuación dentro del texto en el dominio  $z$ , sobre la entrada  $(x)$  en el dominio  $z$ , se despejó para llegar a la ecuación de diferencias en la Ecuación 3.

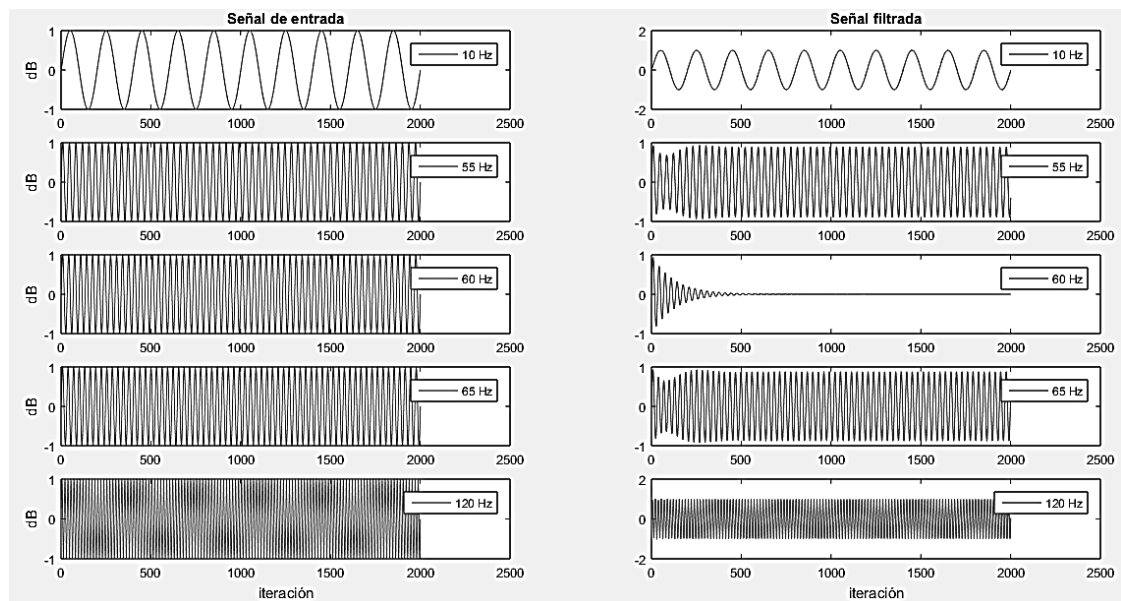
$$H(z) = \frac{0.9921 - 1.9490z^{-1} + 0.9921z^{-2}}{1 - 1.9490z^{-1} + 0.9841z^{-2}} \quad (2)$$

$$y[n] - 1.9490y[n-1] + 0.9841y[n-2] = 0.9921x[n] - 1.9490x[n-1] + 0.9921x[n-2]$$

$$y[n] = 1.9490y[n-1] - 0.9841y[n-2] + 0.9921x[n] - 1.9490x[n-1] + 0.9921x[n-2] \quad (3)$$

Implementando este filtro en Matlab, se comprobó el comportamiento ingresando distintas señales de entrada y graficando su salida con base en estas entradas. Para verificar que el filtro realiza la función deseada se colocaron cinco distintas funciones de entrada con distintas frecuencias. Como se puede apreciar en la Figura 89, hay una frecuencia de 10 Hz, una de 55 Hz, 60 Hz, 65 Hz y una última de 120 Hz.

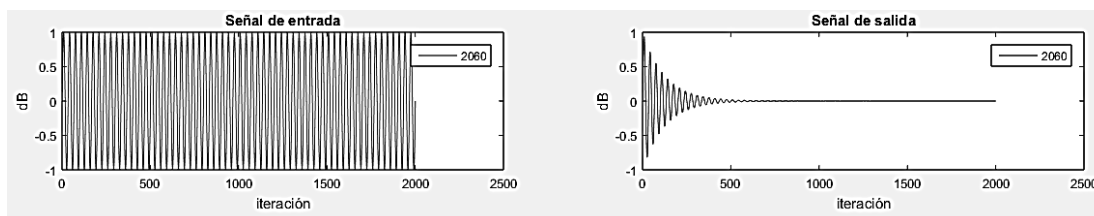
Figura 89. filtro notch, comportamiento de señales a distintas frecuencias



En la Figura se observa claramente que la señal de 60 Hz se atenuó hasta llegar a una amplitud de 0 dB. La señal de 10 Hz no se vio afectada, al igual que la señal de 120 Hz. Por último, están las señales de 55 y 65 Hz las cuales se deformaron en las primeras iteraciones, sin embargo, el sistema llegó a estabilizarse y se contó con una amplitud levemente alterada en el resto de las iteraciones. Esta leve deformación se produjo porque la banda de paso que se seleccionó fue de 10 Hz.

Teniendo una frecuencia máxima de 500 Hz para las señales electromiográficas, se tomó una frecuencia de muestreo de 2000 Hz. Para demostrar la importancia del filtro anti-alias en el proceso de la obtención de la señal, se ingresó una señal sinusoidal con frecuencia de 2060 Hz para graficar su comportamiento ante el filtro de muesca.

Figura 90. Filtro “notch” en frecuencias mayores a la frecuencia de muestreo

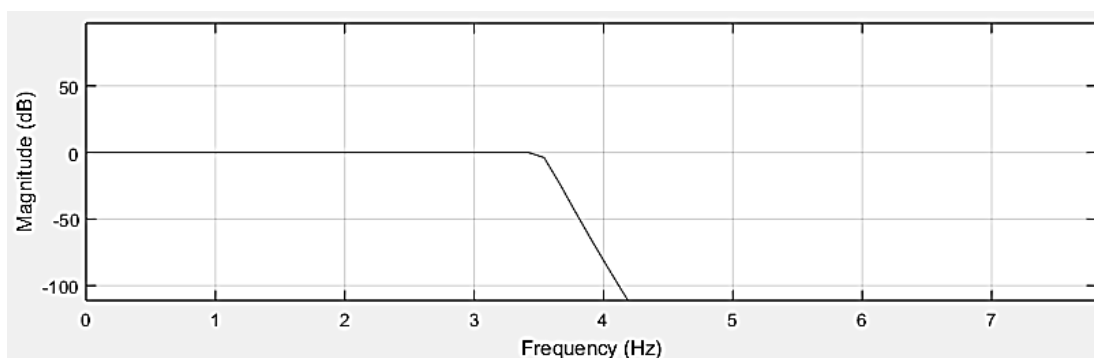


Como se puede observar en la Figura 90, la señal de 2060 Hz se comportó de la misma manera que lo hace la señal de 60 Hz, por lo que al tener una frecuencia de muestreo de 2 kHz se cuenta con ruido si no es filtrado para conseguir la ventana de interés.

3. **Procesamiento de señales EKG.** Para el análisis de las señales del ritmo cardíaco, se hizo un enfoque en una ventana de 0.5 a 4 Hz. Para esto se tomó la misma señal de entrada que para EMG y se realizó un filtro pasa baja digital. Para esto se utilizó la herramienta de análisis y diseño de filtros, ingresando los valores de frecuencia de muestreo, frecuencia de corte y el tipo de filtro.

En la Figura 91 se muestra el comportamiento de la señal en el dominio de las frecuencias para el filtro pasa baja con frecuencia de corte de 4 Hz.

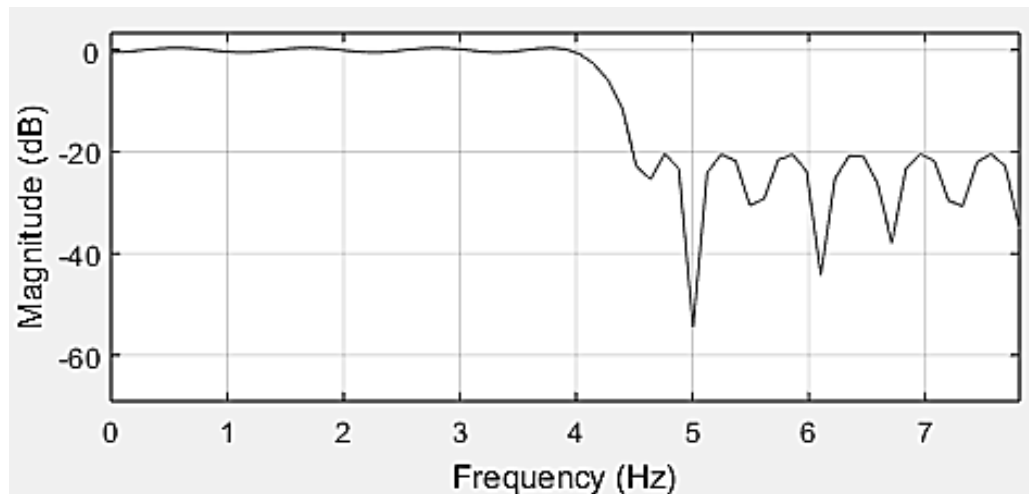
Figura 91. Filtro pasa baja con frecuencia de corte a 4Hz tipo IIR



Para el caso de las señales electrocardiográficas se evitó realizar un filtro de este tipo, ya que los resultados obtenidos por medio de filtros IIR presentan distorsiones de fase no deseadas. Desafortunadamente, los filtros

del tipo FIR resultaron ser ineficientes en este caso y el fallido intento de realizar el filtro deseado por medio de la herramienta de diseño de filtros se muestra en la Figura 92. En esta Figura se muestra uno de los varios comportamientos que toma la señal con estas especificaciones en el dominio de frecuencias al momento de tomar una de las posibles soluciones a la tarea. Este filtro se conformó por más de 300 coeficientes en el numerador, lo que lo hizo poco práctico para implementar.

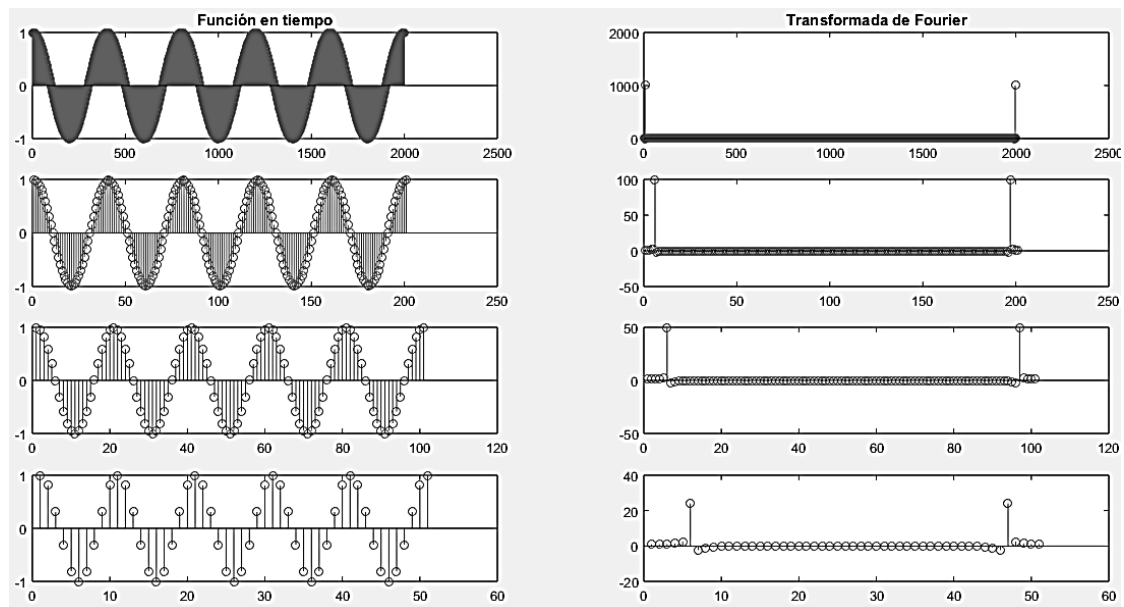
Figura 92. Filtro pasa baja con frecuencia de corte a 4Hz tipo FIR



Partiendo de esto, se requirió utilizar la transformada rápida de Fourier. Como se ha mencionado anteriormente, la frecuencia de muestreo para la obtención de datos es de 2 kHz, sin embargo estas señales son de muy bajas frecuencias. Por esta razón se realizó un diezmado (decimation) de la señal.

En la Figura 93, en la esquina superior izquierda se encuentra una señal cosenoidal de 50 Hz con una frecuencia de muestreo de 20 kHz. A la derecha está graficada su transformada de Fourier. Como se puede observar, es difícil de determinar visualmente las frecuencias que componen la señal al ver el diagrama de magnitud en frecuencia. Esto se debe a que se cuenta con varios elementos que no son de interés para este análisis. La segunda Figura de la columna izquierda contiene la misma señal, realizando un diezmado con un factor de 10. La señal se siguió comportando de la misma manera, sin embargo, la frecuencia de muestreo disminuyó, por lo que se nota menos relleno el gráfico de vástagos (stem plot). Su transformada de Fourier asociada mostró una escala que detalló de mejor manera las frecuencias de interés, obviando las que eran innecesarias en este proceso. Se repitió este procedimiento en las gráficas inferiores, dilatando la frecuencia de muestreo y mejorando la resolución en las frecuencias de utilidad.

Figura 93. Transformada rápida de Fourier



En las siguientes dos gráficas se cuenta con una dilatación de la frecuencia de muestreo de 20 y 40. En ambos diagramas se obtuvo mayor exactitud en la frecuencia fundamental que compone la señal, detallando de mejor manera el comportamiento en este rango. La frecuencia original de esta señal es de 20 kHz y en la última señal se utilizó un diezmado de 40, al dividir  $\frac{20,000\text{Hz}}{40}$  se obtiene 50Hz, lo cual se aprecia en la escala del gráfico inferior derecho.

Utilizando esta información se calculó la frecuencia original de la señal, obteniendo el valor frecuencial de la lectura. Se extrajo el valor máximo y la ubicación del vector que compone el resultado de la FFT. Se encontró que el valor que corresponde al máximo se encontró en la posición 6. Ingresando estos valores a la Ecuación 4, se calculó la frecuencia que compone la señal original.

$$f = \frac{F_s}{2 * \pi} * \omega \quad (4)$$

$$f = \frac{50}{2 * \pi} * 6 \approx 47.8 \text{ Hz} \approx 50 \text{ Hz} \quad (5)$$

Al sustituir los valores, se pudo determinar que el resultado no resultó ser exactamente la frecuencia ingresada originalmente al sistema (50 Hz). Al analizar la Figura detenidamente, se estipuló que hay leves fluctuaciones en los números cercanos a 6. Ya que este vector está conformado por números discretos y no todo el espectro, no se obtuvo una respuesta más precisa. Esta aproximación se tomó como suficientemente

buena para simplificar los cálculos del ritmo cardiaco, por lo que el diezmo que se realizó es suficientemente representativo para poder obtener una mejor definición a bajas frecuencias.

Por medio del ejercicio anterior se estableció la importancia del diezmo para poder realizar interpretaciones con mayor precisión. Al momento de diezmar por un factor, se disminuyó el tiempo de cómputo para realizar la transformada de Fourier, ya que los datos transformados fueron menos al momento de realizar este proceso.

En el Cuadro 1 se observan los distintos tiempos requeridos para obtener las transformadas de Fourier de la Figura 15. Al tener una menor cantidad de datos, el tiempo para transformar la información al dominio de las frecuencias es mucho menor.

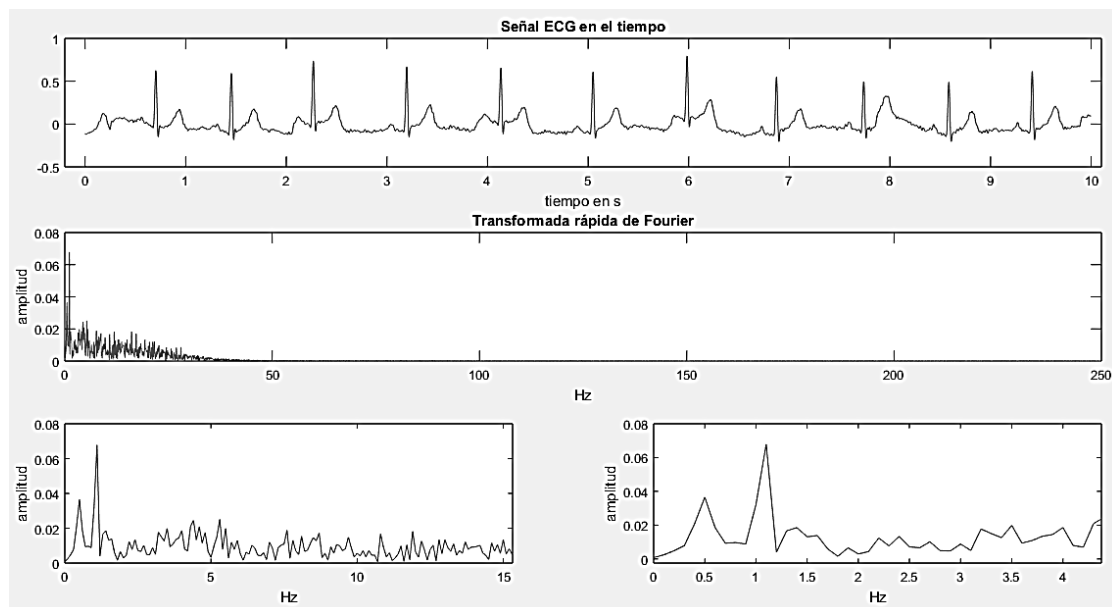
Cuadro 1. Tiempo requerido para transformaciones de Fourier

<b>Factor de diezmo</b>	<b>Cantidad de datos</b>	<b>Tiempo → FFT [ <math>\mu s</math> ]</b>
<b>Original</b>	2001	79
<b>10</b>	201	15
<b>20</b>	101	12
<b>40</b>	51	06

En el caso del análisis del ritmo cardiaco se utilizó información proporcionada por PhysioBank ATM (Goldberger, y otros, 2000), realizados a pacientes tanto de prueba como con enfermedades cardiacas pertinentes al análisis de estas señales. Se descargó una base de datos de un sujeto de prueba con el cual se realizaron los debidos experimentos.

Esta base de datos cuenta con una muestra de 10 segundos y una frecuencia de muestreo de 500 Hz. Esta data se puede observar en la Figura 16, al igual que su respectiva transformada de Fourier y acercamientos de la segunda. Al momento de transformar al dominio de Fourier, se identificó el valor máximo a una frecuencia de 1.1Hz.

Figura 94. Señal ECG junto con su transformada de Fourier y acercamientos en las frecuencias de interés



En la Figura se contabilizó un total de once picos del tipo R para el intervalo de 10 segundos, esto confirmó los resultados obtenidos por la transformada al dominio de las frecuencias, ya que multiplicando la lectura de 1.1Hz por el intervalo de tiempo resultó en un total de 11 períodos cardiacos. Análogamente, se amplió la predicción hacia un intervalo de 60 segundos para establecer las pulsaciones por minuto, consiguiendo en un total de 66 pulsaciones por minuto.

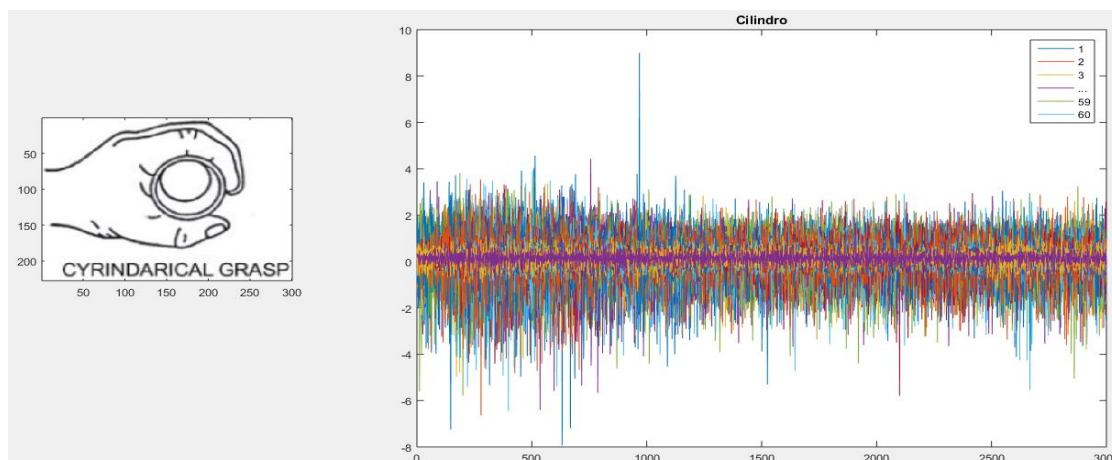
4. **Clasificación de señales EMG.** Para la clasificación de las señales EMG se utilizó información proporcionada por la Universidad Irvine de California que se tomó en el 2013 (Sapsanis, Georgoulas, Tzes, & Lymberopoulos, 2013). Esta información consta de una base de datos que contiene señales electromiográficas del brazo de cinco personas distintas (tres mujeres y dos hombres) en seis distintas posiciones. Para cada una de las personas se realizaron 60 muestras de cada gesto.

En las Figuras 95 a 102 se puede observar el comportamiento de las seis posiciones para la primera mujer muestreada. En cada una de estas se muestra el gesto realizado con la mano y la señal electromiográfica que detectan los electrodos. Cabe mencionar que las 60 mediciones distintas están graficadas en cada imagen para poder identificar el comportamiento general. De igual forma, cabe recalcar que cada una de estas gráficas muestra las señales obtenidas de una sola persona, esta información está disponible para los otros cuatro pacientes.

En la primera de estas gráficas se logra observar el comportamiento de las señales electromiográficas cuando la mano sostiene un cilindro de algún tipo. Se identifica que las señales no mostraron un

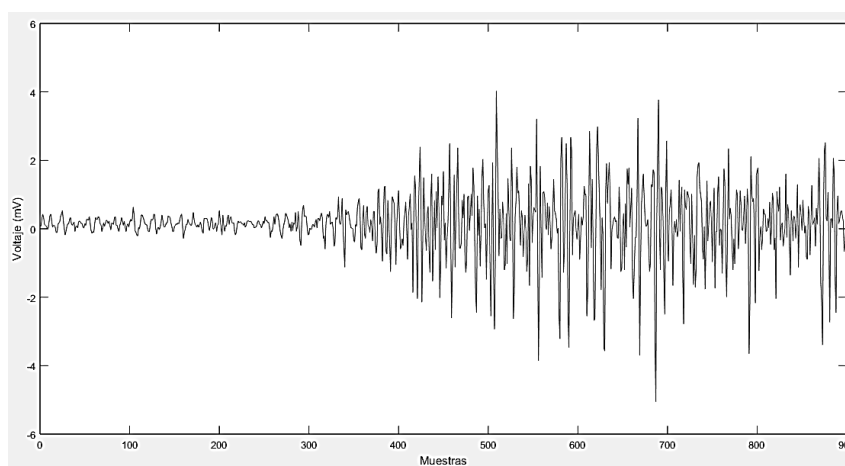
comportamiento homogéneo. Los colores de cada señal muestran que las amplitudes de cada muestreo varían sin un patrón reconocible.

Figura 95. Señal electromiográfica de mujer #1 sosteniendo un cilindro



En la Figura 96 se observa un acercamiento de una de las señales que conforman la Figura 16. En esta Figura se muestra únicamente el voltaje de la primera señal, durante las primeras 1000 muestras. De igual manera, no se logra reconocer algún tipo de patrón, sin embargo se puede apreciar un comportamiento fluctuante. También es notorio el hecho que dentro de la misma señal se tuvieron distintas amplitudes para un solo gesto. Esto quiere decir que no es una señal que varía sobre amplitudes determinadas.

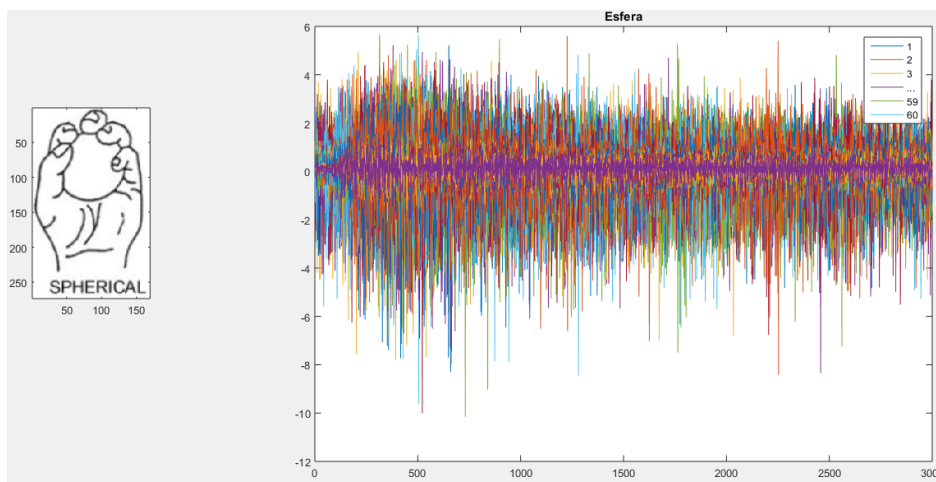
Figura 96. Detalle de señal electromiográfica de mujer #1 sosteniendo un cilindro



En la siguiente Figura se muestran 60 tomas de datos para la misma persona realizando un nuevo gesto. El gesto, como lo muestra la imagen, es el de sostener un objeto esférico. Los resultados fueron parecidos al primer muestreo, a pesar que los datos provenían de otro gesto. El hecho de graficar las 60 muestras al mismo

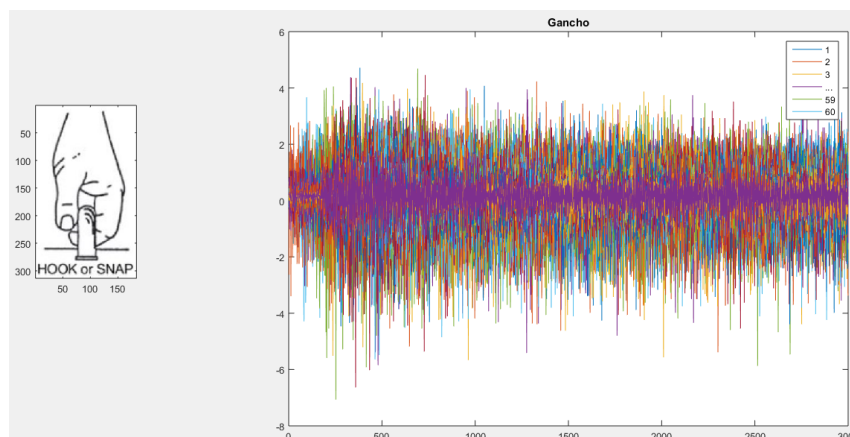
tiempo proporcionó un panorama general del comportamiento general de la señal, tratando de observar algún tipo de características que permita diferenciar una Figura de la otra.

Figura 97. Señal electromiográfica de mujer #1 sosteniendo una esfera



La siguiente imagen muestra otro gesto y sus respectivas señales para la misma persona. Se observó que los tres gestos mostrados hasta el momento eran un tanto similares, por lo que las gráficas no mostraron cambios significativos. Hasta el momento se encontró que el comportamiento general de la señal cambió según el gesto, sin embargo, no fue tan notorio por la similitud de los gestos.

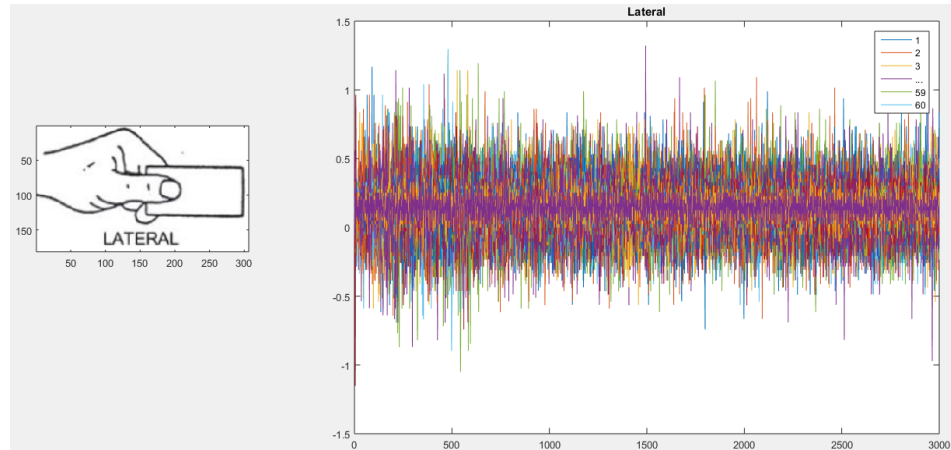
Figura 98. Señal electromiográfica de mujer #1 haciendo un gancho con la mano



En la Figura 99 se encuentra un gesto que se diferenció de los anteriores. En los gestos anteriores, se estaba sujetando algún objeto con toda superficie interior de la mano y se utilizaron la mayoría de los dedos, ahora el gesto constaba de la sujeción de un objeto plano por medio de los dedos índice y pulgar. Una de las principales características que ayuda a diferenciar este gesto de los anteriores es la amplitud de la señal.

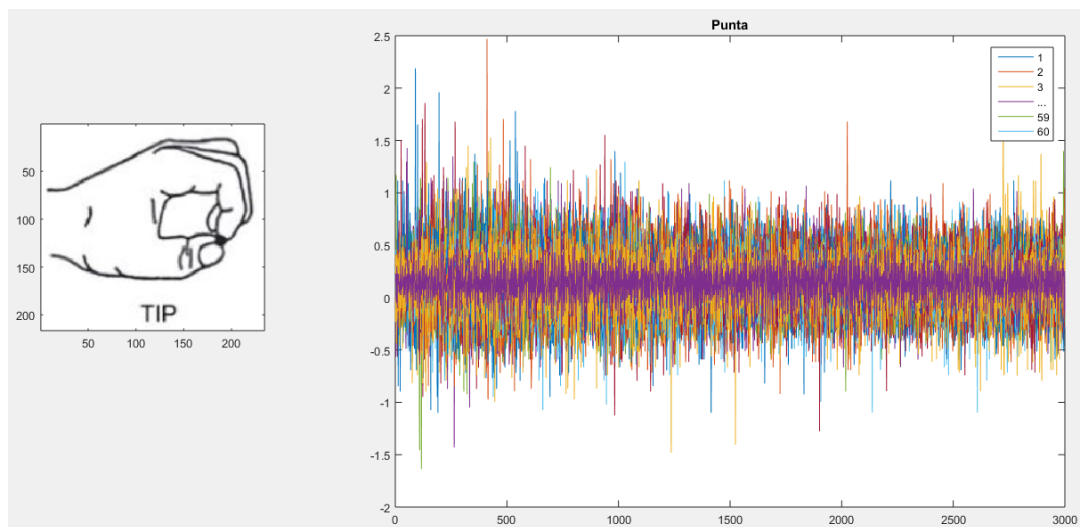
Como se puede apreciar, anteriormente se contó con unos picos promedio de 3 mV, mientras que la escala del nuevo gesto no rebasó los 2 mV.

Figura 99. Señal electromiográfica de mujer #1 sosteniendo un objeto plano de forma lateral



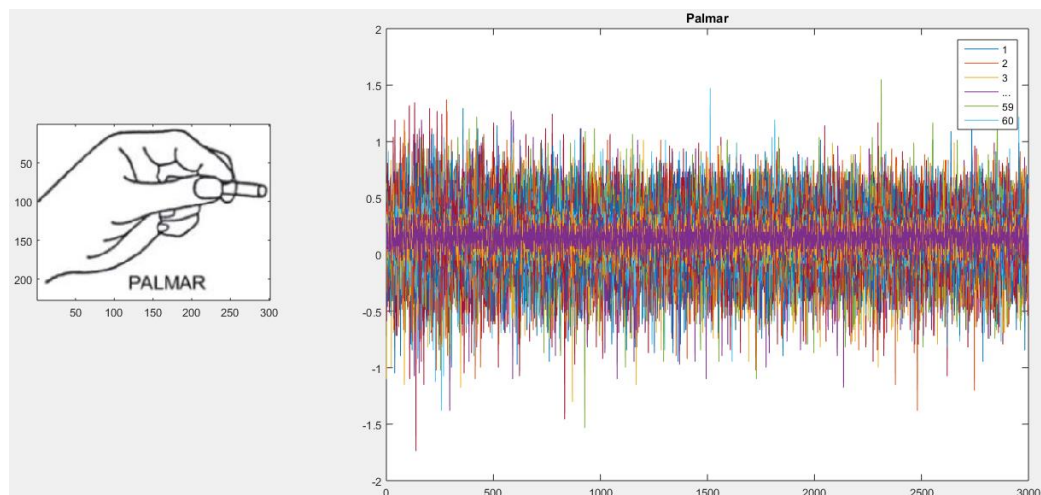
La siguiente Figura cuenta con una señal muy similar a la anterior, la amplitud también se mantuvo en el rango de 1 mV y el comportamiento fue muy parecido.

Figura 100. Señal electromiográfica de mujer #1 sosteniendo un objeto con la punta de los dedos índice y pulgar



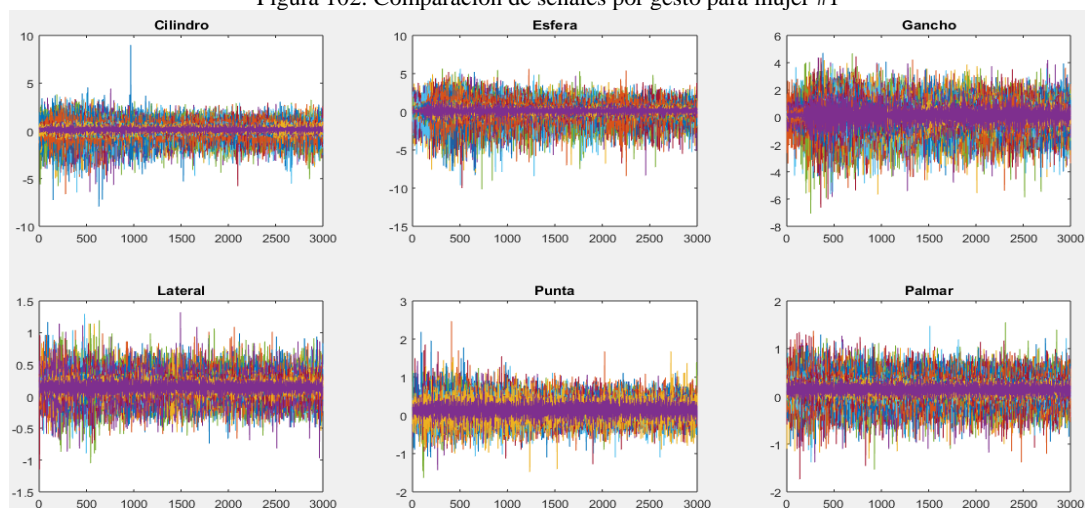
Por último, se tiene una sujeción palmar de un objeto. Este gesto es el que se realiza al momento de sujetar un bolígrafo al momento de escribir.

Figura 101. Señal electromiográfica de mujer #1 sosteniendo un bolígrafo



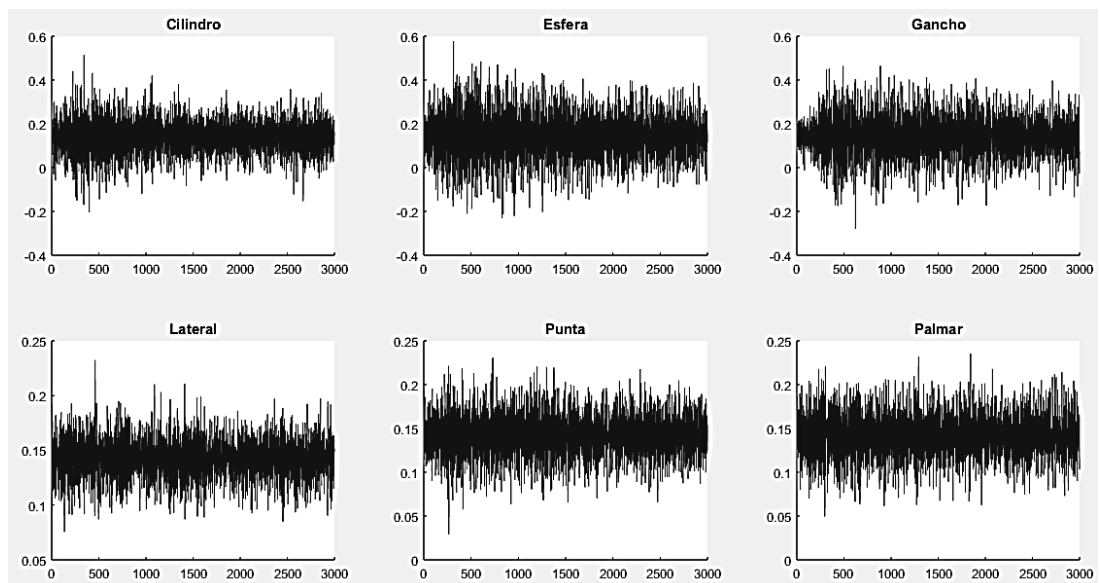
La Figura 102 muestra las seis gráficas anteriores consolidadas en una sola para poder observar las similitudes y diversidades entre las gráficas. Las primeras tres imágenes muestran una escala que supera los 5 mV, mientras que las tres inferiores tienen un máximo de 3 mV. Claramente las tres superiores cuentan con una amplitud más pronunciada. Las señales fluctuaron en todos los casos, como se pudo apreciar en el detalle que se hizo de la Figura 96, las señales de las gráficas superiores también pueden tomar rangos menores. Esto quiere decir que no fue posible regirse únicamente por la amplitud, ya que, al analizar un corto tramo de señal, no resultó contundente clasificar la señal a una de las seis categorías.

Figura 102. Comparación de señales por gesto para mujer #1



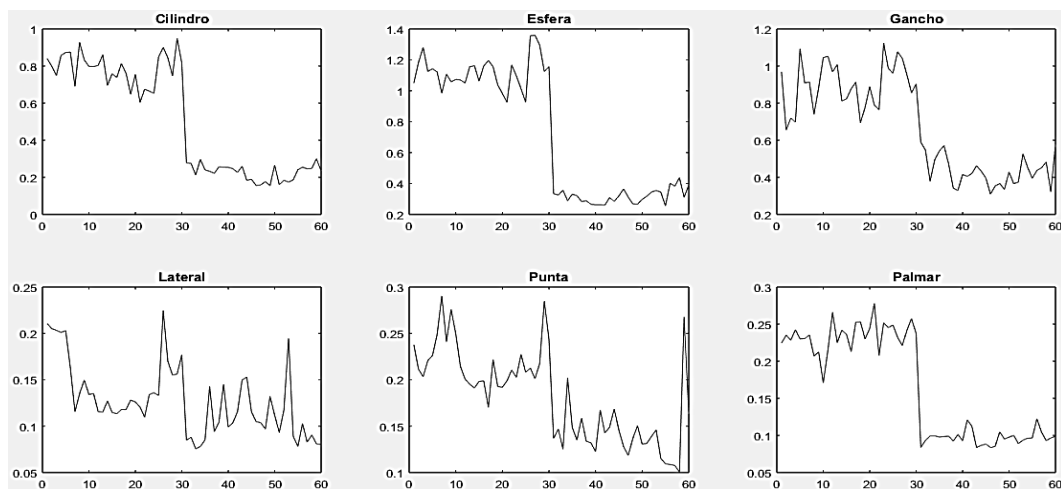
Tratando de encontrar algún patrón de comportamiento se graficó el promedio de las señales anteriores. Para esto se tomó los 60 datos de las señales y se trazó nuevamente. En este nuevo gráfico se pudo ver que la amplitud general de las gráficas se atenuó y se obtuvieron valores centrados alrededor de 1.5 mV.

Figura 103. Promedio de 60 señales por gesto de mujer #1



Adicionalmente se graficó la desviación estándar de cada una de las señales. Para esta Figura se contó con 60 datos distintos, los cuales muestran la desviación estándar para cada muestreo realizado. Como se puede apreciar, la desviación varió para cada muestra tomada.

Figura 104. Desviación estándar por muestra

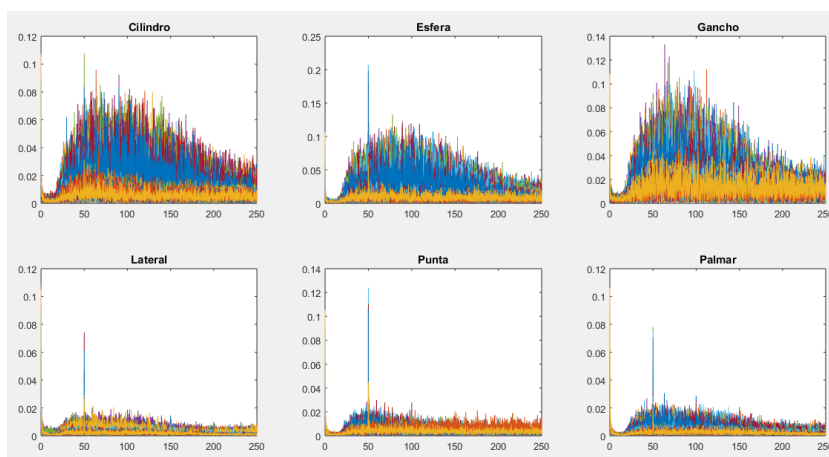


Habiendo realizado las pruebas anteriores se dedujo que la señal en el dominio del tiempo no proporciona información sencilla de analizar e interpretar. Por esta razón fue necesario utilizar más herramientas para que se pudiera contar con información que proporcionara algún tipo de clasificación de las señales. Lo que se

buscó fue obtener señales que fuesen replicables y reconocibles según el gesto al que pertenecen. Estas cualidades que se buscan en las señales se puede apreciar de mejor manera en la Figura 105. En esta Figura se encuentran las mismas 60 señales para cada gesto de la misma persona, en el dominio de las frecuencias. Esta imagen resultó de realizar la transformada rápida de Fourier, tomando en cuenta su frecuencia de muestreo de 500 Hz.

Es importante señalar el hecho que estas frecuencias no dependen del tiempo y pudieron ser alteradas por ruidos externos. Esto quiere decir que una muestra se puede ver comprometida al momento de tener un ruido significativo que altere las frecuencias que verdaderamente componen a la lectura de interés.

Figura 105. Transformada de Fourier de todas las señales correspondientes a la mujer #1

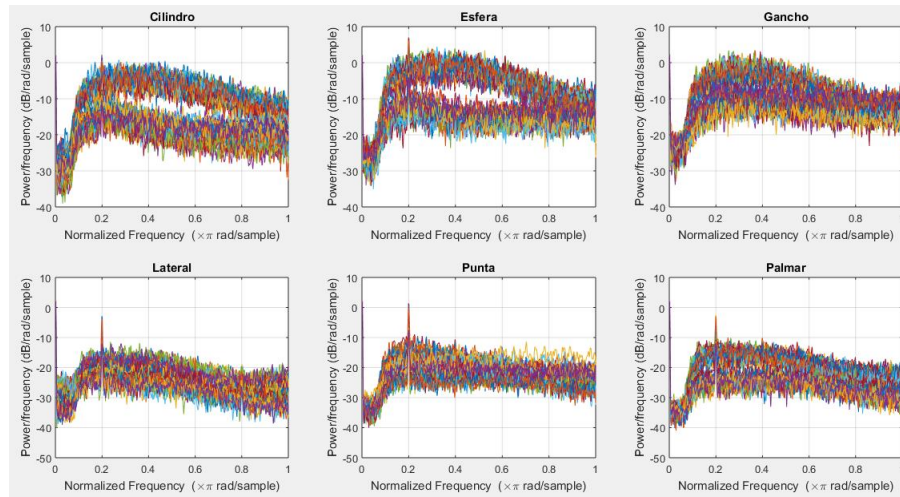


Claramente se observó un comportamiento más predecible en el dominio de las frecuencias ya que este proporcionó características que conforman la señal. Como ya se había mencionado, los tres gestos superiores mostraron ser similares, al igual que el grupo inferior. Anteriormente se enfatizó su similitud solamente en amplitud por muestra, amplitud promedio y desviación estándar, sin embargo, al tener el espectro frecuencial se confirmó con mayor claridad la relación.

Se realizó la Figura correspondiente a la densidad espectral de potencia por medio del método de Welch (Gupta, 2013). La densidad espectral permitió ver la señal en el dominio de las frecuencias eliminando el ruido blanco. Cuando se realizó la aproximación de Welch para la densidad espectral, se produjo un filtrado del ruido blanco, junto con un promediado de las distintas densidades. Esto permitió ver un panorama más claro y reproducible.

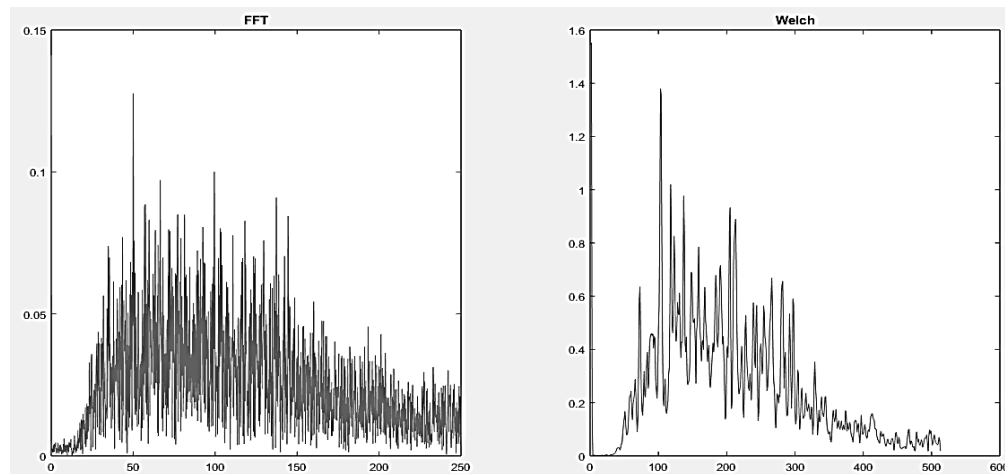
El hecho de contar con este filtrado antes de realizar la clasificación ayudó al algoritmo de reconocimiento para poder obtener características más relevantes para la señal en el dominio de las frecuencias.

Figura 106. Densidad espectral de potencia de cada señal para mujer #1



En la Figura 107 se encuentra una comparación de una de las señales que conforman el gráfico 106, acompañada con transformación de Fourier. Se identificó el fenómeno que resulta al realizar las operaciones necesarias para llegar de la transformada de Fourier al dominio de la potencia espectral.

Figura 107. Comparación de transformada de Fourier con densidad espectral de potencia



Teniendo toda la información en el dominio de Fourier y en PSD (densidad del espectro de potencia), se consolidó cada una en una variable para poder ingresarla a la herramienta de aprendizaje de clasificación. Para el consolidado de la información, se realizó una matriz que contiene todos los datos como lo ilustra el Cuadro 2. En la parte superior se colocó un clasificador, el cual sirvió para determinar la clasificación de cada dato. Cada fila corresponde a una medición de los seis gestos por persona. Los datos se adjuntaron de tal forma que todas las posiciones se encontraran en una fila.

Cuadro 2. Matriz de datos para entrenamiento de máquina de aprendizaje clasificadora

		Cilindro	Gancho	Esfera	Punta	Lateral	Palmar
Clasificador		1	2	3	4	5	6
<b>Mujer 1</b>	Medición 1	[pwelch( Medición 1 )(,1)]	• • •			[pwelch( Medición 1 )(,6)]	
	Medición 2	[pwelch( Medición 2 )(,1)]					[pwelch( Medición 2 )(,6)]
	...	...					...
<b>Mujer 2</b>	Medición 60	[pwelch( Medición 60 )(,1)]					[pwelch( Medición 60 )(,6)]
	Medición 1	[pwelch( Medición 1 )(,1)]					[pwelch( Medición 1 )(,6)]
	Medición 2	[pwelch( Medición 2 )(,1)]					[pwelch( Medición 2 )(,6)]
<b>Mujer 3</b>	...	...					...
	Medición 60	[pwelch( Medición 60 )(,1)]	•				[pwelch( Medición 60 )(,6)]
	Medición 1	[pwelch( Medición 1 )(,1)]		•			[pwelch( Medición 1 )(,6)]
<b>Hombre 1</b>	Medición 2	[pwelch( Medición 2 )(,1)]					[pwelch( Medición 2 )(,6)]
	...	...					...
	Medición 60	[pwelch( Medición 60 )(,1)]				•	[pwelch( Medición 60 )(,6)]
<b>Hombre 2</b>	Medición 1	[pwelch( Medición 1 )(,1)]					[pwelch( Medición 1 )(,6)]
	Medición 2	[pwelch( Medición 2 )(,1)]					[pwelch( Medición 2 )(,6)]
	...	...					...
<b>Hombre 2</b>	Medición 60	[pwelch( Medición 60 )(,1)]					[pwelch( Medición 60 )(,6)]
	Medición 1	[pwelch( Medición 1 )(,1)]					[pwelch( Medición 1 )(,6)]
	Medición 2	[pwelch( Medición 2 )(,1)]					[pwelch( Medición 2 )(,6)]
<b>Hombre 2</b>	...	...					...
	Medición 60	[pwelch( Medición 60 )(,1)]	• • •			[pwelch( Medición 60 )(,6)]	

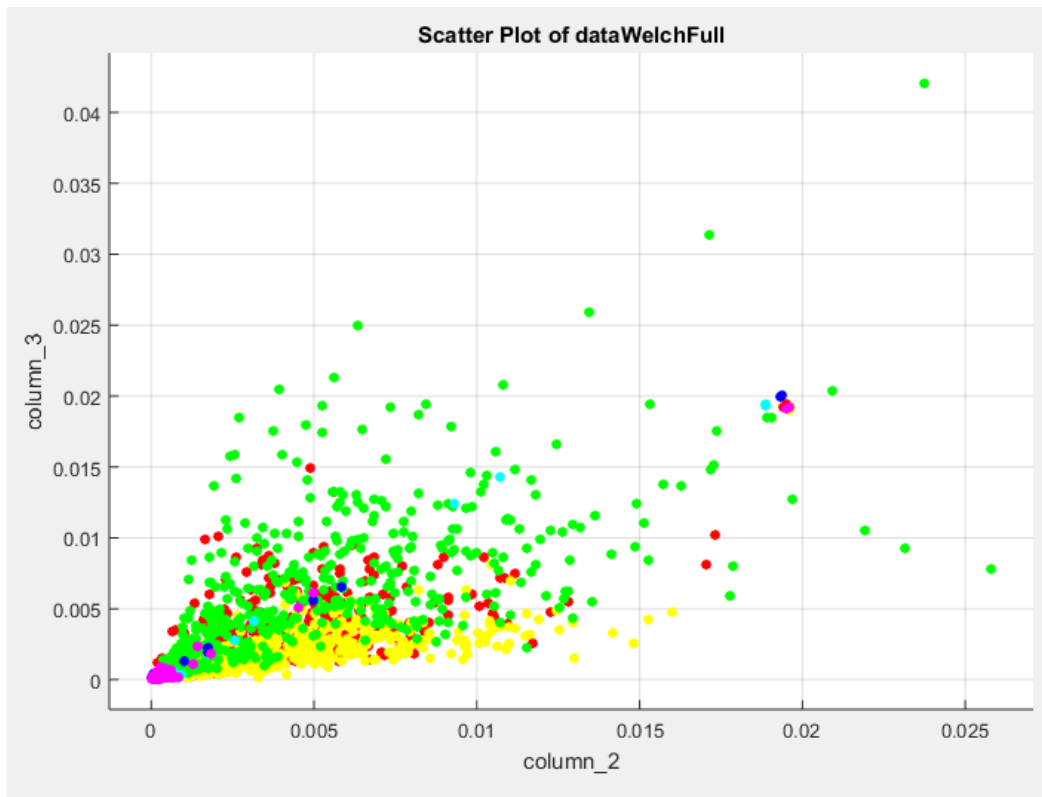
Luego de obtener la matriz como se muestra en el Cuadro, tomando en cuenta que la información que proporcionó la transformada Welch en cada una de las mediciones fue un vector fila y el número clasificador también está conformado por un vector fila con las mismas dimensiones, se contó con una matriz de 301 x 3,078 datos. Análogamente se hizo una matriz de dimensiones 301 x 3,000 para la transformada de Fourier.

Luego de contar con las matrices correspondientes, se inicializó el proceso para entrenar el algoritmo de aprendizaje de clasificación. Este algoritmo de aprendizaje tomó 80% de las mediciones para entrenarse por medio de características identificadoras de cada señal y el otro 20% para validar sus predicciones. Como primera instancia, el algoritmo comparó la información entre las filas de modo combinatorio y asignó un color correspondiente a cada gesto según el clasificador que se encontraba en la primera fila de la matriz.

primera instancia, el algoritmo comparó la información entre las filas de modo combinatorio y asignó un color correspondiente a cada gesto según el clasificador que se encontraba en la primera fila de la matriz.

En la Figura 108 se encuentra la comparación entre las filas 2 y 3 mostrando comportamientos propios de cada señal. Como se puede observar se obtuvo regiones a las que pertenecen los datos, mostrando comportamientos similares en cada región.

Figura 108. Correlación de señales en las columnas 2 y 3 de la matriz de densidad espectral de potencia



Luego se selecciona un tipo de clasificador, en este caso se realizó un enfoque en utilizar máquinas de soporte vectorial lineales (Linear SVM). No obstante, se realizó la comparación con otros métodos como lo son las máquinas de soporte cuadráticas y cúbicas al igual que árboles complejos y simples. En seguida se entrenó el algoritmo de máquina SVM lineal y se obtuvo un porcentaje de rendimiento. En el Cuadro 3 se puede observar el rendimiento de distintas formas de entrenamiento.

Cuadro 3. Rendimiento de algoritmos clasificadores entrenados

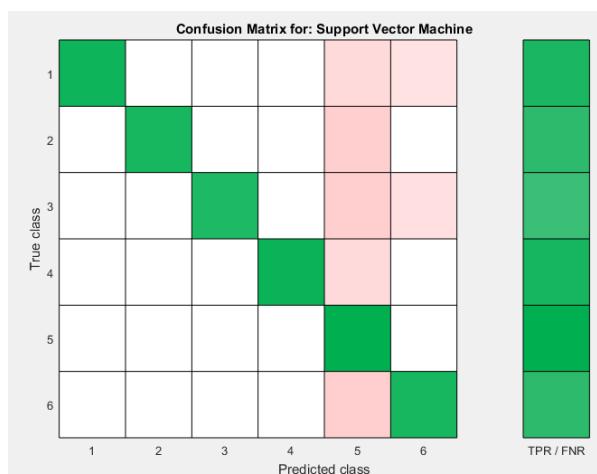
Método	Rendimiento datos Welch	Rendimiento datos Fourier
SVM Lineal	92.8%	97.3%
SVM Cuadrático	94.8%	98.7%
SVM Cúbico	95.0%	99.5%
Árbol Simple	75.3%	73.7%
Árbol Medio	95.4%	89.7%
Árbol Complejo	97.2%	94.3%

Hay que tomar en cuenta que el proceso de selección para los datos de entrenamiento y de validación fue de forma aleatoria, por lo que estos porcentajes varían según el comportamiento de la data y lo representativo que es el muestreo realizado.

El enfoque que se utilizó para la clasificación de los gestos fue por medio de máquinas de vectores de soporte. Como su nombre lo indica, se generaron vectores que dividieron la información según su ubicación en el plano de la Figura 108. Habiendo hecho esto, se entrenó un algoritmo de máquinas de vectores de soporte que lograra clasificar cada una de las seis secciones con un rendimiento del 92.8%.

Luego se generó una matriz de confusión. En esta se identifica claramente cuáles de las regiones son las que presentaron problemas al momento de ser clasificadas, obteniendo gestos con predicciones sobrantes y faltantes. En las filas se encuentra los seis gestos verdaderos y en las columnas se cuenta con la predicción realizada por el algoritmo del 20% que se utilizó para validación.

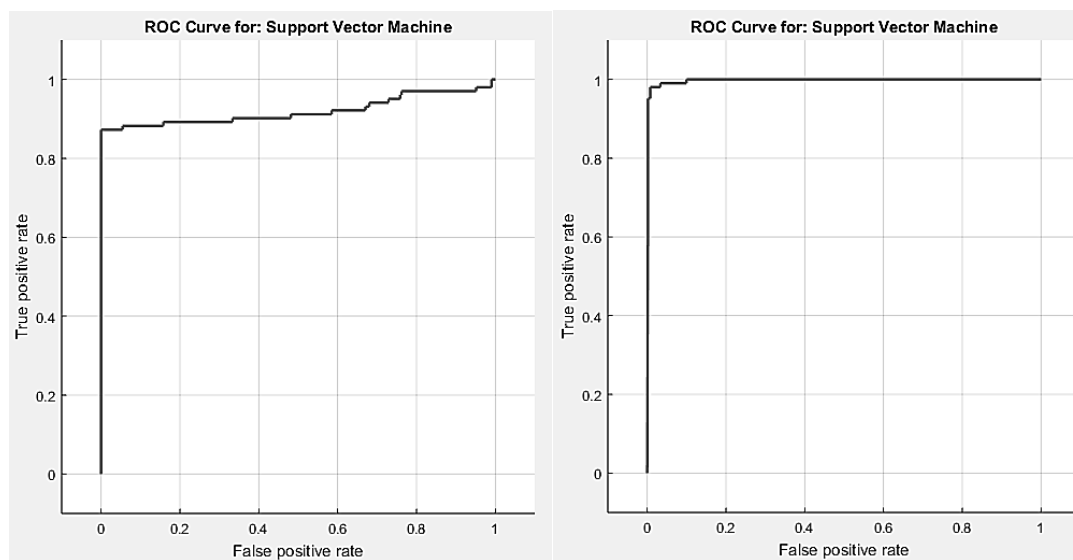
Figura 109. Matriz de confusión para máquina de soporte vectorial para análisis en el dominio de densidad espectral de potencia



Claramente se puede observar que la predicción para el gesto 5 contó con predicciones falsas, debido a que sus características identificadoras resultaron ser poco marcadas como las de los demás, mezclándose así con las regiones aceptables para los demás parámetros. La máquina de soporte vectorial lineal genera vectores que separa toda la información en regiones que describen la clasificación de los datos. Ya que este es un problema de optimización complejo, la herramienta busca la forma de identificar las características de manera que la separación sea lo mejor posible.

En la Figura 110 se encuentra la región de convergencia (ROC) para los gestos 3 y 5. Como se observó en la matriz de confusión, el gesto 5 obtuvo una mayor tasa de predicciones correctas, ya que tuvo un sobredimensionamiento de acertamientos. Esto significa que incluso cuando se tiene un dato que proviene de otro gesto, el clasificador lo incluyó en el resultado del gesto 5. Para este gesto sobredimensionado se contó con un rendimiento porcentual del 99.72% mientras que el gesto 3 tiene un 92.02% de aciertos.

Figura 110. Región de convergencia para los gestos 3 y 5 con análisis de densidad espectral de potencia por medio de SVM



En su totalidad, se logró conseguir un rendimiento del 92.8% para todo el modelo, de tal manera que al momento de ingresar nueva data al clasificador que pertenezca a algunos de los gestos, el clasificador sería capaz de registrarlos como uno de los seis gestos con este rendimiento. Esto quiere decir que mientras más datos se ingresen mejor será el resultado, ya que teniendo extensivas mediciones en un intervalo razonable e ingresándolas al algoritmo, se puede obtener el promedio de estas mediciones y llegar a un resultado que permita clasificar estas mediciones.

## C. Resultados

Para poder validar los resultados de la clasificación de gestos por medio del algoritmo de máquinas de soporte vectorial se realizó una interfaz que analiza datos de forma aleatoria para obtener su porcentaje de predicción. El medio de esta interfaz para lograr validar el porcentaje fue utilizando la matriz que se utilizó para entrenar el código. De esta matriz que contiene los patrones de las señales para todos los gestos de todas las personas, se seleccionó una sección de forma aleatoria y se ingresó al algoritmo de reconocimiento entrenado anteriormente.

En las Figuras 111 a la 116 se puede observar los resultados de tres iteraciones para la transformada de Fourier y tres para la densidad espectral de potencia. En la interfaz se muestran seis distintos gráficos. En la esquina superior izquierda se muestra la señal ingresada al algoritmo durante las últimas 100 iteraciones del código. La siguiente ventana muestra el rendimiento porcentual del algoritmo, seguido de una ventana que muestra la predicción en color verde y el valor real en rojo. La fila inferior cuenta con dos imágenes de los gestos de la mano, la de la izquierda muestra el valor real, mientras que el de la izquierda muestra el predicho. Por último, se encuentra una Figura de barras, en esta se muestra el historial de las predicciones. En azul se encuentra el total, en verde los aciertos y en rojo las falsas predicciones.

Al momento de seleccionar la sección de la matriz de forma aleatoria, se cuenta con el puntero que indica la posición, por lo que fue posible determinar al gesto al que pertenece, al igual que la persona. En base a esto se obtuvo los porcentajes antes mencionados en las gráficas. Cada vez que se seleccionaba los datos a ingresar al algoritmo, se utilizaban los siguientes diez y se repetía el proceso de selección, esto se repitió durante 50 veces para un total de 500 predicciones en cada una de las imágenes mostradas.

Figura 111: Iteración 1 de interfaz Figura para reconocimiento de gestos con FFT

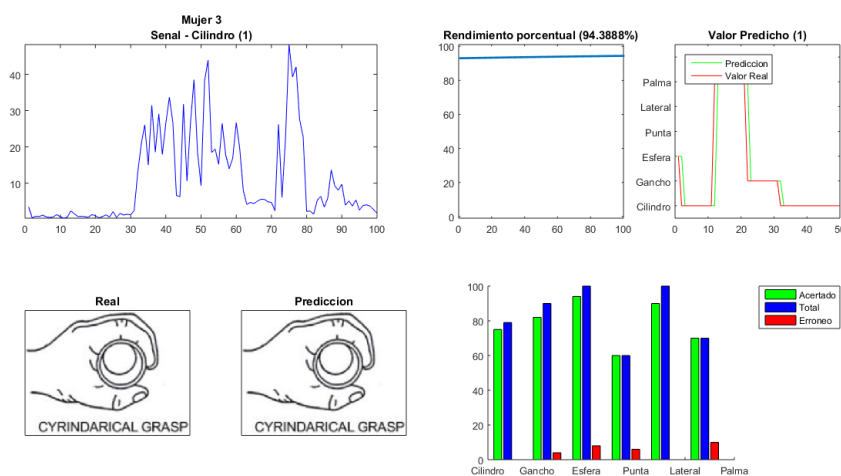


Figura 112: Iteración 2 de interfaz Figura para reconocimiento de gestos con FFT

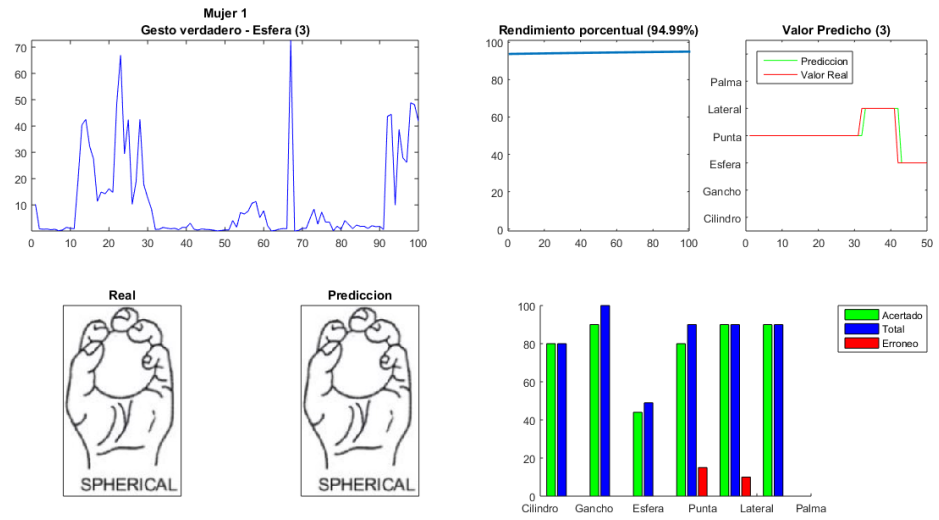


Figura 113. Iteración 3 de interfaz Figura para reconocimiento de gestos con FFT

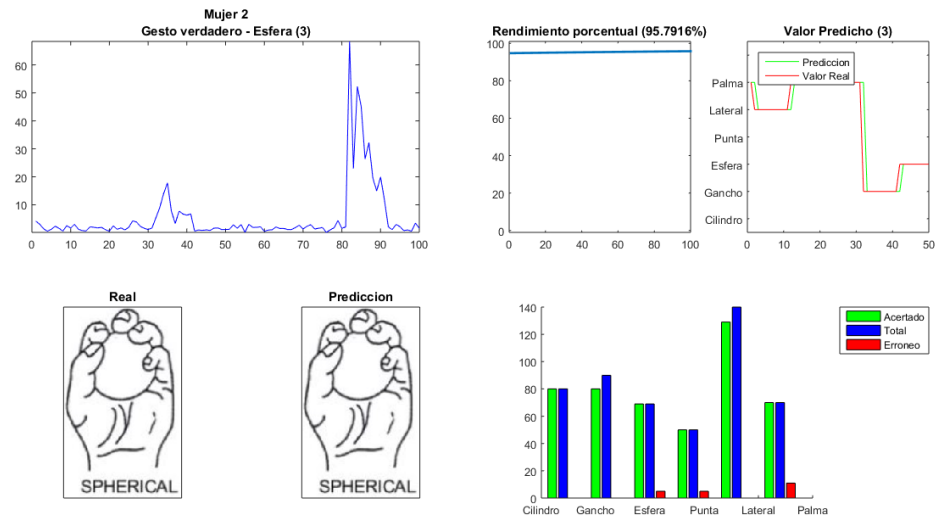


Figura 114. Iteración 1 de interfaz Figura para reconocimiento de gestos con PSD

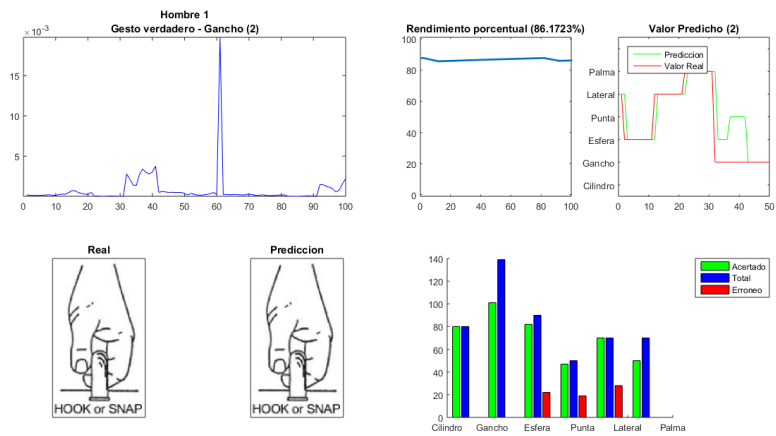


Figura 115. Iteración 2 de interfaz Figura para reconocimiento de gestos con PSD

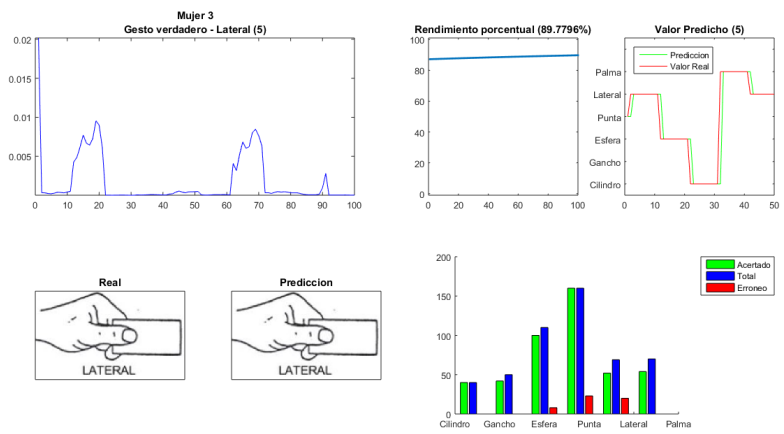
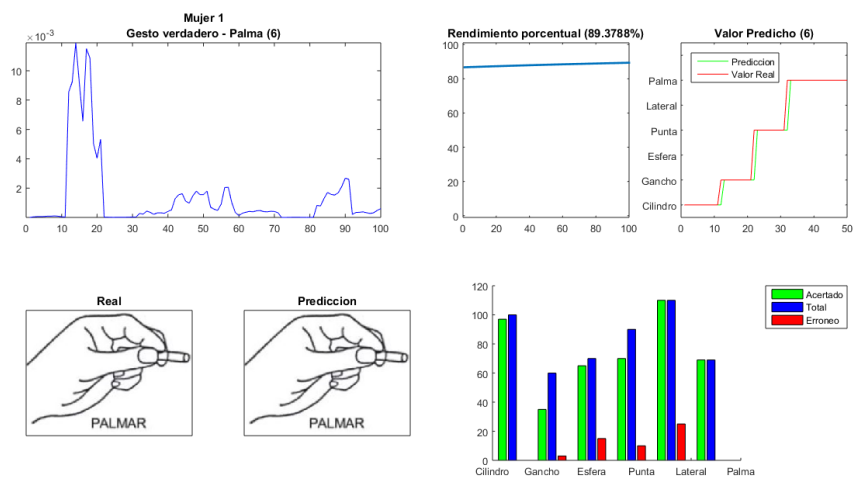


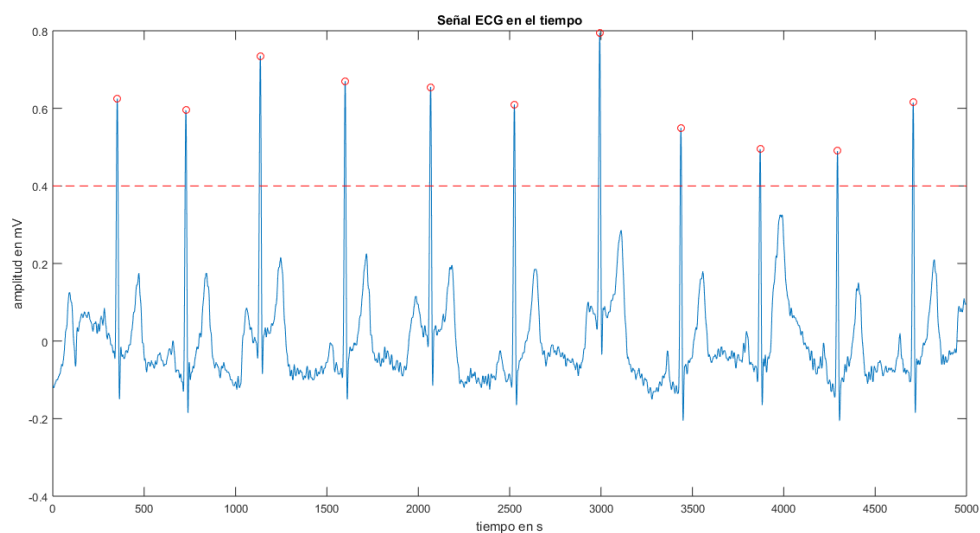
Figura 116. Iteración 3 de interfaz Figura para reconocimiento de gestos con PSD



Como se puede apreciar en las imágenes, se obtuvo un 95% para el proceso con la matriz que contiene la transformada de Fourier de las señales EMG, mientras que la densidad espectral de potencia mostró un 89%.

Para la lectura electro cardiográfica se generó la siguiente Figura, colocando un límite inferior para los picos R que sirvieron para determinar la frecuencia cardíaca. En la Figura se muestra cómo se tomaron los picos que únicamente se encontraron sobre el límite marcado por la línea punteada.

Figura 117. ECG mostrando los picos sobre el límite inferior de 0.4 mV



## D. Discusión

Hasta el momento se cuenta con dos grandes resultados, uno relacionado al reconocimiento de gestos de la mano y el otro al ritmo cardíaco. Ambos fueron con base en datos recolectados por otras universidades que se encuentran a disposición del dominio público.

Para el reconocimiento de ritmo cardíaco con base en estos datos se encontró un 100% de coincidencia para el método de transformada de Fourier y un 100% para el método de umbral. En ambos casos se obtuvo una respuesta lo suficientemente buena para poder detectar un ritmo cardíaco con base en los datos proporcionados. Este ritmo cardíaco se muestra ser constante en la Figura 94, en la cual se cuenta con su respectiva transformada de Fourier.

En este caso, no se pudo detectar arritmia, ya que el análisis espectral no hizo uso del tiempo al cual cada frecuencia se da. La forma en la que se trató fue una sola amalgama de datos, de la cual se seleccionó una ventana de interés y sobre esta se observó el dominio frecuencial. La frecuencia de interés fue la señal

portadora, la cual es la que repite el proceso del consolidado de ondas PQRST. De esta forma se observó únicamente el periodo de la onda, haciendo énfasis en la frecuencia portadora, la cual es la que predominó. En otras palabras, si hay un valor que se encuentra de forma aperiódica dentro de estos datos, simplemente se mostrará como un exabrupto en el dominio de las frecuencias y alterará levemente la frecuencia portadora mostrada.

El reconocimiento de gestos de la mano mostró un comportamiento favorable para la máquina de soporte vectorial para la clasificación de datos. En este caso se contó con resultados mayores al 90% de concurrencia, variando de prueba en prueba por la forma de tomar los datos. Como ya se mencionó anteriormente, la información para la validación de los datos y el entrenamiento se selecciona de forma aleatoria dentro de la matriz compuesta por transformadas de Fourier y al dominio de densidad espectral de potencias.

En ambos casos se obtuvo resultados aceptables dentro de los parámetros para distintos métodos de máquinas de aprendizaje. Aunque en la Figura 107 se mostró que el comportamiento de la señal en el dominio de densidad espectral de potencia obtuvo características más adecuadas para la clasificación que la transformada de Fourier en el Cuadro 3 se pudo observar claramente un mejor rendimiento al momento de ingresarlo al algoritmo clasificador. En este caso, ambos métodos fueron concluyentes, aunque la transformada de Fourier mostró ser más eficiente.

## E. Conclusiones

1. Por razones prácticas se decidió utilizar información recaudada por otras universidades para la detección del ritmo cardíaco al igual que el entrenamiento y reconocimiento de gestos de la mano
2. Se logró identificar seis gestos de la mano, tanto con la transformada de Fourier como con la densidad espectral de potencia, obteniendo resultados más efectivos con la transformada de Fourier por medio de pruebas de validación con 500 datos aleatorios de la matriz de entrenamiento.
3. Se logró clasificar las señales electromiográficas con un 90% utilizando máquinas de soporte vectorial lineal, variando de prueba en prueba por el factor aleatorio de entrenamiento.
4. Se logró obtener un 100% de exactitud del reconocimiento del ritmo cardíaco utilizando dos métodos distintos, utilizando un umbral y empleando la transformada de Fourier.
5. Por complicaciones en el diseño de hardware no se logró concretar mediciones de acelerómetro y giroscopio para el posicionamiento angular del brazo.
6. Se logró realizar un prototipo conceptual de la pulsera, junto con el flujo de procesos para analizar las señales obtenidas por los electrodos y determinar posiciones. Este concepto se logró validar por medio de pruebas realizadas con mediciones de otras universidades.

## F. Recomendaciones

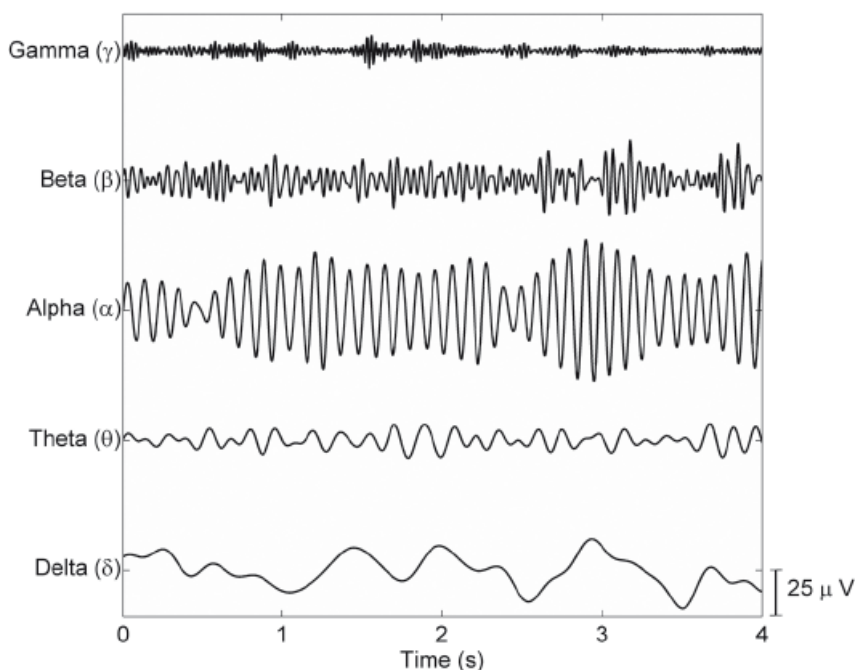
1. Se recomienda entrenar el algoritmo para la persona que lo va a utilizar y ampliar la base de datos por medio de repetición.
2. Se recomienda filtrar la señal antes de realizar el análisis digital, para poder facilitar el proceso de clasificación y reconocimiento de patrones, al igual que el reconocimiento del ritmo cardiaco.
3. Se recomienda probar distintos algoritmos de clasificación, con el objetivo de mejorar los resultados que los obtenidos por una máquina de soporte vectorial lineal.
4. Se recomienda realizar pruebas con datos reales para trascender del análisis computacional a un enfoque aplicado y lograr resultados tangibles.

## VI. MÓDULO 3: COMUNICACIÓN CON CASCO Y PULSERA PARA RECEPCIÓN Y ALMACENAMIENTO DE DATOS

### A. Marco teórico

1. Electroencefalografía y electromiografía. La electroencefalografía es una técnica de visualización médica que lee las señales eléctricas en el cuero cabelludo generadas por la actividad eléctrica del cerebro (Teplan, 2002). La actividad eléctrica del cerebro genera ciertos estados como resultado de las interacciones de las neuronas. Estos patrones resultan en ondas que se caracterizan por su amplitud y frecuencia distintas para cada estado como se puede observar en la Figura 118. Las ondas cerebrales se han clasificado en cinco grupos: delta, theta, alfa, beta y gamma. La actividad eléctrica en el cerebro se genera por dos razones: la primera, se debe a las acciones involuntarias inherentes a una persona, como la respiración, la digestión, el movimiento de las pestañas, etc., mientras la segunda se debe a estímulos externos a través del sentido corporal (Aceves-Fernandez, Fernandez-Fraga, Pedraza-Ortega, & Tovar-Arriaga, 2016).

Figura 118. Clasificación de las señales EEG



Tener una percepción acerca del estado de la mente puede ser un recurso muy valioso en muchas situaciones. Por ejemplo, el monitoreo de somnolencia en los conductores de trenes o autobuses podría evitar un accidente. Los dispositivos llamados EEG headsets pueden servir para estudiar los estados mentales de pacientes que padecen de ansiedad o depresión. Otra posible aplicación implica estudiar la reacción de una

persona ante cierto estímulo. Conocer si un individuo está reaccionando emocional o racionalmente ante un anuncio puede ser utilizado para adaptar el contenido y llevar a cabo un análisis social (Bashivan, Rish, & Heisig, 2016).

La electromiografía (EMG) es una técnica utilizada para evaluar y analizar la actividad eléctrica de los músculos. Es uno de los métodos auxiliares más útiles para la clínica y la histología para analizar desórdenes neuromusculares, que afectan a las neuronas motoras, uniones musculares o fibras musculares esqueléticas (Reyna, 2016). La señal EMG es la manifestación eléctrica de la activación neuromuscular asociado con un músculo que se contrae. Es una señal muy complicada de analizar ya que se ve afectada por las propiedades anatómicas y fisiológicas de los músculos, el esquema de control del sistema nervioso periférico, así como las características de la instrumentación que se utiliza para detectar y observarlo (Basmajian & De Luca, 1985).

Una Interfaz Cerebro – Computadora o BCI es un sistema que descubre patrones específicos en la actividad cerebral de una persona relacionados con su intención de controlar un dispositivo. Estos patrones son detectados en las señales EEG por el sistema BCI y con base en su significado, realiza la acción solicitada por el usuario (Bashashati, Ward, Birch, & Bashashati, 2015). Por su parte, la Interfaz Musculo – Computadora o MCI lee, analiza e interpreta las señales de la electromiografía EMG.

2. **Los artefactos.** Por artefacto se entiende todas aquellas señales que no provienen del cerebro y aparecen durante el análisis de la señal EEG (Garces & Laciari, 2011). Existe una variedad de artefactos que pueden afectar el procesamiento de señales bioeléctricas, algunos más fáciles de detectar y otras que ni se pueden identificar por el ojo de una persona en el encefalograma. Una categorización de los artefactos es con base en su origen, lo cual puede ser técnico o psicológico. Los técnicos se producen durante la adquisición de los datos y sus causas pueden ser la mala colocación de los electrodos, el cuero cabelludo no está limpio, equipos electromagnéticos cercanos a los electrodos, etc. Estos pueden evitarse limpiando el ambiente donde se van a realizar las lecturas y evitando utilizar cualquier equipo innecesario que pueda ser una fuente de ruido. Los de origen fisiológico son imposibles de evitar durante la lectura y para ello existen algoritmos de eliminación de ruido que se encargan justamente de eliminar los artefactos de origen fisiológico. Tales artefactos podrían ser: el parpadeo de los ojos, el ritmo cardiaco, movimiento del glóbulo ocular, movimiento de los músculos, etc. (Sörnmo & Laguna, 2005).

3. **Análisis de señal EEG y EMG con estadísticas.** El proceso de eliminación del ruido es una tarea que no puede obviarse si el objetivo es obtener buenos resultados, y para ello existen varios algoritmos de tipo noise filtering complejos que tratan con este problema. Sin embargo, antes de llegar a estos algoritmos existen otras formas más sencillas de analizar la señal mediante ciertos criterios de rechazo y algunas técnicas de estadística. Aplicando alguno o varios de estas técnicas en conjunto, se puede dar una percepción acerca de la calidad de la señal.

Las propiedades estadísticas de las señales EEG pueden contener información relevante sobre posibles artefactos. Por ejemplo, determinar la tendencia lineal puede ayudar a aislar los datos entrantes que no sigan la tendencia. Otro método más sofisticado es mediante experimentos, determinar la distribución de probabilidad de la señal y luego por cada valor entrante se calcula su probabilidad de ocurrencia y si esta probabilidad está debajo de un threshold determinado, se descarta automáticamente. El análisis de la curtosis de un conjunto de lecturas en un rango de tiempo puede dar indicios acerca de su distribución y comportamientos anormales. De la misma manera, el análisis de la varianza de los datos también puede descubrir anomalías en el comportamiento de la señal (Delorme *et al.*, 2007).

Por último, el método más sencillo para eliminar artefactos que son obvios es definir thresholds o cotas que limitan el rango de valores válidos. De esta manera, cualquier lectura que se salga de este rango es rechazada automáticamente. Este enfoque se basa en la amplitud máxima aceptable de la señal, sin embargo, puede combinarse con el análisis de la varianza definiendo thresholds para los mismos. Los valores que se asignan a los thresholds son valores empíricos o valores obtenidos a partir de experimentos con datos reales (Bashivan *et al.*, 2016).

4. **PySerial.** PySerial es un módulo para Python de código abierto que encapsula el acceso al puerto serial (Liechti, 2016). Soporta las versiones de Python 2.7 y 3.4 en adelante y funciona correctamente para las tres plataformas más populares, Windows, OS X y Linux (Liechti, 2016). Esta librería proporciona un set de funciones que facilitan la comunicación con el puerto serial sin necesidad de descender a un código de bajo nivel o trabajar con protocolos propios de los puertos seriales. En el sistema operativo Windows es suficiente con conocer el puerto COM al que se quiere acceder y se puede comenzar a recibir los datos provenientes del dispositivo conectado a dicho puerto o a enviar instrucciones por el mismo canal.

La librería PySerial permite configurar varios parámetros al momento instanciar el objeto Serial para modificar el comportamiento por defecto que este trae. Otros dos parámetros importantes para crear la conexión son el tiempo de espera (timeout) y la velocidad de transmisión (baud rate). El primero define el tiempo máximo que debe esperarse para obtener una respuesta del puerto serial, mientras que el segundo define que tan rápido debe transmitir y recibir datos el puerto en términos de bits por segundo.

El puerto serial es una interfaz de transmisión de datos en el que cada bit de información se transmite uno tras otro a diferencia de la transmisión en paralelo que permite enviar varios bits al mismo tiempo a través de diferentes canales. Las computadoras modernas, sobre todo las portátiles ya no traen este puerto, sin embargo, existen drivers para cada sistema operativo que permiten crear puertos COM virtuales y la comunicación puede darse a través del puerto USB o por algún medio inalámbrico.

5. **Arduino y Arduino nano.** Arduino es una plataforma electrónica de código abierto que provee software y hardware para implementar sistemas que interactúan con dispositivos físicos o con la computadora, basada en una placa con un microcontrolador y un IDE para programarlo. Las placas Arduino son capaces de leer entradas - la luz en un sensor, un dedo sobre un botón o un mensaje de Twitter - y convertirlo en una salida - la activación de un motor, encender un LED, publicar algo en línea (Arduino, 2016).

El lenguaje de programación utilizado por Arduino es una extensión de los lenguajes C++ y C, lo cual facilita la programación del microcontrolador desde un lenguaje de alto nivel y pone a la disposición del usuario varias de las funciones disponibles en estos lenguajes. Algunas de las ventajas principales de Arduino son:

- a. No es caro. Comparado con otros microcontroladores es relativamente barato y por otro lado les da a los usuarios más experimentados la posibilidad de construir su propio Arduino a mano. Por otro lado, los módulos ya ensamblados tienen un costo menor a los \$50 (Arduino, 2016).
- b. Multiplataforma. El IDE que provee Arduino puede correr sobre Windows, Macintosh OSX y Linux (Arduino, 2016).
- c. Simple ambiente de desarrollo. El IDE que provee Arduino es fácil de utilizar para programadores principiantes como también puede adaptarse a las necesidades de usuarios más avanzados (Arduino, 2016).
- d. Software y hardware extensible. El lenguaje permite el uso de las librerías de C++ y para la gente que quiere entrar en más detalles, puede saltarse directamente a la programación del AVR en C, que es en lo que se basa Arduino (Arduino, 2016). AVR es el microcontrolador utilizado por Arduino cuyo fabricante es Atmel y es programado en C. Por otro lado, los planos de los circuitos de Arduino han sido publicados bajo la licencia Creative Commons lo cual permite que los usuarios más avanzados puedan construir su propio circuito a partir de la base y personalizarlo para ahorrar dinero (Arduino, 2016). Las licencias Creative Commons permiten que los usuarios puedan utilizar herramientas sin fines de lucro manteniendo los derechos de autor (Creative Commons, 2016).
- e. Comunicación serial por USB. Las tarjetas de Arduino permiten comunicarse con la computadora mediante una comunicación serial sobre USB. Además, el software de Arduino trae los drivers FTDI que permiten tener puertos COM virtuales en la PC (Arduino, 2016).

El Arduino Nano es un pequeño microcontrolador de Arduino que a pesar de tener pocos recursos, se puede utilizar para pequeños proyectos que no requieren tanta memoria ni procesamiento. Cuenta con un puerto Mini USB que permite conectarlo con la computadora y programarlo desde el IDE de Arduino. Está disponible en dos versiones, el primero tiene una memoria flash de 16KB y una SRAM (Static RAM) de

1KB, mientras el segundo tiene 32KB de memoria flash y 2KB de SRAM (Arduino, 2016). La memoria flash es donde se carga el código mientras la SRAM se utiliza durante el proceso de ejecución.

Figura 119. Arduino Nano



6. **MongoDB.** MongoDB es un gestor de bases de datos No-SQL de código abierto orientado a documentos desarrollado por MongoDB, Inc. No-SQL es un nuevo paradigma para la gestión de bases de datos y se caracteriza principalmente porque no tiene un esquema de Cuadros fijas como sucede con las bases de datos relacionales. En este gestor de base de datos, en vez de tener Cuadros se tienen colecciones, los cuales consisten en estructuras similares a los objetos JSON y pueden almacenar diferentes tipos de objetos. MongoDB está optimizado para realizar consultas sobre grandes cantidades de datos y para escalar fácilmente. En su página oficial, MongoDB destaca estas fortalezas:

«MongoDB ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad, escalando de una implantación de servidor único a grandes arquitecturas complejas de centros multidados.»

Los objetos almacenados en las colecciones se les llama también documentos y consisten en una lista de pares llave – valor. Los valores puede ser de tres tipos: un valor primitivo, un arreglo de documentos o de nuevo, una lista de pares llave – valor (Suter, 2012). MongoDB permite que los esquemas cambien cuando las aplicaciones crecen, ofreciendo las mismas funcionalidades que ofrecen los manejadores de bases de datos tradicionales, tales como índices, un lenguaje completo para realizar consultas y consistencia estricta (MongoDB, Inc., 2016).

Una de las desventajas de MongoDB es que no soporta referencias hacia otras colecciones (foreign key) y uniones de colecciones (joins), lo cual lo vuelve muy lento cuando se requiere realizar consultas complejas que involucran varios joins. Sin embargo, para aplicaciones que requieren de las operaciones simples INSERT, CREATE, UPDATE y DELETE, MongoDB es mucho más rápido que cualquier gestor de base de datos relacionales.

## B. Antecedentes

Para el análisis de señales EEG y EMG existen en el mercado varias empresas que comercializan productos BCI y/o MCI para usuarios normales. Además, muchos de ellos proveen a los desarrolladores el SDK para que puedan acceder a los datos generados por sus dispositivos, y ejemplo de ellos son los que se mostraron en el Cuadro 1. Para el objetivo principal de este módulo, el cual consiste en la lectura de datos desde el puerto USB utilizando PySerial, se encontró que el proyecto OpenBCI utiliza esta librería para acceder a los datos que generan sus dispositivos.

El proyecto de código abierto está disponible en github y sirvió de base para tomar ejemplos sobre el uso correcto de la librería PySerial. De hecho, fue en este proyecto que se descubrió que la comunicación serial a través de bluetooth se lograba mediante un dongle que se conecta en el puerto USB de la computadora. En la bibliografía se deja el enlace para acceder al repositorio de OpenBCI para Python, ya que también provee el SDK para Java y Node.

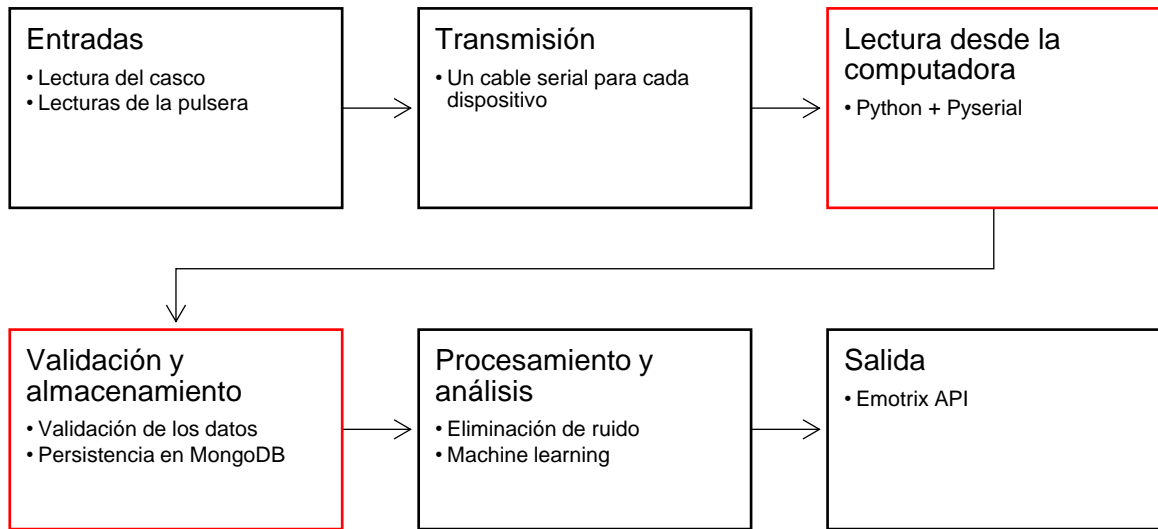
## C. Marco metodológico

1. **Proceso general de Emotrix.** En el siguiente diagrama de flujo se muestra el proceso a nivel macro que Emotrix realiza para leer e interpretar las señales EEG y EMG. El proceso comienza con la toma de los datos mediante los dispositivos de hardware que en este caso son el casco y la pulsera. Estos dispositivos, mientras realizan la lectura transmiten al mismo tiempo los datos hacia la computadora a cierta frecuencia, cada uno mediante un cable serial USB. Luego desde la computadora, utilizando el lenguaje Python y la librería PySerial se conecta a los puertos seriales COM<sup>1</sup> dónde cada dispositivo está transmitiendo y se toman los datos. A continuación, los datos pasan por un pre-filtro para verificar que los valores sean consistentes y lógicos de manera que el módulo de procesamiento trabaje únicamente con lecturas dentro del rango válido. Una vez se ha aceptado una muestra, se procede a almacenarlo para que pueda ser consumido posteriormente por el algoritmo analítico. El resultado es un set de funciones (API) que permiten conocer el estado emocional y algunos signos vitales de la persona que se coloque los dispositivos.

---

<sup>1</sup> Se llaman puertos COM únicamente en Windows. En Linux los puertos seriales se nombran mediante /dev/ttySx (x = 0, 1, 2,..., N), mientras que en Mac OS X se nombran bajo /dev/tty.\* donde \* varía dependiendo del driver. Para mayor información, consultar la documentación de dichas plataformas.

Figura 120. Fases principales de Emotrix



El módulo que se desarrolla en este trabajo se encarga específicamente de las dos fases remarcadas en la Ilustración 2. El resto de la metodología se enfoca en estos dos problemas y describe el proceso que se llevó a cabo para llegar a la solución.

2. **Herramientas.** A continuación, se listan y describen las herramientas tanto de software como de hardware utilizadas durante el proceso de desarrollo de este proyecto.

#### a. **Software**

- Python. Lenguaje utilizado para el desarrollo de la API y para el que será disponible. Se utilizó específicamente la versión 2.7.6.
- PySerial. Librería de Python que permitió la comunicación serial con los dispositivos a través del puerto USB.
- Arduino IDE. Entorno de desarrollo para programar el microcontrolador Arduino Nano. Además, trae los drivers FTDI que permiten emular los puertos COM en Windows.
- MongoDB. Gestor de base de datos No-SQL orientado a documentos, utilizado para persistir los datos en un documento.

#### b. **Hardware**

- Arduino Nano. Tarjeta de Arduino con un microcontrolador ATmega328 de Atmel, programable desde el IDE de Arduino. Este dispositivo sirvió para simular la señal de los dispositivos de entrada. Se necesitaron dos de ellos, uno para el casco y otro para la pulsera.

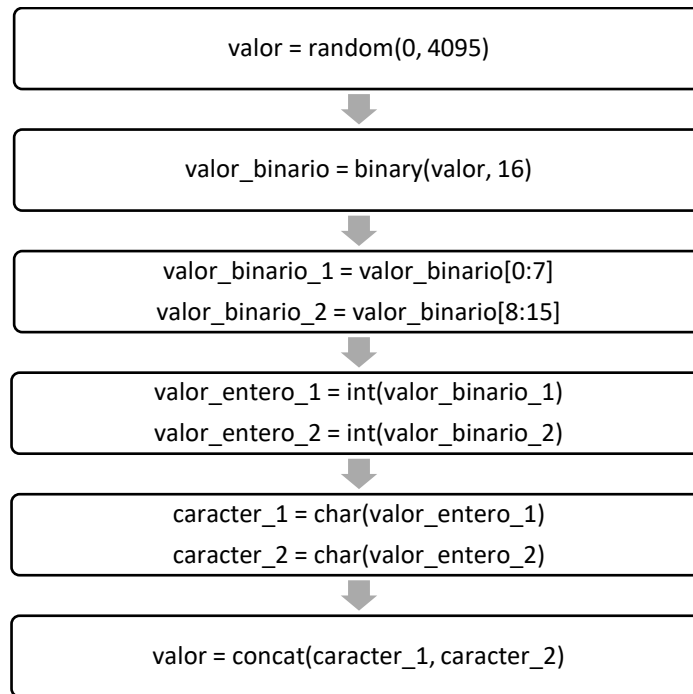
3. **Simulación de dispositivos con el Arduino.** Para poder realizar pruebas de lectura de datos desde el puerto USB era necesario primero tener una fuente de entrada, es decir, un dispositivo que enviara información hacia la computadora a través de un canal serial. Sin embargo, los dispositivos reales con los que va a funcionar el módulo todavía no están disponibles por lo que se tuvo que simular dichos dispositivos a través de un microcontrolador Arduino Nano. Se implementó los programas necesarios para las pruebas desde el IDE de Arduino y se cargó el código al microcontrolador desde el mismo IDE.

Se implementó tres programas distintos para cada dispositivo: uno que emula una buena señal, otro que emula una mala señal y otro que emula la carencia de señal. Lo que cada programa hace es generar valores aleatorios entre un rango predefinido, construir la cadena para una muestra y enviarla a través del canal serial hacia la computadora. Los números aleatorios se generan mediante una distribución uniforme por lo que todos los valores dentro del rango tienen la misma probabilidad de ocurrir. Cuántos datos generar y qué estructura debe tener una muestra, fueron normas definidas junto con los encargados de los módulos que desarrollan estos dispositivos.

a. **Protocolo del casco.** Para esta primera fase del proyecto, el módulo que desarrolla este dispositivo determinó utilizar únicamente 4 electrodos para la toma de datos. Así mismo, también definió que para evitar transmitir demasiados caracteres se iba a convertir los valores de lectura de cada sensor en dos caracteres ASCII. Los valores de lectura que el casco va a estar transmitiendo van a estar entre el rango de 0 a 4095, es decir, un número de 12 bits. La velocidad (baud rate) a la que debe transmitir el dispositivo debe ser la misma que la velocidad en que se lee los datos desde la computadora. El proceso de conversión de los valores decimales a dos caracteres se describe a continuación y se resume en la Figura 121:

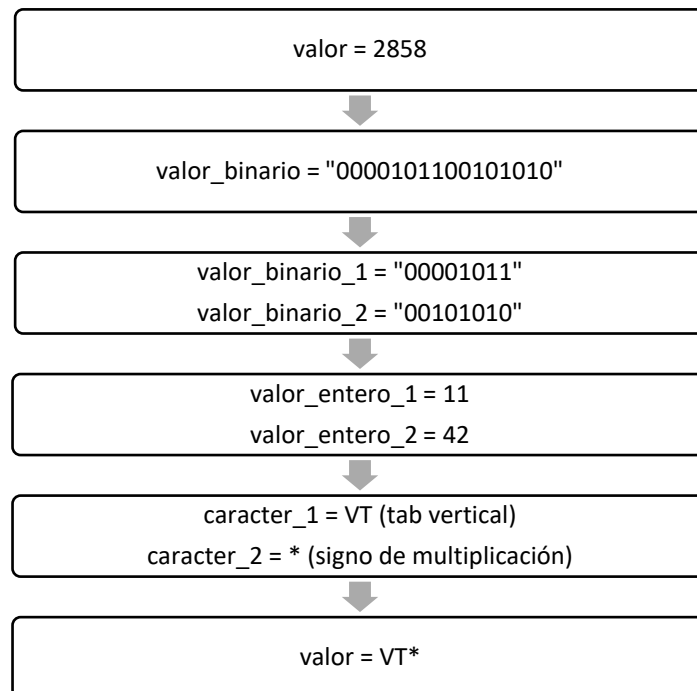
- Se toma el valor entero que corresponde a la lectura del electrodo y se convierte a un número binario de dos bytes.
- Se toman los primeros 8 bits más significativos del binario anterior (los 8 bits de la izquierda) y se convierte a un entero. Este entero se convierte al carácter correspondiente en el Cuadro de caracteres ASCII y este será el primer carácter del valor de la lectura del electrodo. Nótese que al ser un binario de 8 bits, el máximo valor que toma es de 255, lo cual es el valor máximo en el Cuadro caracteres ASCII extendida.
- Luego se toman los últimos 8 bits menos significativos (los de la derecha) y se convierte a un entero. De la misma manera, este entero se convierte al carácter correspondiente y de esta forma se obtiene el segundo carácter del valor de la lectura del electrodo.

Figura 121. Proceso de conversión de las lecturas del casco a 2 caracteres



El proceso anterior puede verse de mejor manera con el ejemplo de la 122. Si el valor de lectura del sensor fuera 2858, el proceso descrito anteriormente aplicaría de la siguiente manera:

Figura 122. Ejemplo de conversión de una lectura del casco a 2 caracteres



La sintaxis que debe tener una muestra para ser considerada sintácticamente válida se muestra a continuación, donde C1 y C2 corresponden al primer y segundo carácter, respectivamente. También se muestra un ejemplo de una muestra que respeta esta nomenclatura.

$$\{s1: C1C2, s2: C1C2, s3: C1C2, s4: C1C2, cs: C1C2\}$$

$$\{s1: KT, s2: HJ, s3: LS, s4: KL, cs: RH\}$$

Cada electrodo se identifica mediante la letra “s” y un número de 0 a 9. Actualmente este número es necesariamente menor a 10 ya que desde la computadora se verifica que la longitud de la cadena sea igual a cierto valor. Esto sugiere que el algoritmo de lectura soporta hasta 9 electrodos, si los identificadores comienzan desde 1. Si comienzan desde 0, entonces soporta hasta 10 electrodos. Nótese que al final de los valores de los electrodos se agrega otro valor cuya identificación es “cs”. Este corresponde al checksum, el cual consiste en la suma de los valores enteros de los electrodos, es decir, la suma de los valores antes de ser convertidas a caracteres. En la discusión se habla acerca de este concepto.

Al igual que con los electrodos, el valor del checksum se convierte a dos caracteres ASCII. Esto sugiere una advertencia y una delimitación, ya que al hablar de dos caracteres significa que se tiene hasta 16 bits para almacenar el valor entero, lo cual implica que el valor máximo que se puede almacenar es  $2^{16} - 1 = 65535$ . Luego, tomando en cuenta que el valor máximo que pueden tener los electrodos es de 4095, dos bytes son suficientes para almacenar la suma de 16 electrodos exactos. Tomando en cuenta la restricción anterior de que la cantidad máxima de electrodos es de 10, no existe ningún problema con el valor máximo que pueda llegar a tener el checksum. Si en el futuro cambia esta restricción, será necesario tomar en cuenta esta otra restricción relacionada con el checksum.

Otra observación importante acerca de las señales que transmite el casco de Emotrix es que al oscilar entre 0 y 4095, se definió que el centro debía estar en 2048.

**b. Simulación del casco.** Debido a la conversión de los valores decimales a caracteres mencionado en el protocolo, fue necesario implementar funciones para convertir un número decimal entero a su correspondiente en binario representado como un string y viceversa. No se encontró en el lenguaje de Arduino funciones nativas que hagan este tipo de conversiones. Las conversiones que se necesitaron fueron las siguientes:

- Convertir la cadena (string) de números binarios “11001101” a decimal.
- Convertir el entero 19563 a binario pero en forma de cadena (string).

### 1) Simulación de buena señal.

Tomando en cuenta que el valor máximo que puede llegar a transmitir el casco para un electrodo va de 0 a 4095, es válido pensar que cuando se está tomando buena señal el valor oscila entre este rango. El procedimiento que se siguió por lo tanto, fue generar números aleatorios entre este rango y aplicar la regla

de conversión a caracteres mencionado anteriormente. Se construye la cadena para la muestra y se transmite a través de la comunicación serial.

**2) Simulación de la ausencia de señal.** En un ambiente libre de ruido, la ausencia de señal significa lecturas iguales a cero, sin embargo, en la realidad los electrodos a pesar de no estar en contacto con la persona, siempre está detectando el ruido del ambiente y por muy pequeño que sea el voltaje que genera, jamás va a ser cero. Por esta razón, al igual que en la simulación de buena señal se definió un rango de oscilación de la señal cuando está leyendo “nada” (nada, refiriéndose a las señales EEG). Tomando en cuenta que la señal debería estar entre 0 y 4095, es admisible suponer que cuando se está leyendo nada, la amplitud máxima que alcanza la señal durante un tiempo determinado (1 segundo, por ejemplo) es de 10. Podría ser otro valor y eso se definirá cuando se realicen las pruebas reales. Por ahora, para efectos de prueba se puede suponer dicho valor.

Para generar la señal mencionada anteriormente no era suficiente con generar valores aleatorios entre 0 y 10. Debido a que el centro de la señal está en 2048, cuando no se detecta señal el valor oscilaría entre 2043 y 2053 tomando en cuenta la suposición de que la amplitud máxima que alcanza es de 10. Siendo así, el procedimiento que se siguió para simular la ausencia de señal fue generando valores aleatorios entre 2043 y 2053 para los 4 electrodos.

**3) Simulación de mala señal.** Cualquier señal que no entre en las dos categorías anteriores es considerada mala señal. Los dos escenarios que cubre este criterio serían cuando la señal se sale del rango válido o cuando la amplitud que alcanza es muy pequeña relativo a la amplitud que alcanza cuando oscila entre 0 y 4095. El procedimiento que se realiza del lado de la computadora para determinar la calidad de la señal toma en cuenta otro criterio, la varianza. Más adelante se describe cómo este concepto de estadística es aplicable para este problema.

**c. Protocolo de la pulsera.** Este dispositivo de entrada tiene pocas restricciones en cuánto al protocolo de comunicación, sin embargo, es necesario definirlos para obtener los resultados esperados. La pulsera transmite únicamente dos variables: el ritmo cardiaco y el valor de la señal EMG tomadas en un instante determinado. A diferencia del casco, por ser pocos datos los que tiene que transmitir, no se hace ninguna conversión a caracteres sino se transmite los valores enteros como tal.

El ritmo cardiaco es un valor que puede ir desde 0 a 255 y el API de Emotrix lo acepta sin problemas, sin embargo, se debe tener cuidado con este dato ya que un ritmo cardiaco de 0 o 255 es ilógico. Según la American Heart Association, el ritmo cardiaco puede ir desde 75 hasta 200 pulsaciones por minuto dependiendo de muchos factores, como la edad, el estado de la persona (trabajando, haciendo ejercicios, descansando, etc.), su zona geográfica, etc. Emotrix descarta automáticamente valores que estén fuera del rango 0 a 255.

El valor permisible para la señal EMG es de 0 a 65535. Valores fuera de este rango, son descartados. Por otro lado, al igual que el casco la pulsera también incluye el checksum de los datos en cada muestra, que

consiste en la suma del valor del ritmo cardiaco más el valor de la señal EMG. Por último y no menos importante, está claro que el dispositivo debe transmitir (baud rate) a la misma velocidad que se lee desde la computadora.

La sintaxis que debe tener una muestra para ser considerada sintácticamente válida se muestra a continuación. También se muestra un ejemplo de una muestra que respeta esta nomenclatura.

$$\{ "bpm": N1, "emg": N2, "chksum": SUMA(N1, N2) \}$$

$$\{ "bpm": 202, "emg": 43553, "chksum": 43755 \}$$

#### d. Simulación de la pulsera

**1) Simulación de buena señal.** Siguiendo el mismo procedimiento de la simulación del casco, se generaron números aleatorios para cada campo en el rango válido correspondiente a cada uno: de 75 a 200 para el ritmo cardiaco y de 0 a 65535 para la señal EMG.

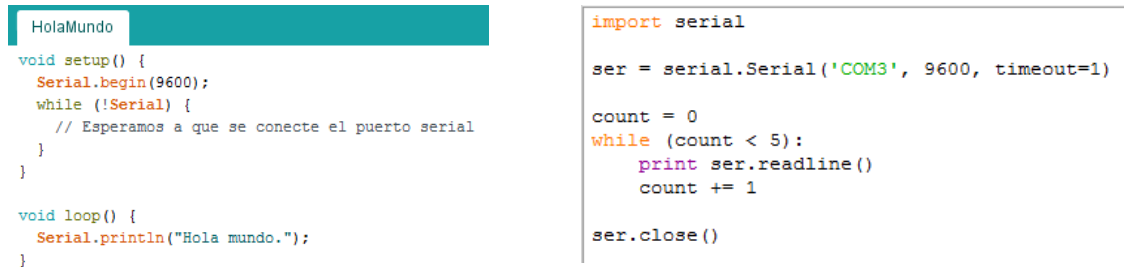
**2) Simulación de la ausencia de señal.** Para el ritmo cardiaco, se definió que el rango para suponer que se está leyendo nada es de 0 a 10, de igual manera sujeto a cambios cuando se realicen las pruebas reales. En el caso de la señal EMG, también se definió que la señal oscila en una amplitud máxima de 10 centrado en cero cuando el dispositivo está leyendo nada. Dado que la señal oscila entre 0 y 65535, que haya ausencia de señal significa que está oscilando entre 32763 y 32773 suponiendo que la señal está centrada en 32768.

**3) Simulación de mala señal.** Para el ritmo cardiaco se generó valores aleatorios que no están dentro del rango válido del ritmo cardiaco pero que tampoco representa ausencia de señal. Para la señal EMG se generó valores aleatorios entre 0 y 3000, lo cual es indicio de una señal pobre si se compara contra el rango de buena señal. De igual manera, el análisis realizado desde la computadora involucra análisis de varianza.

### 4. Lectura de los datos

**a. Recepción de los datos a través del puerto USB.** Para la primera prueba de concepto de lectura de datos a través del puerto USB, se implementó un simple programa en Arduino que envía una cadena de caracteres a la computadora (ie. "Hola mundo") y desde Python se implementó otro programa que tratara de leer este texto desde el puerto USB utilizando el módulo PySerial. En la Figura 123 se muestra el código implementado en Arduino y Python para esta prueba de concepto y el resultado fue exitoso al imprimirse cinco veces el texto "Hola mundo" en el Shell de Python. El procedimiento seguido fue primero escribir el programa en el IDE de Arduino y luego cargarlo al dispositivo Arduino Nano. Para ello fue necesario desde el IDE configurar el tipo de tarjeta que en este caso fue Arduino Nano, configurar el tipo de procesador de la tarjeta (el que se utilizó tiene ATmega328) y el puerto COM al que se encuentra conectado el dispositivo. Una vez configurado dichos parámetros, se procedió a cargar el programa al microcontrolador.

Figura 123. Prueba de concepto para la recepción de datos utilizando PySerial



```

HolaMundo
void setup() {
  Serial.begin(9600);
  while (!Serial) {
    // Esperamos a que se conecte el puerto serial
  }
}

void loop() {
  Serial.println("Hola mundo.");
}

```

```

import serial

ser = serial.Serial('COM3', 9600, timeout=1)

count = 0
while (count < 5):
  print ser.readline()
  count += 1

ser.close()

```

Luego que la prueba de concepto fue exitosa se procedió a programar las funciones que la API va a disponer para cada dispositivo comenzando con el casco que es el más importante. En realidad, las funciones implementadas para cada uno fueron las mismas ya que para ambos dispositivos se utilizó la comunicación serial, sin embargo, fueron desarrollados en diferentes programas ya que la estructura de los datos que recibe cada uno recibe no es la misma. En la sección de resultados se describen las funciones implementadas para la API de Emotrix, sin embargo, se recomienda leer los siguientes tres conceptos para poder entender lo que hace y devuelve cada función.

**b. Criterio para verificar la calidad de la señal.** Los algoritmos que procesan las señales EEG y EMG para determinar el estado de una persona, implementan sus propios métodos de eliminación del ruido tomando únicamente las frecuencias que les interesa. Sin embargo, una de las funciones que están relacionados con la conectividad de los dispositivos es la de verificar la calidad de la señal que se está recibiendo y por consiguiente, es responsabilidad de este módulo implementar dicha funcionalidad. Para ello se implementó un análisis basado en un modelo estadístico para estudiar el comportamiento de la señal y dar una estimación de la calidad de la señal. A continuación, se detalla el proceso que se realizó para determinar la calidad de la señal (por señal se refiere a un conjunto de lecturas).

El análisis se basó en el estudio de la varianza de las lecturas que se encuentran en el *Buffer de datos temporales*, el cual consiste en el valor esperado del cuadrado de las diferencias de cada dato respecto de la media. Siguiendo el mismo procedimiento utilizado para simular una buena señal, se buscó una cota superior y una cota inferior para la varianza, de modo que para evaluar que una señal sea efectivamente buena señal, su varianza debería estar en el rango limitado por estas cotas. Para encontrar las cotas de la varianza de la señal EEG, se definió que la cantidad mínima de muestras para calcular la varianza debe ser de 70 (en la literatura, esto se llama Region Of Interest en sus siglas ROI, o Window Size) y de esta manera, se calculó 1000 veces la varianza de un conjunto de 70 muestras de forma aleatoria tal como se simuló la buena señal. Luego, la cota inferior es la varianza más pequeña obtenida en las 1000 iteraciones, mientras que la cota superior es la varianza más grande obtenida.

Para la señal EMG se aplicó el mismo procedimiento, pero la cantidad de muestras para calcular la varianza es de 300. En la discusión se habla sobre la razón por la que estos valores difieren uno de otro. Por

último, la que difiere completamente de estos dos es la determinación de buena señal para el ritmo cardiaco (bpm) ya que para este en vez de analizar la varianza, lo que se hizo fue promediar las últimas 300 muestras de “bpm” y verificar si el promedio está entre el rango válido definido para el ritmo cardiaco. Si se cumple esta condición, la señal es considerada buena. Por otro lado, puede ser también ausencia de señal cuando el valor oscila entre 0 y 10.

c. **Buffer de datos temporales.** Este buffer almacena las lecturas tomadas durante el último segundo partiendo desde el último dato insertado. La razón por la que se implementó este buffer es porque el módulo que implementa el algoritmo para analizar la señal EEG necesita trabajar sobre trozos o segmentos de lecturas (ROI o Window Size) y en este caso, necesita trabajar sobre las lecturas tomadas durante el último segundo. Entonces para optimizar la consulta de dichos valores, se simuló un buffer que vaya guardando únicamente las lecturas del último segundo y cada vez que llega una nueva muestra, verifica si el primero de la lista ingresó hace más de un segundo. Si es así, lo elimina de la lista para tener únicamente las del último segundo. El valor de un segundo puede cambiarse a dos segundos, tres segundos, etc. Por otro lado, la función que verifica la calidad de la señal toma todas las lecturas que están actualmente en el buffer y calcula la varianza de estos datos para determinar si la señal es buena o mala.

d. **Verificación del checksum.** El checksum es un concepto en telecomunicaciones que sirve para validar la consistencia de los datos durante una transmisión. En el caso de Emotrix, se aplicó únicamente sobre los valores de lectura. Para el casco, el checksum consiste en la suma de los valores de los cuatro electrodos y se manda junto con la muestra. Del lado de la computadora cuando se recibe dicha muestra se verifica que la suma de los valores de los electrodos sea igual al valor del checksum. Si no es así, la información está corrupta y se descarta la muestra completa. En el caso de la pulsera, el checksum consiste en la suma de la señal EMG más el valor del ritmo cardiaco (bpm) y se hace la misma verificación de consistencia y completitud.

e. **Resumen de procedimientos.** Las Figuras 124 y 125 resumen el procedimiento general seguido para simular los dispositivos casco y pulsera respectivamente, y la obtención de los datos desde la computadora a través de PySerial. Nótese que la principal diferencia radica en la conversión a caracteres que se aplica a las lecturas del casco antes de trasmitirlas.

Figura 124. Proceso general para la simulación del casco y la recepción de los datos

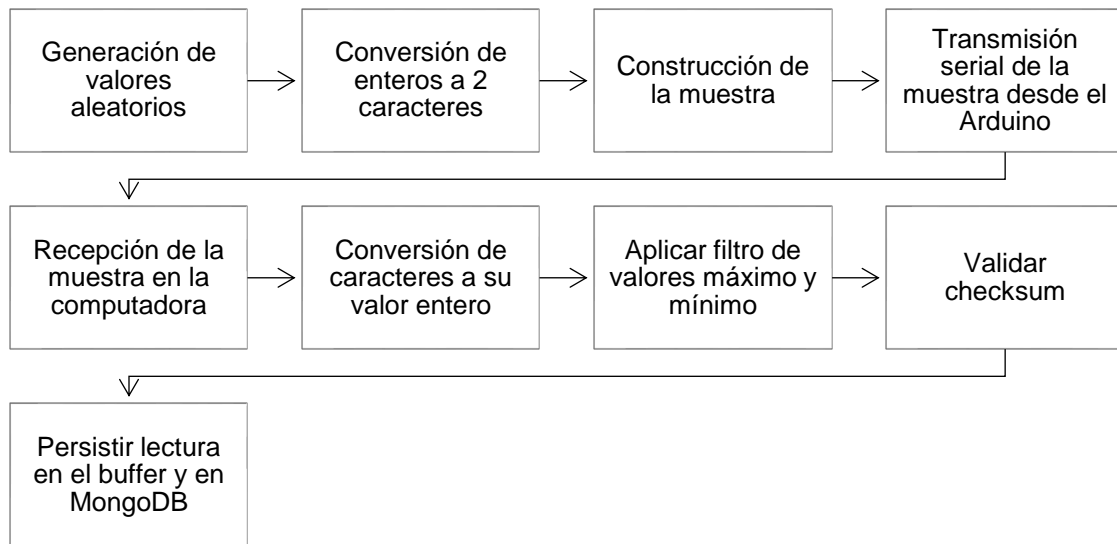
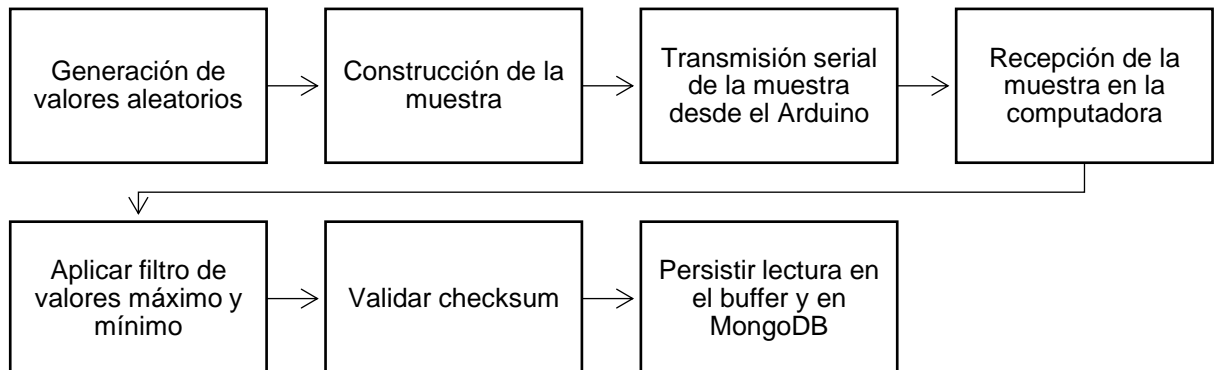


Figura 125. Proceso general para la simulación de la pulsera y la recepción de los datos



## D. Resultados

1. **Funciones del casco para la API.** Para representar los dispositivos físicos, se utilizó el paradigma de Programación Orientado a Objetos, de manera que los métodos que la API ofrece se llamen sobre el objeto relacionado con el dispositivo físico. Así, la representación del dispositivo casco en Emotrix es la clase llamada Headset.

a. **def \_\_init\_\_(self, logging\_level=logging.ERROR).** Esta función es el constructor del objeto Headset y se muestra entre las funciones del casco porque puede recibir un parámetro que puede llegar a ser de gran utilidad para los desarrolladores. Las clases que representan los dispositivos de Emotrix utilizan un módulo llamado “logging” que básicamente es una bitácora que lleva el control de los mensajes lanzados en

tiempo de ejecución por Emotrix. Este módulo maneja varios niveles de mensajes, pero Emotrix solo utiliza los de tipo INFO y WARNING. El parámetro que puede recibir el constructor es el nivel de logging y define hasta qué tipos de mensajes se deben mostrar. Para efectos de depuración, se recomienda pasar como valor a este parámetro la constante logging.INFO ya que esto muestra tanto los mensajes de tipo INFO como los de tipo WARNING. Para profundizar en los detalles se recomienda revisar la documentación del módulo logging.

Por otro lado, en el constructor del objeto casco se establece la conexión con el servidor de MongoDB y si esto no es posible, Emotrix lanza un error. Esto sugiere que antes de ejecutar Emotrix se debe correr desde una terminal el servidor de Mongo para que los dispositivos puedan conectarse y almacenar sus lecturas. No es necesario tener creada la base de datos para poder correr Emotrix ya que el cliente de mongo que se utiliza en Python la crea si no existe. Si se quiere consultar los datos que Emotrix almacena, la base de datos se llama “*emotrix\_db*”.

Otro de los procedimientos importantes que se realiza en el constructor es el establecimiento de las cotas de la varianza máxima y mínima para evaluar la calidad de la señal EEG.

b. **def connect(self, port, baudrate).** Esta función intenta establecer conexión con el dispositivo identificado con el puerto que recibe de parámetro. En Windows, los puertos seriales se identifican con el texto “COM” más un número entero identificador. También necesita recibir el parámetro baudrate, el cual consiste en la velocidad a la que se quiere transmitir/recibir los datos en bits por segundo. El parámetro timeout utilizado en la prueba de concepto es configurado por la función internamente y el valor que toma es de un segundo. Si por alguna razón PySerial no puede establecer la conexión, después de un segundo la librería lanza una excepción. Emotrix solo la captura y se lo muestra al desarrollador. Si no se da ningún error, la conexión con el dispositivo queda establecida y ya se puede empezar a leer los datos.

c. **def isConnected(self).** Esta función devuelve el estado de conexión del dispositivo. Si la conexión ya fue establecida y no se ha cerrado el puerto, devuelve True. De lo contrario devuelve False.

d. **def closePort (self).** Cuando se llama esta función se cierra la conexión con el puerto serial. Sin embargo, para poder cerrar la conexión primero hay que estar conectado y segundo, no se debe estar leyendo. Si el puerto es cerrado con éxito, devuelve True. De lo contrario devuelve False.

e. **def startReading(self, persist\_data=False).** Este método le avisa al objeto Headset que comience a leer los datos que están llegando al puerto al se encuentra conectado. Si el dispositivo no está conectado lanza un error. Además, puede recibir un parámetro booleano que define si se deben persistir las lecturas en la base de datos de MongoDB o sólo se almacena en el buffer de memoria del objeto.

f. **def stopReading(self).** Detiene el proceso de lectura, pero no cierra el puerto. Si se está leyendo el puerto serial, es necesario llamar este método antes de cerrar el puerto.

g. **def getStatus()**. Devuelve el estado de los sensores del casco en cuanto a la calidad de la señal que está recibiendo la computadora. Devuelve un diccionario de Python con la siguiente estructura:

```
{"s1": [0|1|3], "s2": [0|1|3], ... }
```

El valor “[0|1|3]” representa una expresión regular y significa que el valor que devuelve esta función para cada sensor puede ser 0, 1 o 3. La calidad de la señal es 0 cuando se detecta nada, 1 cuando es mala y 3 cuando es buena.

2. **Funciones de la pulsera para la API.** Siguiendo el mismo concepto de programación orientado a objetos, la representación de la pulsera dentro de Emotrix está definida por la clase Bracelet. Esta clase define los mismos métodos que el objeto Headset y lo único que cambia son las implementaciones internas. De hecho, a nivel de programación ambos objetos implementan una interfaz que representa un dispositivo de entrada.

Las dos diferencias principales respecto a la clase del casco son el método constructor y el método getStatus(). El constructor de la pulsera, aparte de inicializar la base de datos para almacenar las lecturas, también establece las cotas de la varianza, pero para la señal EMG. Por otro lado, el método getStatus() devuelve también un diccionario únicamente con los campos “bpm” y “emg”, que son las dos lecturas de la pulsera. De igual manera, devuelve 0 cuando no hay señal, 1 cuando la señal es mala y 3 cuando es buena.

La estructura del diccionario que devuelve getStatus() es la siguiente:

```
{"bpm": [0|1|3], "emg": [0|1|3]}
```

Los métodos de Bracelet para la API serían las siguientes:

- a. `def __init__(self, logging_level=logging.ERROR)`
- b. `def connect(self, port, baudrate)`
- c. `def isConnected(self)`
- d. `def closePort(self)`
- e. `def startReading(self, persist_data=False)`
- f. `def stopReading(self)`
- g. `def getStatus(self)`

3. Prueba de conectividad. La Figura 126 muestra un programa utilizando las funciones de la API para verificar la conectividad de un dispositivo de entrada, en este caso se trata del casco. Para fines de demostración, se le pasa la constante logging.INFO al constructor del casco para que muestre los mensajes de tipo INFO y tipo WARNING. La Figura 127 muestra una conexión exitosa, mientras la Ilustración 128 muestra una conexión fallida.

Figura 126. Programa de Python para prueba de conectividad

```
from Headset import Headset
import logging
import time

puerto = 'COM3'
headset = Headset(logging.INFO)

try:
    headset.connect(puerto, 115200)
except Exception, e:
    raise e

print "Is conected? " + str(headset.isConnected())
print "-----"
time.sleep(5)
headset.closePort()
print "-----"
print "Is conected? " + str(headset.isConnected())
```

Figura 127. Output de conectividad exitosa

```
>>>
INFO:Headset:Starting mongo client...
INFO:Headset:MongoDB server connection established.
INFO:Headset:Connecting to port 'COM3'...
INFO:Headset:Connection to port 'COM3' established.
Is conected? True
-----
INFO:Headset:Closing port COM3...
INFO:Headset:Port COM3 closed successfully.
-----
Is conected? False
>>> |
```

Figura 128. Output de conexión fallida

```
>>>
INFO:Headset:Starting mongo client...
INFO:Headset:MongoDB server connection established.
INFO:Headset:Connecting to port 'COM3'...

Traceback (most recent call last):
  File "C:\Users\CesarLuis\Documents\COLLEGE\Megaproyecto\dev\emotrix\emotrix\HeadsetConnectivityTester.py", line 11, in <module>
    raise e
SerialException: could not open port 'COM3': WindowsError(2, 'The system cannot find the file specified.')
>>> |
```

4. **Lecturas del casco.** La Figura 129 muestra el programa de Python que se utilizó para realizar las pruebas de lectura del casco. Las funciones que se llaman sobre la variable headset son las implementadas para la API. El baud rate que se usa en este programa es de 115200 y será el que use en todas las pruebas. La razón de usar un valor muy grande es para transmitir la mayor cantidad de datos posibles. PySerial soporta valores más grandes, sin embargo, el Arduino Nano soporta hasta este valor. En la discusión se habla sobre este tema.

Lo que el programa hace es conectarse al dispositivo conectado al puerto virtual COM3 y comenzar la lectura de datos sin guardarlos en MongoDB. Espera por 5 segundos para que el programa lea una buena cantidad de muestras y luego detiene el proceso de lectura. Al final se consulta el estado del dispositivo para verificar la calidad de la señal.

Figura 129. Programa de Python para la lectura del casco

```

from Headset import Headset
import logging
import time

puerto = 'COM3'
headset = Headset(logging.INFO)

try:
    headset.connect(puerto, 115200)
except Exception, e:
    raise e

print "Is conected? " + str(headset.isConnected())
print "-----"
headset.startReading(persist_data=False)
time.sleep(5)
headset.stopReading()
headset.closePort()
print "-----"
print "Is conected? " + str(headset.isConnected())
print headset.getStatus()

```

a. **Simulación de buena señal.** La Figura 130 muestra el resultado de correr el programa de la Figura 129 recibiendo una buena señal. El output dice que se recibieron 311 lecturas y ninguna fue descartada, sin embargo, el cálculo del estado de los sensores se calculó sobre 89 lecturas, lo cual significa que partiendo desde el momento en que se mandó a llamar getStatus() y 1 segundo atrás, se recibieron 89 muestras. Es lo que había en el buffer de datos temporales. Lo más importante de esta prueba es el resultado de la función getStatus(), que muestra que todos los sensores están leyendo buena señal, recordando que un 3 es signo de buena señal.

Figura 130. Output de prueba de buena señal del casco

```

>>>
INFO:Headset:Starting mongo client...
INFO:Headset:MongoDB server connection established.
INFO:Headset:Connecting to port 'COM3'...
INFO:Headset:Connection to port 'COM3' established.
Is conected? True
-----
INFO:HeadsetThreadReader:Starting thread: Headset data reader
INFO:HeadsetThreadReader:Exiting thread: Headset data reader
INFO:HeadsetThreadReader:Samples received = 311, Samples ignored:0
INFO:Headset:Closing port COM3...
INFO:Headset:Port COM3 closed successfully.
-----
Is conected? False
INFO:Headset:Status calculated on 88 samples.
{'s3': 3, 's2': 3, 's1': 3, 's4': 3}
>>> |

```

b. **Simulación de mala señal.** La Figura 131 muestra el resultado de una simulación de mala señal. Esta salida es interesante ya que hubo una muestra incompleta y por lo tanto fue ignorado. El resultado de la ejecución es que todos los sensores tienen una calidad de señal de 1, lo cual indica que es mala señal.

Figura 131. Output de prueba de mala señal del casco

```

>>>
INFO:Headset:Starting mongo client...
INFO:Headset:MongoDB server connection established.
INFO:Headset:Connecting to port 'COM3'...
INFO:Headset:Connection to port 'COM3' established.
Is conected? True
-----
INFO:HeadsetThreadReader:Starting thread: Headset data reader
WARNING:HeadsetThreadReader:Missing curly braces: {s1:|È,s2:|M,s3:|J,s4:|;|,cs:T
0
INFO:HeadsetThreadReader:Exiting thread: Headset data reader
INFO:HeadsetThreadReader:Samples received = 249, Samples ignored:1
INFO:Headset:Closing port COM3...
INFO:Headset:Port COM3 closed successfully.
-----
Is conected? False
INFO:Headset:Status calculated on 72 samples.
{'s3': 1, 's2': 1, 's1': 1, 's4': 1}
>>> |

```

c. **Simulación de ausencia de señal.** La Figura 132 muestra el resultado de la simulación de la ausencia de señal. Recordando, la ausencia de señal en el casco se da cuando los valores recibidos oscilan entre 2043 y 2053 y el estado que el API devuelve es 0 para todos los sensores.

Figura 132. Output de prueba de ausencia de señal del casco

```

>>>
INFO:Headset:Starting mongo client...
INFO:Headset:MongoDB server connection established.
INFO:Headset:Connecting to port 'COM3'...
INFO:Headset:Connection to port 'COM3' established.
Is conected? True
-----
INFO:HeadsetThreadReader:Starting thread: Headset data reader
INFO:HeadsetThreadReader:Exiting thread: Headset data reader
INFO:HeadsetThreadReader:Samples received = 312, Samples ignored:0
INFO:Headset:Closing port COM3...
INFO:Headset:Port COM3 closed successfully.
-----
Is conected? False
INFO:Headset:Status calculated on 88 samples.
{'s3': 0, 's2': 0, 's1': 0, 's4': 0}
>>>

```

5. **Lecturas de la pulsera.** La Figura 133 muestra el programa de Python que se utilizó para realizar las pruebas de lectura de la pulsera. Las funciones que se llaman sobre la variable `bracelet` son las implementadas para la API. Al igual que para el casco, el baud rate que se usa es de 115200 y será el que use en todas las pruebas. El programa hace exactamente lo mismo que el utilizado para las lecturas del casco; se llaman exactamente las mismas funciones, pero sobre un objeto `Bracelet` y se sigue el mismo flujo.

Figura 133. Programa de Python para la lectura de la pulsera

```

from Bracelet import Bracelet
import logging
import time

puerto = 'COM3'
bracelet = Bracelet(logging.INFO)

try:
    bracelet.connect(puerto, 115200)
except Exception, e:
    raise e

print "Is conected? " + str(bracelet.isConnected())
print "-----"
bracelet.startReading(persist_data=False)
time.sleep(5)
bracelet.stopReading()
bracelet.closePort()
print "-----"
print "Is conected? " + str(bracelet.isConnected())
print bracelet.getStatus()

```

a. **Simulación de buena señal.** La Figura 134 muestra el resultado de correr el programa de la Figura 133 con datos de la simulación de buena señal para la pulsera. Como puede observarse, el resultado

final dice que el cálculo de la varianza de las últimas 300 muestras refleja que se está leyendo buena señal, tanto para la señal EMG como para el ritmo cardiaco BPM, recordando que 3 es signo de buena señal. Otra observación importante es que para la lectura de la pulsera se obtienen más datos en un segundo de los que se obtienen para el casco, y la razón es que el proceso para construir una muestra para el casco es más complejo que el proceso para construir una muestra de la pulsera (conversión a caracteres) desde el Arduino. Esto se detalla mejor en la discusión.

Figura 134. Output de prueba de buena señal de la pulsera

```
>>>
INFO:Bracelet:Starting mongo client...
INFO:Bracelet:MongoDB server connection established.
INFO:Bracelet:Connecting to port 'COM3'...
INFO:Bracelet:Connection to port 'COM3' established.
Is conected? True
-----
INFO:BraceletThreadReader:Starting thread: Bracelet data reader
INFO:BraceletThreadReader:Exiting thread: Bracelet data reader
INFO:BraceletThreadReader:Samples received = 1058, Samples ignored:0
INFO:Bracelet:Closing port COM3...
INFO:Bracelet:Port COM3 closed successfully.
-----
Is conected? False
INFO:Bracelet:Status calculated on 300 samples.
{'emg': 3, 'bpm': 3}
>>> |
```

b. **Simulación de mala señal.** La Figura135 muestra los resultados de una simulación de mala señal para la pulsera. El estado de las dos señales está en uno a pesar de haber sido calculado sobre 324 muestras. Recordando que una mala señal para el ritmo cardiaco sucede cuando el promedio de las lecturas del último segundo no está entre el rango válido definido para el mismo.

Figura 135. Output de prueba de mala señal de la pulsera

```
>>>
INFO:Bracelet:Starting mongo client...
INFO:Bracelet:MongoDB server connection established.
INFO:Bracelet:Connecting to port 'COM3'...
INFO:Bracelet:Connection to port 'COM3' established.
Is conected? True
-----
INFO:BraceletThreadReader:Starting thread: Bracelet data reader
INFO:BraceletThreadReader:Exiting thread: Bracelet data reader
INFO:BraceletThreadReader:Samples received = 1142, Samples ignored:0
INFO:Bracelet:Closing port COM3...
INFO:Bracelet:Port COM3 closed successfully.
-----
Is conected? False
INFO:Bracelet:Status calculated on 324 samples.
{'emg': 1, 'bpm': 1}
>>> |
```

c. **Simulación de ausencia de señal.** La Figura 136 muestra el resultado producido por una simulación de ausencia de señal de la pulsera. El estado 0 para las dos variables significa que se está leyendo una señal muy muy pequeña.

Figura 136. Output de prueba de ausencia de señal de la pulsera

```
>>>
INFO:Bracelet:Starting mongo client...
INFO:Bracelet:MongoDB server connection established.
INFO:Bracelet:Connecting to port 'COM3'...
INFO:Bracelet:Connection to port 'COM3' established.
Is conected? True
-----
INFO:BraceletThreadReader:Starting thread: Bracelet data reader
INFO:BraceletThreadReader:Exiting thread: Bracelet data reader
INFO:BraceletThreadReader:Samples received = 1096, Samples ignored:0
INFO:Bracelet:Closing port COM3...
INFO:Bracelet:Port COM3 closed successfully.
-----
Is conected? False
INFO:Bracelet:Status calculated on 310 samples.
{'emg': 0, 'bpm': 0}
>>>
```

6. **Prueba de checksum.** Para afectar la comprobación del checksum, en el programa desde el Arduino se generó un nuevo valor random entre 0 y 1 y se le sumó al valor que se había generado primero. Al no sumarlo a la suma del checksum, cuando el API trata de realizar la validación desde la computadora encuentra que no hay consistencia en los datos y por lo tanto rechaza la muestra. Como resultado, se observa en la Figura 137 una parte de las muestras ignoradas porque no cumplen con la validación del checksum, siendo un total de 287 muestras ignoradas de un total de 302.

Este escenario muestra otra de las restricciones de Emotrix y se trata de la cantidad mínima de muestras que requiere para aproximar la calidad de la señal. Cuando no se cuenta con la cantidad mínima de muestras, el API devuelve cero para todos los sensores.

Figura 137. Output de alteración del checksum para el caso

```
WARNING:HeadsetThreadReader:Invalid data. Checksum doesn't match: 10152
{'s3': 1173, 's2': 3732, 's1': 4093, 's4': 1155}
WARNING:HeadsetThreadReader:Invalid data. Checksum doesn't match: 5419
{'s3': 2597, 's2': 631, 's1': 688, 's4': 1507}
WARNING:HeadsetThreadReader:Invalid data. Checksum doesn't match: 8672
{'s3': 2664, 's2': 1750, 's1': 2258, 's4': 2001}
WARNING:HeadsetThreadReader:Invalid data. Checksum doesn't match: 6533
{'s3': 1614, 's2': 2958, 's1': 276, 's4': 1688}
INFO:HeadsetThreadReader:Exiting thread: Headset data reader
INFO:HeadsetThreadReader:Samples received = 302, Samples ignored:287
INFO:Headset:Closing port COM3...
INFO:Headset:Port COM3 closed successfully.
-----
Is conected? False
INFO:Headset:Not enough data to check signal quality. 7 found.
{'s3': 0, 's2': 0, 's1': 0, 's4': 0}
>>> |
```

7. Prueba de persistencia de datos. Las pruebas mostradas hasta ahora no persistían los datos que se leía. Para decirle al objeto que persista los datos, se debe pasar True al método `startReading(persist_data=True)`. Para demostrar la funcionalidad, se utilizó los programas que simulan buena señal para ambos dispositivos. La Figura 138 muestra una consulta de mongo desde el Shell de Windows en el que se busca todas las lecturas del casco. Se observa que cada registro se guarda con un id, el timestamp en que se leyó y los valores enteros de los cuatro sensores. Por otro lado, la Figura 139 muestra una consulta sobre las lecturas de la pulsera y como se puede observar, los registros se guardan con un id, el timestamp del momento en que se leyó y los valores enteros para EMG y BPM.

Figura 138. Lecturas del casco almacenados en MongoDB

```
> db.headset_data.find()
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e4c"), "readed_at" : "1477397155", "s3"
  : 2828, "s2" : 333, "s1" : 552, "s4" : 939 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e4d"), "readed_at" : "1477397155", "s3"
  : 2131, "s2" : 3865, "s1" : 2057, "s4" : 3556 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e4e"), "readed_at" : "1477397155", "s3"
  : 732, "s2" : 3125, "s1" : 989, "s4" : 2123 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e4f"), "readed_at" : "1477397155", "s3"
  : 3421, "s2" : 318, "s1" : 371, "s4" : 2560 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e50"), "readed_at" : "1477397155", "s3"
  : 2780, "s2" : 795, "s1" : 3513, "s4" : 3579 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e51"), "readed_at" : "1477397155", "s3"
  : 2216, "s2" : 1072, "s1" : 2962, "s4" : 371 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e52"), "readed_at" : "1477397155", "s3"
  : 685, "s2" : 1870, "s1" : 3481, "s4" : 3610 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e53"), "readed_at" : "1477397155", "s3"
  : 1355, "s2" : 1094, "s1" : 361, "s4" : 2534 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e54"), "readed_at" : "1477397155", "s3"
  : 3420, "s2" : 2395, "s1" : 2859, "s4" : 2077 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e55"), "readed_at" : "1477397155", "s3"
  : 2396, "s2" : 3564, "s1" : 1098, "s4" : 3469 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e56"), "readed_at" : "1477397155", "s3"
  : 3646, "s2" : 1552, "s1" : 1996, "s4" : 4083 }
< "_id" : ObjectId("580f4aa3f1adb34f8c6a7e57"), "readed_at" : "1477397155", "s3"
```

Figura 139. Lecturas de la pulsera almacenadas en MongoDB

```
> db.bracelet_data.find()
< "_id" : ObjectId("580f6e2bf1adb37708b95f8c"), "readed_at" : "1477406251", "emg"
  : 15800, "bpm" : 151 }
< "_id" : ObjectId("580f6e2bf1adb37708b95f8d"), "readed_at" : "1477406251", "emg"
  : 56659, "bpm" : 159 }
< "_id" : ObjectId("580f6e2bf1adb37708b95f8e"), "readed_at" : "1477406251", "emg"
  : 15083, "bpm" : 181 }
< "_id" : ObjectId("580f6e2bf1adb37708b95f8f"), "readed_at" : "1477406251", "emg"
  : 20951, "bpm" : 174 }
< "_id" : ObjectId("580f6e2bf1adb37708b95f90"), "readed_at" : "1477406251", "emg"
  : 1856, "bpm" : 163 }
< "_id" : ObjectId("580f6e2bf1adb37708b95f91"), "readed_at" : "1477406251", "emg"
  : 33699, "bpm" : 100 }
< "_id" : ObjectId("580f6e2bf1adb37708b95f92"), "readed_at" : "1477406251", "emg"
  : 1810, "bpm" : 97 }
< "_id" : ObjectId("580f6e2bf1adb37708b95f93"), "readed_at" : "1477406251", "emg"
  : 17099, "bpm" : 157 }
< "_id" : ObjectId("580f6e2bf1adb37708b95f94"), "readed_at" : "1477406251", "emg"
  : 15813, "bpm" : 101 }
< "_id" : ObjectId("580f6e2bf1adb37708b95f95"), "readed_at" : "1477406251", "emg"
  : 47939, "bpm" : 175 }
< "_id" : ObjectId("580f6e2bf1adb37708b95f96"), "readed_at" : "1477406251", "emg"
```

## E. Discusión

El proceso de recepción y almacenamiento de los datos es una etapa crucial para que Emotrix obtenga los resultados esperados. Las funciones necesarias para establecer y verificar la conexión con los dispositivos fueron todas implementadas y el módulo está listo para integrarse con los otros módulos del proyecto, esperando que los cambios que tengan que hacerse sean muy pocas y que sean cambios quizás relacionados a los valores de algunas constantes que por ahora se han definido de forma empírica. Además, el código desarrollado para el casco no se limita únicamente a los cuatro sensores que se utilizan en esta primera fase de Emotrix, sino se ha hecho para ser compatible con un casco de hasta diez sensores sin cambiar absolutamente nada. A continuación, se discuten algunos puntos importantes acerca del trabajo realizado para proveer una mejor realimentación a una posible siguiente fase de Emotrix.

Actualmente los valores utilizados para simular los dispositivos fueron generados con la función `random` lo cual está muy lejos de la realidad si se compara con la señal que realmente transmiten los electrodos. Los números aleatorios que produce la computadora no son números completamente aleatorios, en realidad son valores que se les llama pseudoaleatorios ya que son generados por un proceso determinista, es decir, si a la función se le especifica un valor de semilla (`seed`), siempre va a generar la misma secuencia de números. Además, normalmente la función `random` tiene una distribución uniforme para generar los números mientras que las señales EEG y EMG no tienen una función de densidad fija, ya que los valores son muy variantes en el tiempo y debido al ruido que afecta la señal, es difícil predecir su comportamiento. Por esta razón, el hecho de generar números aleatorios para simular el dispositivo es sólo útil hasta el punto de realizar pruebas para envío y recepción de datos. Pero, para la simulación de la calidad de la señal se queda muy corto y no refleja exactamente la realidad. La mejor forma de analizar la calidad de la señal es utilizando datos reales, datos que provienen directamente ya sea del casco o de la pulsera.

Hablando de la calidad de la señal, el algoritmo implementado para resolver este problema utiliza una herramienta estadística para analizar el comportamiento de la señal: la varianza. Se determinó el valor máximo y el valor mínimo que puede llegar a tener la varianza de un número determinado de lecturas. En la electrónica digital, se refiere a estas cotas como `thresholds`. El procedimiento que se siguió para determinar las cotas de la varianza trabaja con números aleatorios y eso hasta cierto punto es sólo un estimador de la calidad de la señal porque como se mencionó anteriormente, la señal se comporta completamente diferente. Sin embargo, este método puede llegar a ser muy eficiente si las cotas mínima y superior se calculan con datos reales provenientes de los dispositivos, repitiendo varias veces el experimento descrito en la metodología hasta converger en un valor mínimo y un valor máximo de la varianza. Por supuesto, no se trata únicamente de generar datos con los electrodos y encontrar estas cotas, sino se debe asegurarse que la señal sobre la que está realizando el experimento sea efectivamente una buena señal. Para esto se necesita del ojo humano para visualizar en una Figura el estado de la señal e identificar cuando la señal es buena para realizar este procedimiento. Una vez determinadas las cotas que rigen el rango válido de la varianza de la señal, esta metodología puede identificar picos en la señal, calidad de señal pobre (amplitud pequeña), demasiado ruido

(amplitud muy grande) y la pérdida de la señal. Cabe mencionar que antes de llegar a la etapa de definir las cotas mediante el análisis de la varianza, primero se tiene que pasar por el filtro de valor mínimo y valor máximo de una lectura para que el algoritmo trabaje únicamente con datos que se encuentran dentro del rango válido. Por ejemplo, para el casco los valores deben estar entre 0 y 4095 por lo que cualquier lectura que no se encuentre dentro de este rango, no puede formar parte del set de datos sobre el que se calcula la varianza.

Otro punto importante que vale la pena mencionar es la cantidad de muestras sobre las que se calcula la varianza. Durante el cálculo de los thresholds de la varianza, para la señal EEG se utilizan 70 muestras mientras que para la señal EMG se utilizan 300 muestras. Por otro lado, cuando se consulta la calidad de la señal (resultado de llamar la función `getStatus()` sobre alguno de los objetos `Headset` o `Bracelet`), tanto para la señal EEG como para la señal EMG se calcula la varianza de las lecturas del último segundo partiendo justo en el momento en que se hace la consulta. De este criterio provienen los valores 70 y 300 definidos para calcular los thresholds. Y la razón es que no es recomendable calcular la varianza de una pequeña cantidad de lecturas porque el análisis no tendría sentido, es necesario definir una cantidad mínima de lecturas aceptable para poder consultar la calidad de la señal. Dicho en otras palabras, no se puede consultar la calidad de la señal cuando apenas se han realizado dos lecturas, por ejemplo.

Para el análisis de la varianza, entre más datos se tiene mejor es el resultado ya que se puede ver mejor el comportamiento de la señal. Sin embargo, la velocidad de transmisión (baud rate) impone la limitación de cuántas muestras se puede leer en un segundo. El microcontrolador del Arduino Nano limita el baud rate hasta 115200 debido a los recursos limitados que posee, mientras que PySerial al correr desde la computadora, soporta valores de hasta 4000000 (Liechti, 2016). Agregado a esto, los procesos para construir las muestras de cada dispositivo difieren en complejidad debido a que el casco necesita convertir a caracteres sus lecturas antes de transmitirlos, lo cual implica más tiempo de procesamiento y menos capacidad para transmitir los datos. Por esta razón específicamente, después de realizar varias pruebas se concluyó que la cantidad mínima de muestras que puede transmitir el casco en un segundo es de 70 mientras que la pulsera puede transmitir al menos 300 muestras en un segundo. Para el análisis de la varianza, se tomaron estos valores como la cantidad de muestras mínima sobre la que se calcula la misma ya que como se mencionó anteriormente, entre más valores haya mejor es el resultado que se obtiene. Vale la pena mencionar que el casco real va a transmitir a un baud rate de 425,000 utilizando otro microcontrolador más potente y teóricamente PySerial debería soportarlo.

Otro concepto que es necesario mencionar es el proceso de validación del checksum. Como se mencionó en el protocolo de cada dispositivo, el checksum consiste en la suma de los valores que se quiere transmitir y se manda junto a la muestra. Esta verificación es muy útil y necesaria porque permite identificar si una muestra no llegó completo a su destino o se está transmitiendo mal. Sin embargo, tiene también algunas desventajas y la primera está relacionado con lo mencionado anteriormente sobre la cantidad de muestras que cada dispositivo logra transmitir. Es claro que, si se quiere transmitir el checksum junto con la muestra, en el casco del casco la longitud de la cadena incrementa en 6 caracteres y eso implica menos muestras para

transmitir en un segundo. En el caso de la pulsera no afecta mucho ya que la cantidad de muestras que se leen en un segundo es relativamente aceptable. La segunda desventaja del checksum es que no hay garantía de que el valor que indica es efectivamente la suma de los valores transmitidos. En el peor de los casos, si durante la transmisión se corrompe específicamente el checksum, la validación de la misma desde la computadora fallaría porque dictaría que los datos son inconsistentes cuando en realidad el valor inconsistente es el propio checksum. A pesar de estas desventajas, esta verificación es necesaria ya que da la seguridad y la garantía de que se está trabajando con datos consistentes y ayuda a encontrar posibles errores de transmisión ingresadas por los desarrolladores.

Finalmente, se aclara nuevamente que este módulo se limitó únicamente a pruebas utilizando el Arduino Nano como dispositivo, de modo que el nivel de confianza del análisis de la señal mediante la varianza es del 100%. Sin embargo, cuando se realicen las pruebas con los dispositivos reales, esto no va a ser necesariamente igual. Antes de utilizar este método para verificar la calidad de la señal, es necesario fijar las cotas de la varianza mediante pruebas experimentales hasta converger en las varianzas mínima y máxima.

## F. Conclusiones

1. Las funciones de la API de Emotrix que permiten la conectividad con los dispositivos fueron todas implementadas y permiten realizar las siguientes acciones: conectar con el dispositivo, verificar si el dispositivo está conectado, comenzar la lectura de datos, detener el proceso de lectura, obtener la calidad de la señal y cerrar la conexión.
2. La eficiencia del análisis de la varianza para determinar la calidad de la señal simulada fue del 100%. La función para verificar la calidad de la señal es capaz de detectar cuando se lee nada, cuando la señal es mala y cuando la señal es buena.
3. El procedimiento seguido para calcular las cotas de la varianza que rigen el rango para que una señal sea considerada buena señal, es solo un estimador de la calidad de la señal debido a que utiliza números aleatorios para calcular las varianzas extremas.
4. Haber utilizado un Arduino Nano para simular los dispositivos tuvo como limitante la velocidad de transmisión (baud rate), cuyo valor máximo es hasta 115200 y como consecuencia, para la simulación de mala señal EEG apenas se alcanza leer 71 muestras por segundo.
5. A pesar de las desventajas que presenta agregar la validación del checksum, este proceso es muy importante para identificar anomalías en los datos y para facilitar la depuración de los datos recibidos.
6. La conversión de los valores que transmite el casco a caracteres implica que el protocolo de Emotrix soporta hasta 16 electrodos, de modo que el valor máximo de la suma de los valores no exceda los 16 bits.
7. A pesar de que el protocolo soporta hasta 16 electrodos, el API de Emotrix soporta hasta 10 electrodos por la forma en que identifica cada sensor numerándolos de 1 al 9 (pudiéndose incluir

también el 0) y la verificación de que la longitud de la cadena recibida sea igual a un valor específico. El cálculo de este valor específico sólo considera electrodos identificados con la letra “s” más un dígito (ej. s4).

## G. Recomendaciones

1. Para que el método de análisis de la señal mediante la varianza sea eficaz, se recomienda primero establecer las cotas inferior y superior de forma experimental y segundo, buscar transmitir lo más rápido posible (aumentando el baud rate) de manera que se reciba la mayor cantidad de muestras por segundo posibles.
2. Para corregir el problema de la limitación del número de electrodos, se recomienda cambiar el protocolo del casco para que se transmitan únicamente los valores de los sensores separados por comas, insertando hasta el final el valor del checksum. Además, sería más eficiente transmitir los valores enteros en vez de hacer la conversión a caracteres. Con el protocolo actual se reduce la cantidad de datos a transmitir, pero se pierde mucho tiempo durante el proceso de conversión a caracteres.
3. Para implementar la comunicación por bluetooth utilizando PySerial, se recomienda implementar un adaptador de bluetooth (dongle) mediante un Arduino conectado a un módulo de bluetooth y por aparte, implementar otro sistema Arduino + Módulo de bluetooth que realice la tarea de simular los dispositivos. Así, el simulador del dispositivo transmite la información al dongle mediante bluetooth y este a su vez, conectado por USB a la computadora se identifica como un puerto serial COM para que PySerial ya lo pueda manejar.

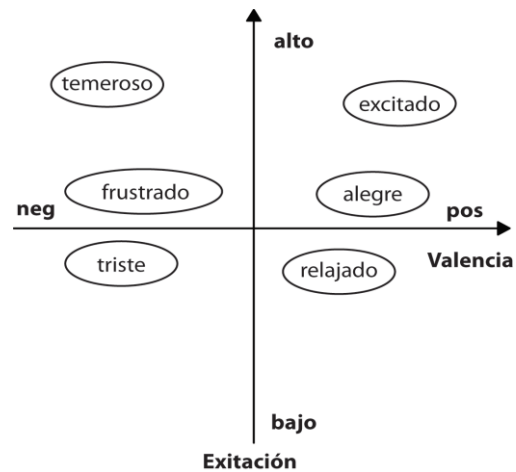
# **VII. MÓDULO 4: DESARROLLO DE SDK ENFOCADO EN EL PROCESAMIENTO DE SEÑALES EEG PARA LA CLASIFICACIÓN Y EL RECONOCIMIENTO DE EMOCIONES**

## **A. Marco teórico**

1. Emociones. Las emociones se consideran una simple expresión que se componen de pensamientos y sentimientos, pero están íntimamente relacionadas con procesos internos inconscientes (Valderrama, 2011). Desde hace mucho tiempo, se ha tratado de detectar las emociones utilizando las ondas electroencefalográficas (EEG). El procedimiento a grandes rasgos, consiste en la lectura de la actividad neuronal proveniente de la zona cortical del cerebro en períodos sumamente cortos. Luego se procede a la identificación de las emociones a través de cambios fisiológicos. De esta manera los científicos han llegado a construir grandes bases de datos de análisis de emociones (Valderrama, 2011).

La existencia de una gran cantidad de emociones hace que sea tan difícil el proceso de reconocimiento. Además, no existen como estados totalmente independientes a lo largo del tiempo en un sujeto, sino que pueden traslaparse unas con otras. Un buen modelo de emociones es aquel, que puede separarlas de forma clara y concisa. Diversos modelos han tratado de definir un conjunto de emociones básicas y representarlas en un modelo dimensional. Las seis emociones básicas son: Asco, miedo, sorpresa, miedo, alegría y tristeza (Friesen, 1982). El modelo dimensional, comúnmente utilizado se caracteriza por tener dos dimensiones: valencia y excitación. Los rangos de valencia en las emociones pueden ir de negativo a positivo, mientras que los rangos para la excitación van de la calmado a excitado. Este modelo es ampliamente utilizado en muchos estudios, porque permite expresar y clasificar las emociones de una manera muy sencilla. (Russell, 1980)

Figura 140. Modelo dimensional de emociones

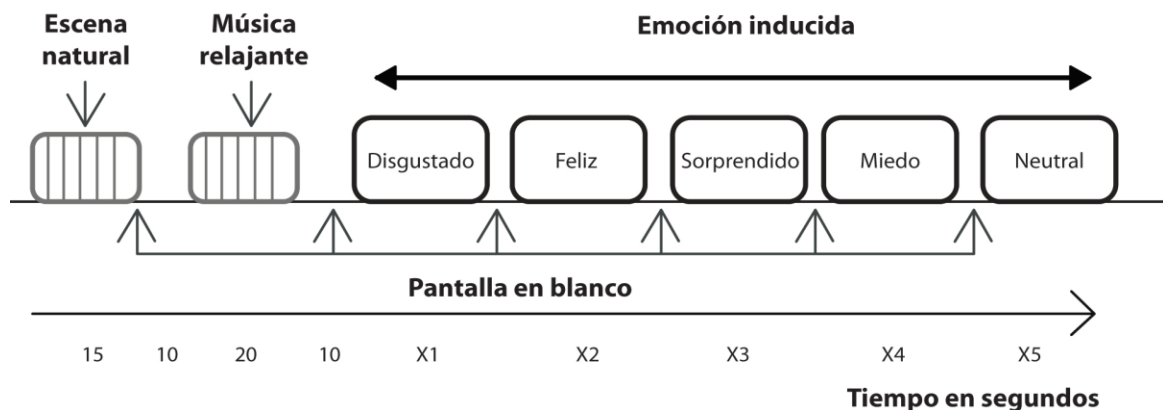


(Horlings, 2008)

2. **Estímulo.** Existe una gran cantidad de métodos para inducir un estado emocional. Entre ellos destacan dos bases de datos ampliamente utilizadas en investigaciones en esta rama, los cuales se conocen como: IAPS (International Affective Picture System) (Bradley & Lang, 2005) y la IADS (International Digitized Sound System) (Bradley M. a., 2007). Estas bases de datos generalmente vienen acompañadas por evaluaciones emocionales, diseñadas bajo el criterio promedio de una gran cantidad de personas.

El procedimiento con el cual se expone al sujeto ante los estímulos, consiste de una serie de etapas o pasos. Primero se necesita poner al sujeto en un estado de calma y relajación, para evitar que cualquier estímulo externo influya en el estado emocional del sujeto. Para ello se puede recurrir a sonidos naturales o música relajante. Luego se debe presentar cada uno de los estímulos en periodos de tiempo específicos alternándolos con estímulos neutros, los cuales pueden ser escenas completamente blancas.

Figura 141. Proceso de adquisición de emociones



(Horlings, 2008)

3. **Aprendizaje de máquina.** El aprendizaje de máquina, conocido popularmente en inglés como Machine Learning, es una rama de la inteligencia artificial que tiene como objetivo crear sistemas capaces de aprender por ellos mismos a partir de un conjunto de datos (data set), sin ser programados de forma explícita. (Moya, 2016) También, puede definirse como un conjunto de métodos que son capaces de detectar patrones en los datos, y después hacer uso de estos patrones descubiertos para realizar predicciones de datos futuros o brindar la posibilidad de realizar una toma de decisiones en condiciones de incertidumbre (Murphy, 2012).

Richiert, la define como una clasificación de algoritmos que tienen como objetivo, enseñar a la máquina la forma correcta de llevar a cabo ciertas tareas, a través del análisis de ejemplos concretos, de qué hacer, para ejecutarla (Richiert, 2016). Los algoritmos de aprendizaje de máquina, se dividen con base en el tipo de aprendizaje que utilizan. Los dos tipos principales son: aprendizaje supervisado y aprendizaje no supervisado.

a. **Aprendizaje supervisado o predictivo.** El objetivo de este aprendizaje es aprender a mapear desde una entrada  $x$  a una salida  $y$ , dado un conjunto de etiquetas de pares de entrada-salida  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ , donde  $\mathcal{D}$  es llamado el conjunto de entrenamiento (training set), y  $N$  es el número de muestras de entrenamiento (Murphy, 2012).

En la configuración más simple, cada entrada de entrenamiento  $X_i$ , es un vector numérico con  $D$  dimensiones. Donde cada uno de los valores del vector son llamados características (features), atributos (attributes) o covariables (covariates). En general,  $X_i$  podría tener una estructura de objetos muy compleja, tales como una imagen, una oración, un mensaje de correo, un conjunto de datos, una forma molecular, un grafo, etc (Murphy, 2012).

De la misma manera, la forma en que se presenta la salida o la variable de respuesta puede ser cualquier cosa, pero la mayoría de los métodos suponen que  $Y_i$  es una variable categórica o nominal de algún conjunto finito de etiquetas. En este caso el problema es conocido como clasificación o reconocimiento de patrones. En cambio, cuando la variable de respuesta representa un valor escalar (como un nivel de ingresos), el problema se conoce como regresión (Murphy, 2012).

b. **Aprendizaje no supervisado o descriptivo.** En este tipo de aprendizaje solo se tiene una entrada  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ , y el objetivo es encontrar patrones interesantes en los datos. Esto a veces es llamado descubrimiento de conocimiento. En otras palabras, no se tiene al alcance un conjunto de datos de entrenamiento inicial (Murphy, 2012).

4. **Métodos de evaluación de un sistema.** Para evaluar la eficacia de un algoritmo de aprendizaje de máquina, es necesario disponer de un conjunto de datos, ya sea etiquetados o no, para entrenar y luego validar dicho algoritmo. Lo cual es un proceso clave, debido a que debe buscarse la utilización óptima de los datos proporcionados. Un error muy común es la utilización de los mismos datos de entrenamiento para la validación. De esta forma se obtienen un rendimiento muy bueno para el algoritmo, pero no se está comprobando la eficiencia del algoritmo de manera correcta (Moya, 2016).

a. **Partición (Hold out).** Este método divide los datos en dos subconjuntos: uno de entrenamiento y uno de prueba. El problema que tiene este método es que los datos del conjunto de prueba, no son utilizados para el entrenamiento, lo cual reduce considerablemente el entrenamiento del algoritmo (Moya, 2016).

Para este método, se recomienda utilizar un 80% de los datos para entrenamiento y un 20% de los datos para la validación. Es por ello que este método entra en problemas, cuando se tiene un número muy pequeño de datos de entrenamiento (Moya, 2016).

b. **Validación cruzada (Cross Validation).** Un enfoque más sofisticado que usar simplemente un conjunto de entrenamiento y uno de validación, es usar el conjunto completo de datos para entrenar y validar un algoritmo dado. A este método se le llama Validación cruzada (Brownlee, 2016).

Básicamente consiste en dividir el conjunto de datos en un determinado número de grupos de igual tamaño denominados folds. Entonces, el algoritmo es entrenado a partir de todos los folds exceptuando uno, el cual es utilizado para la validación. Este proceso es repetido hasta utilizar todos los grupos de datos (folds), como datos de validación. Al final la exactitud del algoritmo es obtenida a partir del promedio de todas las exactitudes obtenidas en cada una de las iteraciones realizadas (Brownlee, 2016).

5. **Señales electroencefalográficas (EEG).** Las señales electroencefalográficas, abreviadas como EEG por sus siglas en inglés. Son fluctuaciones de voltaje que ocurren en el cerebro, las cuales juegan un papel muy importante en la detección de emociones. Se cree que las emociones están relacionadas con un grupo de estructuras en el centro del cerebro llamado sistema límbico, el cual incluye: amígdala cerebral, tálamo, hipotálamo y el hipocampo (Noppadon, 2013).

En el mercado existe una gran cantidad de dispositivos para la medición de señales EEG, en donde destacan: Emotiv EPOC, Neuro Sky, MyndPlay BrainBand. Cada uno de ellos proporcionan un conjunto de características interesantes, pero todas ellas son herramientas comerciales y sus precios rondan entre \$200 y \$400.

6. **Pre-procesamiento de una señal.** Antes de la clasificación de la señal, siempre existe un pre-procesamiento de la misma. En esta etapa se elimina cualquier ruido existente en la señal, que no haya sido producto de la actividad neuronal en el cerebro. A estos agentes que causan este tipo de ruido, se les conoce como artefactos y son generalmente provocados por parpadeos, efectos musculares y vasculares y por brillo cinético. Además, es en esta etapa, cuando se realiza la extracción de características. La herramienta más utilizada para este propósito es la transformada de Wavelets (González E. , 2011).

7. **Extracción de características.** Se le conoce así, al proceso en el cual se obtienen características representativas de una señal, durante un periodo de tiempo específico, denominado ventana o segmento de aquí en adelante. Existen una gran cantidad de enfoques distintos para la extracción de dichas características. La característica ampliamente utilizada para la detección o clasificación de emociones es PSD (Power Spectrum Density), que básicamente consiste en determinar el poder de una señal EEG, en un rango de frecuencias específicas. La anterior, se ha popularizado tanto, debido al poco costo computacional que conlleva, lo cual lo hace ideal para un ambiente de reconocimiento de emociones en tiempo real (Noppadon, 2013).

La señal EEG, suele dividirse en rangos de frecuencias muy específicos, debido a que es en esta división, donde realmente se pueden determinar las características básicas presentes en la señal (Noppadon, 2013). Dichos rangos son descritos en la siguiente Cuadro:

Cuadro 4. Rangos de frecuencias que componene las señales EEG

<b>Nombre</b>	<b>Frecuencia (Hz)</b>
Delta	0-4
Theta	4-8
Alpha	8-16
Beta	16-32
Gamma	32-64

(Noppadon, 2013)

Las bandas que se encuentran estrechamente relacionadas con el estado emocional de una persona, son únicamente las Alpha y las Beta (Ramírez, 2008). Algunas investigaciones todavía toman en cuenta a la banda Gamma, para este propósito, pero en general, las primeras dos son las más utilizadas.

Recientes investigaciones señalan que los electrodos AF3, AF4, F3 y F4, que se ubican en el córtex prefrontal del cerebro, son comúnmente utilizados en la detección de emociones. Debido a que es en ellos, donde se muestra un aumento dominante de ondas Alpha y ondas Beta, en estados de excitación, alerta y

relajación. Además, están ubicados en el lóbulo prefrontal, el cual juega un papel crucial en la regulación de emociones y durante estados conscientes (Ramírez, 2008).

8. **Transformada de Wavelets.** Wavelet es una onda limitada durante un tiempo, que posee un valor promedio de cero. Es buena para describir anomalías, pulsos y otros eventos que empiezan y terminan dentro de una señal. Permite realizar análisis tomando en cuenta el tiempo, de una gran señal. Lo cual brinda la posibilidad de encontrar discontinuidades o picos de corta duración que de otra manera sería complicado detectar y tratar (Cortés, 2007).

La transformada de Wavelets es muy similar a la transformada de Fourier, en la cual se utiliza el concepto de aproximación de señales usando la superposición de otras de ellas. El análisis utilizando transformada de Fourier consiste en descomponer la señal original en señales sinusoidales de diferente frecuencia. De forma similar, la transformada Wavelet descompone la señal en versiones trasladadas en el tiempo y escaladas de la Wavelet original, conocida como Wavelet Madre (González E. , 2011).

La transformada de Wavelet, combina las dimensiones tiempo y frecuencia, permitiendo la extracción de los detalles y de las aproximaciones de la señal. Lo cual no puede ser obtenido aplicando ninguna de las versiones de la transformada de Fourier, lo cual la hace muy importante en el procesamiento de señales electroencefalográficas (Mallat, 1989).

9. **Algoritmo de máquina de soporte vectorial.** En el tema de clasificación, las máquinas de vectores de soporte, son algoritmos que se han popularizado en los últimos años. El algoritmo se le conoce como SVM, por sus siglas en inglés (Support Vector Machine). Se basa principalmente en establecer un hiperplano que separa de forma óptima los subconjuntos de puntos correspondientes a cada una de las clases. Para ello utiliza los vectores de características, que fueron descritos en el apartado anterior. Los cuales han sido proyectados previamente en un espacio de dimensionalidad superior. Es capaz de crear el modelo, a partir de un conjunto de datos de entrenamiento. Lo cual lo convierte en una técnica con una gran eficiencia computacional (Sherwood, 2009).

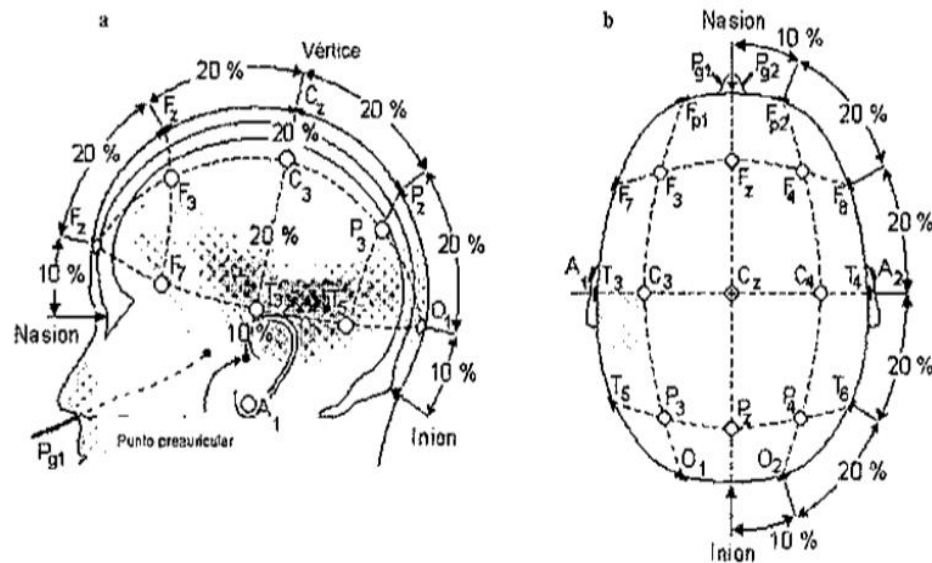
El concepto detrás del algoritmo de las máquinas de vectores de soporte, es maximizar el discriminante marginal, lo cual se logra al proponer una solución utilizando optimización cuadrática por métodos de Lagrange (Sherwood, 2009).

En algunas investigaciones previas, se ha podido comparar el algoritmo de SVM, con otras técnicas de clasificación, llegando a la conclusión que este tiene una mejor precisión al clasificar. Pero lo más importante de todo, es que SVM se considera un clasificador binario, lo cual lo hace muy útil, en escenarios donde se quieren clasificar únicamente dos clases (Kotsiantis, 2007).

10. **Electrodos.** Para ilustrar la complejidad del cerebro, hay que mencionar que un solo milímetro cúbico contiene al menos 40,000 neuronas y 1,000 millones de conexiones de fibras nerviosas. Cada neurona y conexión genera campos eléctricos, que pueden llegar a cancelarse o sumarse dependiendo de su posición (Thevenet, 2008). Por eso los neuropsicólogos han desarrollado sistemas de posicionamiento de electrodos, que buscan obtener lecturas óptimas de dichas señales.

11. **Sistema internacional 10-20.** En este sistema cada localización viene definida por dos coordenadas: una, su proximidad a una región concreta del cerebro y, otra, su ubicación en el plano lateral (Núñez Peña, 2004). La proximidad a la región concreta del cerebro está definida en los lóbulos que encontramos, y la del plano lateral está determinado por números impares a la izquierda, números pares a la derecha, y la letra Z para los centrales.

Figura 142. Sistema Internacional 10-20. Distribución de 10s electrodos sobre la superficie de la cabeza con las posiciones básicas obtenidas a partir del 10% y 20% de la medida total del cráneo.



(Núñez Peña, 2004)

12. **Kit de desarrollo de software (SDK).** Se le conoce como SDK, por sus siglas en inglés (Software Development Kit), es un conjunto de herramientas que permiten a los programadores desarrollar aplicaciones de software, para una plataforma o lenguaje específico. El SDK, la mayoría de las veces es proporcionado por empresas de hardware o software para ayudar a los desarrolladores en el proceso de creación de nuevas aplicaciones para productos existentes (Morley, 2016).

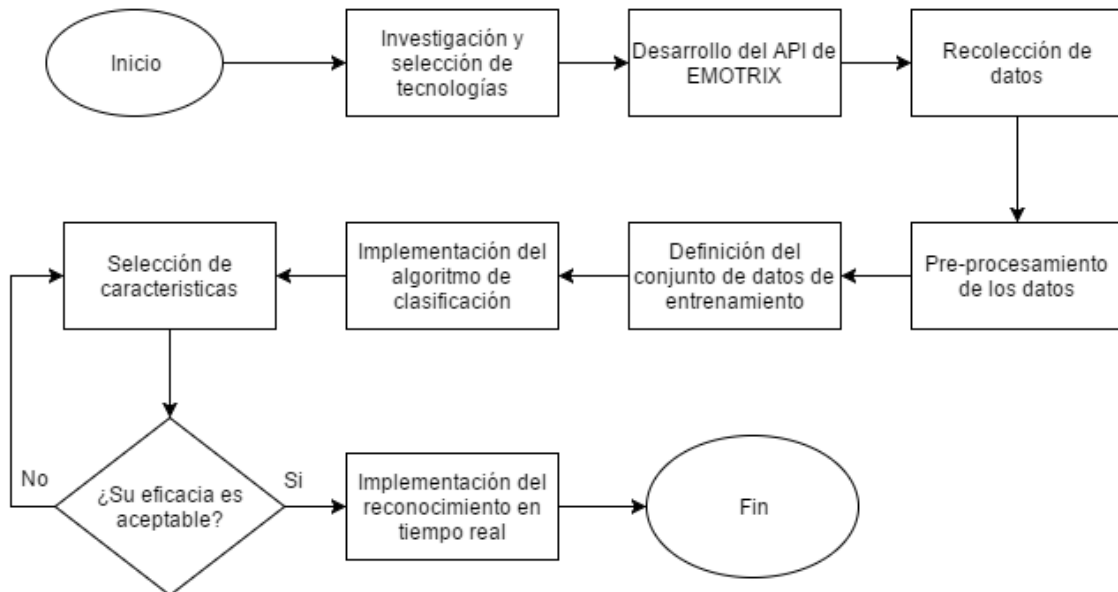
Un SDK abarca muchos componentes, pero los principales son: Herramientas de programación, documentación, códigos de ejemplo y como mínimo un API (Application Program Interface) (Morley, 2016).

13. **API.** Application Program Interface (API) es un conjunto de rutinas, protocolos y herramienta para el desarrollo de aplicaciones de software (Morley, 2016).

## B. Marco metodológico

En cuanto a la metodología utilizada para el desarrollo del presente módulo, se decidió utilizar una serie de pasos, los cuales se describen en el diagrama de la Figura 4.

Figura 143. Metodología de trabajo representada en un diagrama de flujo.



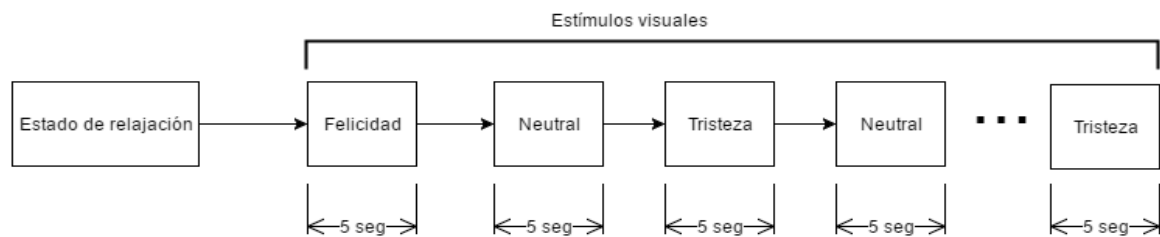
1. **Investigación y selección de tecnologías.** El primer paso en la metodología fue la investigación exhaustiva de estudios similares, para tener una base fundamentada y concreta sobre el trabajo a realizar. Esta etapa fue de vital importancia, debido al poco conocimiento previo que se tenía, sobre este campo al iniciar el proyecto. Gran parte de la teoría leída se centró en el funcionamiento del cerebro y como surgen las emociones.

2. **Desarrollo del API de Emotrix.** En cuanto al desarrollo de las funciones del API de EMOTRIX, se utilizó como lenguaje Python. El ambiente de desarrollo, fue construido utilizando la herramienta Pycharm, el cual nos provee todas las herramientas necesarias para un proyecto de dicha escala. Las funciones del API de EMOTRIX, fueron desarrolladas en dos categorías distintas: Funciones para conectividad con dispositivos de hardware y las funciones para la detección de emociones. El presente módulo se enfocó únicamente en el desarrollo del segundo grupo de funciones.

Además del desarrollo de las funciones, también se realizó toda la documentación relacionada a cada una de ellas, en la cual se indica toda la información relevante, para su correcta implementación. Toda esta documentación fue desarrollada en un documento con formato pdf, el cual estará adjunto en el SDK de EMOTRIX.

3. **Toma de datos.** En esta etapa se expuso a 20 sujetos, a una prueba que consistía en 20 estímulos visuales. De los 20 estímulos visuales, 10 estaban destinados para provocar felicidad y los otros 10 para provocar tristeza. Los estímulos, fueron presentados de forma alterna e intercalando un estímulo neutral entre cada uno de ellos. El tiempo de exposición de cada estímulo fue de 5 segundos. Dicho procedimiento se detalla a profundidad en la Figura 144.

Figura 144. Procedimiento para la exposición de los estímulos visuales



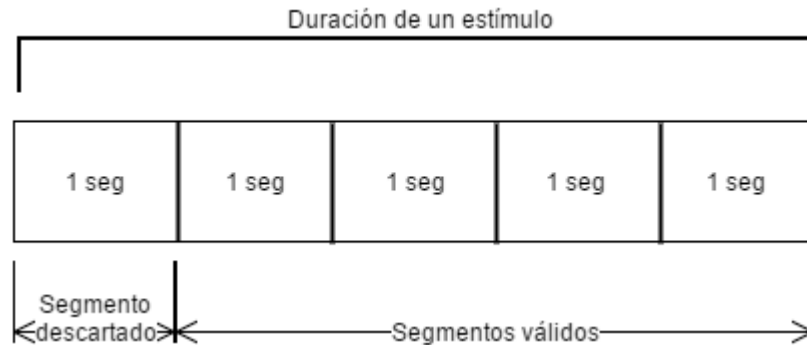
Para la lectura de las señales EEG de los sujetos evaluados se utilizó el dispositivo Emotiv EPOC, el cual nos provee un API para poder obtener la data en crudo de cada uno de los electrodos que lo componen.

Para los propósitos de este módulo, se utilizó únicamente la lectura de los electrodos F3, F4, AF3 y AF4.

Los estímulos fueron obtenidos, luego de ser evaluados por 92 personas utilizando una encuesta en línea. Se les solicitaba a los participantes evaluar cada una de las imágenes en una escala de 0 a 10, donde 0 indicaba felicidad y 10 tristeza. Los resultados fueron tabulados y luego se extrajeron las 10 imágenes más representativas para cada emoción.

4. **Pre-procesamiento de los datos.** Durante esta etapa se prepararon los datos para ser utilizados como entrada para el algoritmo de SVM. En primera instancia, este pre-procesamiento incluyó la división, en segmentos, de los datos obtenidos para cada uno de los sujetos evaluados. Estos segmentos tienen una duración de un segundo. Se debe mencionar que el primer segmento fue descartado, debido a que este podría contener efectos causados por el estímulo anterior. Por lo tanto, se extrajeron únicamente cuatro segmentos válidos por cada estímulo presentado. Véase Figura 145.

Figura 145. División de cada bloque de datos en segmentos de 1 segundo.



El segundo paso fue la implementación de la transformada de Wavelet. Dicha transformada es capaz de descomponer la señal en distintos conjuntos de coeficientes de aproximación y de detalle, que representan las distintas bandas de frecuencia que componen la señal. Esta descomposición fue realizada, sobre cada uno de los segmentos resultantes del paso anterior. La razón principal por la cual se utiliza este tipo de transformada y no la de Fourier, es debido a la naturaleza de las señales EEG. El análisis de Fourier es empleado para funciones estacionarias, debido a que pierde la dimensión del tiempo al pasarse al dominio de la frecuencia. Mientras que las señales EEG se comportan como señales no estacionarias, es por eso que el mejor candidato para este análisis es la transformada de Wavelet, donde se puede extraer las frecuencias de la señal sin perder la dimensión del tiempo.

5. **Definición del conjunto de datos de entrenamiento.** Al implementar un algoritmo de machine learning que utiliza aprendizaje supervisado, fue necesario definir un conjunto de datos de entrenamiento. Tomando en cuenta, que la lectura de las señales EEG, fue realizada sobre 20 sujetos. Los cuales fueron expuestos a 10 estímulos de tristeza, 10 de felicidad y un estímulo neutro entre cada uno de los anteriores, hacen un total de 40 estímulos por sujeto. Además, cada estímulo fue presentado durante 5 segundos, pero se extrajeron solo 4 segmentos válidos para cada uno. Por lo tanto, se obtuvo un total de 3,200 segmentos etiquetados, de los cuales 800 fueron para felicidad, 800 para tristeza y 1600 para un estado neutral. Utilizando ese conjunto de datos etiquetados e implementando cross validation, con diferentes subconjuntos, se obtuvo los datos de entrenamiento para el algoritmo.

6. **Implementación del algoritmo de clasificación.** Como algoritmo de clasificación, se utilizó SVM. Su implementación se llevó a cabo mediante la librería Scikit-learn, la cual nos provee una amplia gama de herramientas y métodos para el procesamiento de datos utilizando Python.

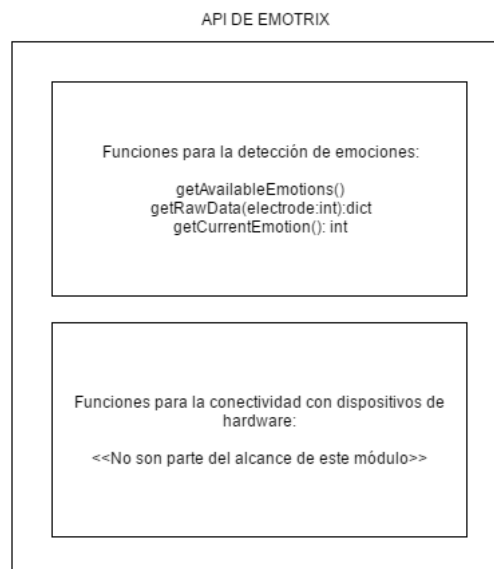
7. **Proceso de selección de características.** La selección de características es una de las etapas más críticas al implementar un algoritmo de aprendizaje de máquinas. Es por ello que gran parte del tiempo invertido en el módulo, fue enfocado en el trabajo realizado en esta sección. Para el reconocimiento de emociones se utilizó dos tipos distintos de características: Características en el dominio del tiempo y características en el dominio de la frecuencia. Las primeras fueron extraídas directamente de la señal original. Las segundas fueron extraídas únicamente de las bandas de interés: alpha, beta y gamma. Lo cual, fue posible, gracias a la descomposición de la señal utilizando la transformada de Wavelet. Pero, con el objetivo de hacer más evidente el peso que tiene la incidencia de ciertas frecuencias en la señal, se utilizó la densidad espectral de potencia (PSD) sobre el conjunto de coeficientes obtenidos, en las bandas mencionadas anteriormente.

8. **Implementación del reconocimiento en tiempo real.** Por último, se llevó a cabo la implementación del algoritmo, para el reconocimiento de emociones en tiempo real. Para ello se implementó un buffer, el cual nos permite mantener en memoria, las lecturas de las señales EEG que ocurrieron en una ventana de tiempo, desde el instante actual. El tamaño es configurable, brindando la capacidad de adecuarse a las necesidades del algoritmo implementado. El tamaño del buffer o tamaño de la ventana es básicamente la cantidad de datos que se están utilizando para realizar la detección de emociones.

## C. Resultados

1. **Estructura del API de Emotrix.** En la Figura 146, se describe la estructura resultante para el API de Emotrix. Como se mencionó anteriormente el API está dividido en dos partes principales: El grupo de funciones enfocadas en la detección de emociones y el grupo de funciones encargadas de todo lo relacionado con la conectividad con los dispositivos de hardware de Emotrix.

Figura 146. Estructura del API de Emotrix



2. Pre-procesamiento de los datos. Antes del proceso de clasificación, las señales EEG fueron pre-procesadas con dos objetivos principales: eliminar la mayor cantidad de ruido posible y el segundo, extraer únicamente los componentes que contengan información importante para la detección de emociones. A continuación, se presentan las transformaciones aplicadas a las señales:

Figura 147. Señales EEG para los electrodos F3, F4, AF3, AF4 del sujeto #5.

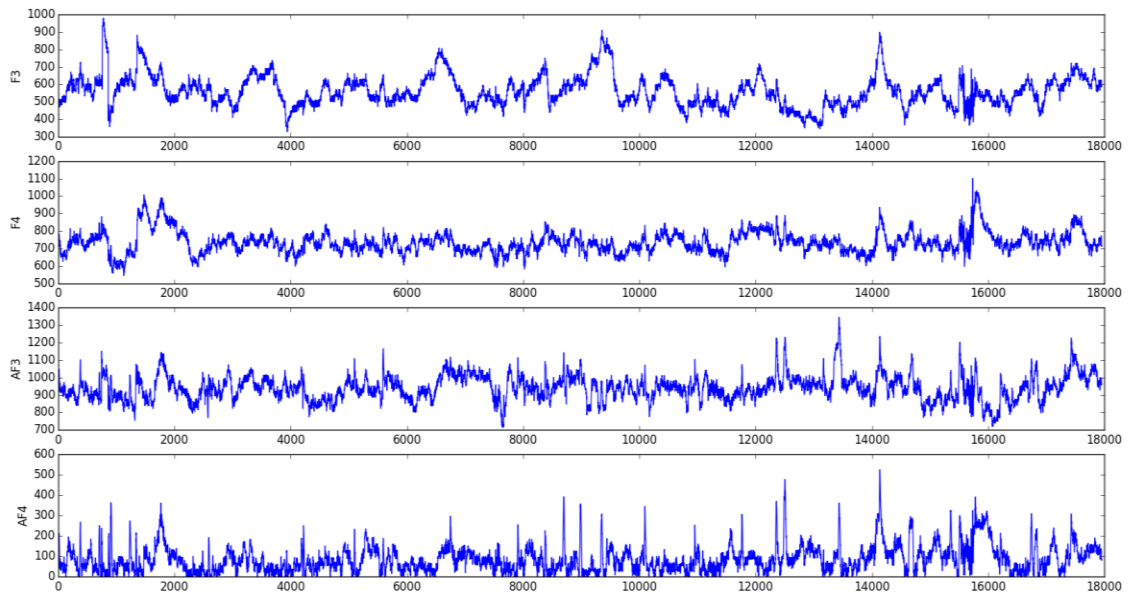


Figura 148. Señal EEG contenida en un segmento de 1 segundo de duración (128 muestras).

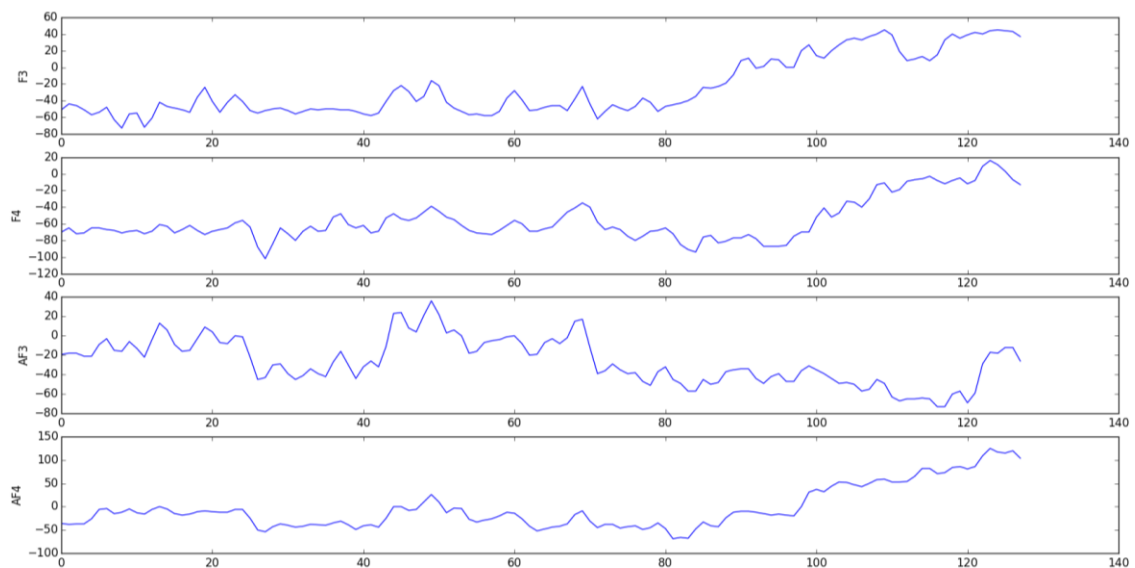


Figura 149. Comparación de las señales EEG, para los electrodos F3, F4, AF3, AF4. Segmento etiquetado como tristeza=rojo, alegría = azul y neutral = negro.

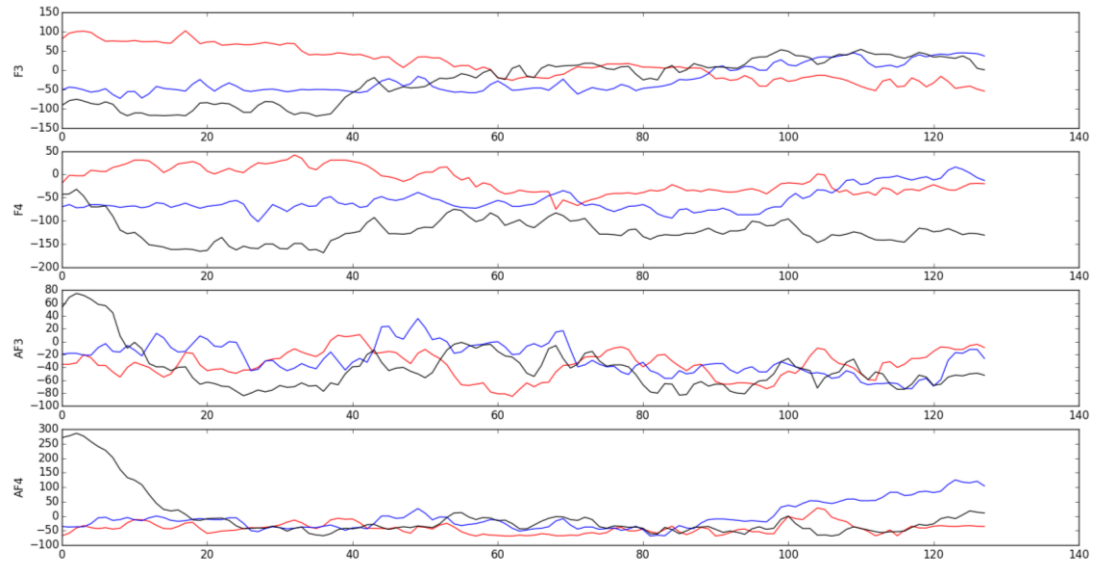


Figura 150. Aplicación de la transformada de Wavelet sobre el electrodo AF4, con 4 niveles de descomposición, utilizando una wavelet madre Daubechies 4 (db4), para separar las distintas bandas de frecuencia de las señales EEG.

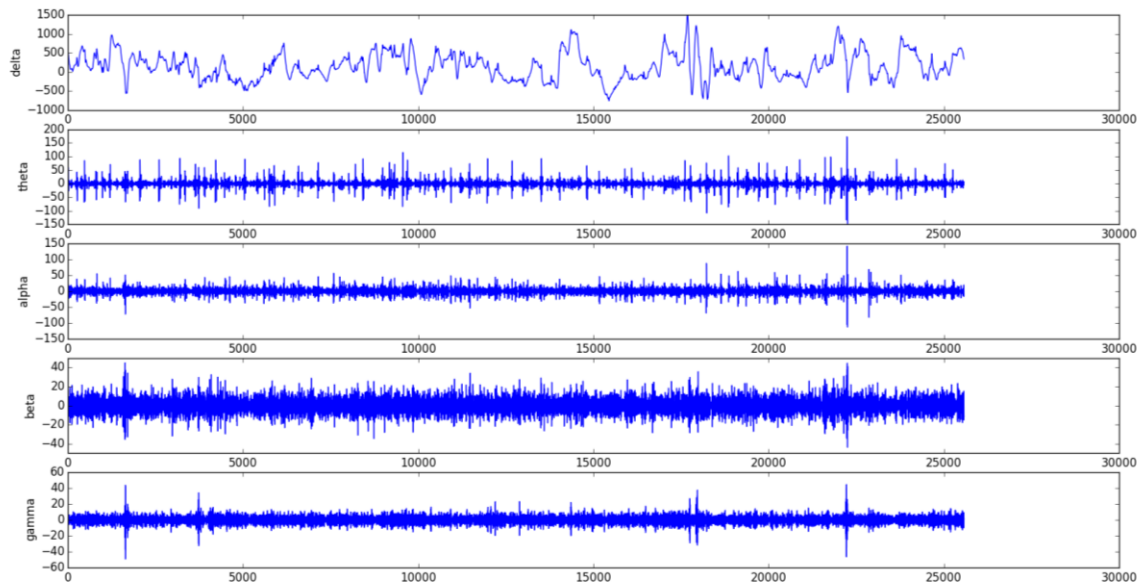
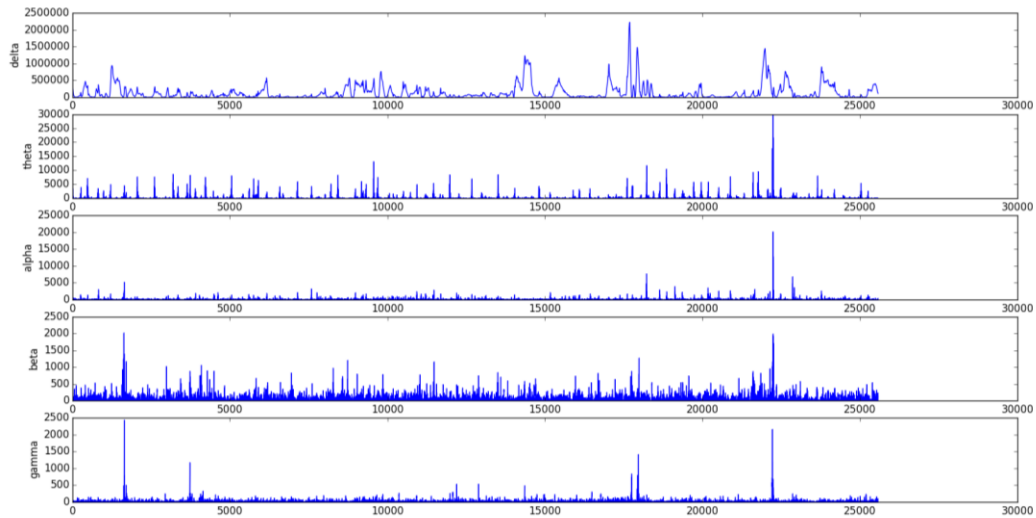


Figura 151. Obtención de la densidad espectral de potencia, luego de haber sido aplicada la transformada Wavelet en el electrodo AF4, para el sujeto #5.



3. **Eficacia del algoritmo.** La eficacia del algoritmo fue obtenida al medir la cantidad de aciertos y fallos que obtiene el algoritmo de SVM, con el conjunto de datos de prueba definido. La variación de distintos parámetros puede aumentar o disminuir el porcentaje de eficacia del algoritmo. A continuación, se muestran las variaciones realizadas:

a. **Variación de las características utilizadas**

Cuadro 1. Comparación de la eficacia del algoritmo, al implementar PSD

Clases (estados emocionales)			Aplicación de PSD.	Eficacia del algoritmo
Felicidad	Tristeza	Neutral		
X		X	no	77%
X		X	si	78%
	X	X	no	71%
	X	X	si	77%
X	X		no	68%
X	X		si	74%
X	X	X	no	58%
X	X	X	si	62%

b. **Variación del tamaño de la ventana**

Cuadro 5. Eficacia del clasificador SVM, con tamaños de ventana de 128 (1 seg) y 256 (2 seg), sin traslape.

Clases (estados emocionales)			Tamaño de la ventana	Eficacia del algoritmo
Felicidad	Tristeza	Neutral		
X		X	128	74%
X		X	256	81%
	X	X	128	73%
	X	X	256	80%
X	X		128	59%
X	X		256	51%
X	X	X	128	51%
X	X	X	256	52%

Cuadro 6. Eficacia del clasificador SVM, con tamaños de ventana de 128 (1 seg) y 256 (2 seg), con traslape.

Clases (estados emocionales)			Tamaño de la ventana	Eficacia del algoritmo
Felicidad	Tristeza	Neutral		
X		X	128	70%
X		X	256	77%
	X	X	128	69%
	X	X	256	75%
X	X		128	63%
X	X		256	71%
X	X	X	128	55%
X	X	X	256	61%

### c. Variación del límite de calidad permitida

Cuadro 7. Eficacia del clasificador SVM, sin un límite de calidad permitido.

Clases (estados emocionales)			Tamaño de la ventana	Número de segmentos			Eficacia del algoritmo
Felicidad	Tristeza	Neutral		Felicidad	Tristeza	Neutral	
X		X	256	5000	0	11400	70%
	X	X	256	0	5000	11400	69%
X	X		256	5000	5000	0	54%
X	X	X	256	5000	5000	10000	50%

Cuadro 8. Eficacia del clasificador SVM, con un límite de calidad de la lectura mayor a 50%

Clases (estados emocionales)			Tamaño de la ventana	Número de segmentos			Eficacia del algoritmo
Felicidad	Tristeza	Neutral		Felicidad	Tristeza	Neutral	
X		X	256	3768	0	8482	71%
	X	X	256	0	3675	8482	72%
X	X		256	3775	3675	0	57%
X	X	X	256	3768	3675	7439	53%

Cuadro 9. Eficacia del clasificador SVM, con un límite de calidad de la lectura mayor a 70%

Clases (estados emocionales)			Tamaño de la ventana	Número de segmentos			Eficacia del algoritmo
Felicidad	Tristeza	Neutral		Felicidad	Tristeza	Neutral	
X		X	256	436	0	1051	77%
	X	X	256	0	456	1051	75%
X	X		256	436	456	0	71%
X	X	X	256	436	456	918	61%

## D. Discusión

En cuanto a la estructura del API de Emotrix, se propuso una estructura bastante simple. Se optó por la separación de las funciones en dos grupos: Funciones para la detección de emociones y las funciones para la conectividad con dispositivos de hardware. La razón principal de tal separación, fue básicamente la modularización del API. Con ello, los dos módulos podían ser realizados de forma independiente y de manera simultánea.

El hecho de haber utilizado únicamente los electrodos F3, F4, AF3 y AF4 es debido al aumento de ondas Alpha y Beta que ocurre en esta área del cerebro, durante estados de alerta, excitación y relajación. (Ramírez, 2008). En la Figura 8, se muestra el comportamiento de las señales EEG leídas durante un período total de 40 segundos, para cada uno de los electrodos mencionados anteriormente. Observando detalladamente la imagen, puede notarse que no existe un patrón obvio del comportamiento de las señales EEG, cuando el sujeto estuvo expuesto a los diferentes estímulos (felicidad, tristeza y neutral).

Como se mencionó anteriormente, los datos fueron divididos en segmentos. La razón de tal segmentación fue debido a que es muy complicado y carece de sentido, determinar una emoción basado en un solo dato. La Figura 9 muestra el tamaño del segmento (o tamaño de la ventana), utilizado en primera instancia para realizar el análisis. La longitud del segmento fue seleccionada con base en la cantidad de muestras que pueden obtenerse en un segundo de lectura. Para el caso de Emotiv son 128 muestras por segundo, debido a que utiliza una tasa de muestreo de 128 Hz. La Figura 10, presenta una breve comparación entre segmentos de datos obtenidos con diferentes estímulos, los cuales fueron extraídos de manera aleatoria. Nuevamente no puede detectarse un patrón claro y definido en cada uno de ellos. Es por ello, que pretender realizar un análisis para el reconocimiento de emociones, utilizando únicamente esta información no tiene mucho sentido.

La Figura 11, muestra el resultado de la aplicación de la transformada Wavelet sobre las señales EEG obtenidas para un sujeto en específico. En ella se puede visualizar la separación de las cinco bandas de frecuencias que componen la señal: Delta, theta, alpha, beta y gamma. Algo bastante claro de identificar en la imagen, es la incidencia mayoritaria de las ondas tipo alpha, delta y gamma. Con ello se corrobora lo encontrado en la literatura con respecto a la relación que tienen dichas bandas con las emociones (Ramírez, 2008).

Como se mencionó en la metodología, la aplicación de la densidad espectral de potencia, hace más evidente el peso que tienen ciertas frecuencias en la señal. Dicho resultado se representa en la Figura 12.

Para la descomposición de la señal utilizando la transformada Wavelet, se utilizó una Daubieches 4 (db4) como función Wavelet madre, con cuatro niveles de descomposición. Su elección, principalmente está basada por lo encontrado en la literatura (Noppadon, 2013). Es necesario mencionar, que se evaluó, utilizando otras dos funciones de la misma familia: db2 y db6 sin mostrar resultados significativos en la eficacia del algoritmo.

La eficacia del algoritmo fue comparada, mediante la modificación de diversos parámetros, entre ellos: Características de los datos, tamaño de los segmentos, su traslape y la calidad de la señal. El Cuadro 2, compara los resultados obtenidos cuando se aplicó la densidad espectral de potencia y cuando no. Resulta evidente la mejora alcanzada por el algoritmo, al implementar PSD, mejorando en promedio un 6% su desempeño; alcanzando 78% para Felicidad-Neutral, 77% Tristeza-Neutral, 74% Felicidad-Tristeza y 62% para Felicidad-Tristeza-Neutral. Su justificación, PSD hace más evidente la incidencia de las frecuencias dentro de la señal, debido a que eleva al cuadrado cada uno de los coeficientes. Por ello hace más significativas las diferencias en cuanto a las frecuencias para cada una de las clases.

El segundo parámetro a evaluar fue el tamaño de la ventana o del segmento utilizado. Dicha comparación se resume en el Cuadro 3. Los tamaños de ventana comparados fueron de 128 y 256, lo que equivale a 1 y 2 segundos de lectura respectivamente. Se puede observar un incremento significativo en la eficiencia del algoritmo en casi todos los escenarios: del 74% a 81% para Felicidad-Neutral; del 73% al 80% para Tristeza-Neutral y del 51% al 52% para Felicidad-Tristeza-Neutral. Con excepción de cuando se clasificó únicamente tristeza y felicidad dejando a un lado el estado neutral (disminuyó del 59% al 51%). Una de las causas por las cuales el algoritmo disminuyó su eficacia, en este escenario, puede ser debido a la escasa cantidad de muestras que se utilizaron, debido a que son las clases con menores muestras. Y es que tiene sentido, si se habla que se está duplicando el tamaño de la ventana, obviamente el tamaño de las muestras se reduce a la mitad y si se trabaja con pequeños conjuntos de datos, puede afectar el desempeño del algoritmo.

Una de las soluciones para evitar la disminución exagerada de muestras, cuando se duplica el tamaño de la ventana, fue crear segmentos que no fuesen mutuamente excluyentes. Lo que significa, que ahora los segmentos contiguos comparten muestras, ósea que poseen una zona de traslape. Por lo tanto, se realizó la misma comparación anterior, pero ahora utilizando traslape entre las muestras. Los resultados están descritos en el Cuadro 4. En esta combinación de parámetros ocurre algo interesante, los primeros dos escenarios (alegría-neutral y tristeza-neutral) disminuyen la eficacia comparados con la combinación anterior; Mientras que los últimos dos (alegría-tristeza y alegría-tristeza-neutral) logran una mejora significativa. La razón, puede ser debido a la cantidad desigual de muestras que involucran cada uno de los escenarios. En cuanto a los primeros dos escenarios, la proporción de muestras es de 1 a 2 (1 para alegría/tristeza y 2 para neutral). Mientras que, para los otros dos escenarios, la proporción de muestras es más equilibrado. Por lo tanto, duplicar el número de muestras utilizando esta idea, no crea diferencias más grandes, comparado con el número de muestras originales y por ello, su rendimiento no se ve afectado.

El siguiente parámetro, para mejorar la eficacia de clasificación del algoritmo, fue la calidad de la lectura. En primera instancia no se utilizó ningún tipo de filtro para la calidad de la lectura obtenida, por ello se utilizaron todas las muestras recopiladas. Se debe aclarar que se siguió utilizando una ventana de 256 muestras con traslape. Los resultados están descritos en el Cuadro 5, donde se observa un número elevado de muestras utilizadas: 5000 para tristeza, 5000 para alegría y 10000 para neutral, pero una eficacia de

clasificación para el algoritmo bastante baja. Obteniendo solo un 54% de clasificación exitosa para las dos emociones: alegría y tristeza. En la segunda corrida se decidió utilizar un filtro, que solo permitiese utilizar aquellas lecturas con una calidad mayor al 50%. Los resultados se resumen en el Cuadro 6, si se observa detenidamente se puede observar la reducción en el número de muestras, debido a la implementación del filtro. Dicho filtro, provocó la reducción en un 30% de las muestras originales, lo cual es un porcentaje bastante elevado si se tienen pocas muestras originales. Esta disminución de las muestras se ve compensada con un aumento en la eficacia del algoritmo, comparada con la corrida anterior en aproximadamente un 3%. Lo cual se considera un aumento considerable, debido a que ocurrió en todos los escenarios. Por último, se realizó el mismo experimento, pero a un nivel mayor, al implementar un filtro que solo permitiese la utilización de aquellas lecturas con una calidad mayor al 70%. Con este filtro, la muestra se redujo en un 91.5%, quedando entonces, solo un 9.05% de la muestra original. Los resultados para esta corrida están descritos en el Cuadro 7. Lo sorprendente de la implementación de dicho filtro fue la mejora considerable del desempeño del algoritmo de clasificación, en promedio aumentó un 7.75% de eficacia tomando en cuenta todos los escenarios.

## E. Conclusiones

1. La separación de las funciones del API de Emotrix en dos grupos: funciones para la conectividad con dispositivos de hardware y funciones para la detección de emociones, permitió la modularización del mismo. Con ello, se pudo trabajar de manera simultánea e independiente los dos módulos, sin ningún inconveniente.
2. La mejor clasificación, para el escenario cuando se comparó: tristeza y felicidad fue de: 74% de eficacia; lo cual se logró utilizando la densidad espectral de potencia (PSD) en las características, una ventana con traslape de 256 muestras (2 segundos), y un filtro que permitiese solo la utilización de aquellas muestras con una calidad de lectura mayor al 70%.
3. La mejor clasificación, para los escenarios cuando se comparó: felicidad vs neutral y tristeza vs neutral fue de: 81% y 80% de eficacia respectivamente; lo cual se logró utilizando la densidad espectral de potencia (PSD) en las características, el tamaño de la ventana de 256 muestras (2 segundos) pero sin traslape. Y el filtro para la calidad de muestras mayor al 70%.
4. La implementación de la densidad espectral de potencia (PSD), al resultado de la aplicación de la transformada de Wavelet, mejoró significativamente el porcentaje de eficacia del algoritmo de clasificación. El aumento fue, en promedio, del 6% en todos los escenarios.
5. La implementación de una ventana con 256 muestras (2 segundos), representó una mejora considerable en la eficacia del algoritmo, a excepción de aquellos escenarios donde el conjunto de muestras se volvió demasiado pequeño.

6. Filtrar la calidad de las señales utilizadas, fue clave para lograr un aumento en el porcentaje de eficacia del 7.75%. Aunque ello también representó la disminución del conjunto de muestras en un 90% del tamaño original.
7. La utilización de una ventana con traslape, puede ser útil, en aquellos escenarios donde se encuentra con la problemática de poseer un conjunto de muestras bastante limitado.
8. La creación de la documentación relacionada al API de Emotrix, permite al desarrollador entender de una mejor manera el comportamiento de cada una de las funciones que lo componen. Al mismo tiempo que complementa las partes que conforman el SDK completo de Emotrix.

## F. Recomendaciones

1. Todo el desarrollo del presente módulo fue utilizando únicamente el algoritmo de máquina de vectores de soporte (SVM). Para futuros trabajos, se recomienda la implementación de otro tipo de algoritmo de aprendizaje de máquina como complemento a las SVM, para determinar si se logra un mejor porcentaje de eficacia o no, en el proceso de clasificación.
2. Se recomienda utilizar estímulos estandarizados y validados por una mayor cantidad de personas, como por ejemplo el IAPS (International Affective Picture System) y el IADS (International Affective Digitized Sounds) ampliamente utilizados en este tipo de investigaciones.
3. El algoritmo de SVM, tiene una mayor eficacia con base en la calidad de las características seleccionadas para el proceso de clasificación. Por lo tanto, se recomienda utilizar otro tipo de datos, que complementarían la información relacionada a cada una de las emociones. Estos datos pueden ser: Ritmo cardíaco, sudoración, reconocimiento de micro expresiones y tipo de respiración.
4. Para el trabajo realizado en este módulo, se utilizó únicamente la información proporcionada por 4 electrodos distintos: F3, F4, AF3 y AF4, cuando existen otros 10, si se utiliza el dispositivo Emotiv EPOC. Para futuros trabajos, se recomienda la utilización de estos electrodos, para poder comparar cuáles de ellos proporcionan información más significativa en la detección de emociones.

## VIII. MÓDULO 5: DISEÑO Y DESARROLLO DE LA PLATAFORMA FIGURA PARA EMOTRIX

### A. Marco teórico

1. **Python.** Python es un lenguaje de programación orientado a objetos, sencillo, pero de gran potencia, que permite el desarrollo rápido de aplicaciones. Es un lenguaje interpretado y altamente modular, por lo que la etapa de desarrollo se reduce considerablemente y, además, resulta muy sencillo añadir nuevos elementos a posteriori al programa. (Bauer, 1996) (Van Rossum & Drake, 2000).

2. **WxPython.** WxPython es una librería de Python para desarrollo de ambientes gráficos que posean una interfaz enfocada en el usuario, simple y fácil de usar. Es implementada como un módulo de extensión de Python. Esta librería es Open Source, lo que significa que cualquier persona puede la puede utilizar y el código de la misma. Asimismo, WxPython es una herramienta multiplataforma. Esto significa que el mismo programa se puede ejecutar en diferentes plataformas sin hacer ninguna modificación al código. Al tener esta cualidad, esta librería es soportada en sistemas operativos como Windows, cualquier distribución de Linux y Macintosh OS, que sean de 32 bits. (Dunn, 1990)

3. **Aplicación.** El término aplicación proviene del latín “applicatio”, “applicatiōnis” compuesto lexicalmente con el prefijo “ad” equivalente “hacia”, más la voz “plicare” que quiere decir “doblar” o “hacer pliegues” y el sufijo “cion” de acción y efecto; por ende, según su etimología se puede decir que la palabra aplicación hace referencia a la acción y el efecto de aplicar o aplicarse. En el ámbito de la informática, el que corresponde al trabajo, el término Aplicación se puede definir como un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos. (Comunidad, Definición de Aplicación, 2014)

4. **Interfaz.** Según Williams, Sochats y Morse (Williams, Sochats, & Morse, 1995) definen al término Interfaz como “El filtro por el cual unos conjuntos de datos modelados por un sistema informático se presentan al usuario”.

Desde este punto de vista, la interfaz de usuario se puede presentar como un canal de comunicación entre un individuo y una máquina. Debido a esto, y para que se genere una interacción entre individuo-máquina, la interfaz debe ofrecer a cada uno de los dos agentes soluciones a sus necesidades, no sólo ser una pantalla divisoria.

5. ¿Cómo funciona EMOTIV?. El enfoque del dispositivo “Emotiv” es la lectura de las ondas cerebrales por medio de electrodos, filtrando dichas lecturas para posterior análisis. Emotiv cuenta con siete pares de electrodos colocados de forma simétrica a lo largo del cuero cabelludo.

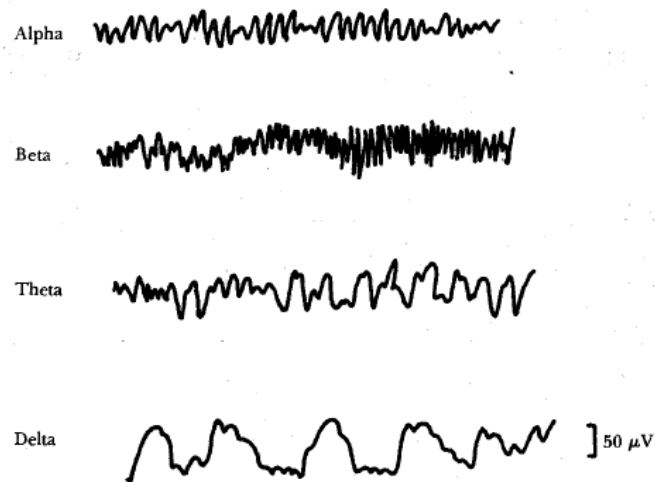
El cerebro está conformado por más de doce billones de neuronas distribuidas a lo largo del órgano encefálico. Este órgano cuenta con pliegues entrantes y fisuras que proveen una mayor área superficial en donde se encuentran las neuronas. Las neuronas, conformadas por dendritas, axón, núcleo y pericarion, envían señales a otras neuronas por medio de las neuronas adyacentes. Estas señales son pulsos eléctricos que pueden ser captados por los electrodos.

Es de suma importancia recalcar el hecho que el cerebro tiene evaginaciones e invaginaciones características de cada individuo, por lo que un mapeo tridimensional de las señales eléctricas diferirá de un cerebro a otro incluso cuando se obtiene la misma respuesta a un estímulo, ya que la superficie no es idéntica. Sin embargo, los estudios encefalográficos han demostrado que hay distintas regiones que se comportan de manera singular según distintas actividades cerebrales. Es por esto que se ha logrado determinar múltiples regiones del cerebro con distintas funciones a su cargo.

Hay distintos tipos de electrodos con distintos propósitos. Entre estos electrodos se encuentran los superficiales (sobre el cuero cabelludo), los basales (base del cráneo sin procedimiento quirúrgico), y los electrodos quirúrgicos (corticales o intracerebrales). Estas distintas herramientas se utilizan para realizar los electroencefalogramas (EEG), electrocorticograma (ECoG) y/o estéreo electroencefalograma (E-EEG). Las ondas que estos electrodos detectan se clasifican en distintas categorías según su función y periodicidad. Estas ondas pueden ayudar a catalogar el estado anímico, e incluso el tipo de pensamientos de un individuo. Los tipos de onda son alfa, beta, delta y theta, y aunque se encuentran presentes en todo momento, es la combinación la que varía al cambiar el estado de la persona.

Las ondas alfa indican un estado consciente y relajado. Esto nos indica que el individuo se encuentra sin estrés, ni muy concentrado. Estas ondas tienen una frecuencia entre 8 y 13 Hz, localizándose principalmente en la zona occipital. Su amplitud varía entre 20 y 200 mV. Las ondas beta, entre 14 y 30 Hz (con la posibilidad de llegar hasta 50 Hz) se encuentran principalmente en las regiones parietal y frontal, subdividiéndose en beta 1 y beta 2. Las ondas beta 1 indican un estado consciente y alerta, mientras que las ondas beta 2 indican un estado de tensión. A una frecuencia entre 4 y 7 Hz se encuentran las ondas theta en las regiones parietal y frontal. Estas son indicio de un estado de sueño ligero. También se sugiere que se registran en su mayor parte en el encéfalo de infantes. Por último, las ondas theta se presentan durante sueños profundos. Éstas se encuentran por debajo de los 3,5 Hz.

Figura 152. Tipos de onda detectados por electrodos



6. **Interacción Humano-Computador.** La Interacción Humano-Computador, o más bien conocida por sus siglas en inglés HCI (Human Computer Interaction) es el estudio de cómo las personas interactúan y/o reaccionan con las computadoras y averiguar hasta qué grado las computadoras están diseñadas o no para una interacción satisfactoria. Como su nombre lo indica, HCI está compuesto por tres grandes partes: El Humano, para futuras referencias será conocido como el Usuario, la Computadora y la Interacción. A continuación, se detalla cada una:

- a. **Usuario:** Según la Real Academia Española este término significa “*Que usa algo*”. Bajo el contexto de HCI y con base en la RAE podemos definir a un usuario como la persona que utiliza la computadora y/o el sistema computacional.
- b. **Computadora:** Al hablar acerca de este término en HCI se refiere a cualquier tipo de tecnología que el usuario utilice, ya sea desde una computadora de escritorio a una larga escala de sistemas computacionales.
- c. **Interacción:** Como se sabe, hay obvias diferencias entre las computadoras y los humanos, y a pesar de esto, HCI busca asegurar que ambos puedan interactuar entre ellos de una manera exitosa. (Jones, 2011)
  - a. **Beneficios de HCI.** El beneficio principal de HCI es producir y entregar sistemas seguros, intuitivos y usables. Esta última característica es muy importante y por lo mismo los que utilicen HCI se deben de asegurar de lo siguiente:
    - 1) Entender completamente los factores que determinan cómo las personas utilizan la tecnología
    - 2) Desarrollar herramientas y técnicas que permitan la construcción de sistemas sustentables.

- 3) Obtener eficiencia, efectividad y una interacción segura y apropiada.
- 4) Comprender que las personas son la base fundamental del HCI. (Jones, 2011)

b. **Usabilidad.** Usabilidad es una medida de la interacción de la experiencia del usuario asociado a una interfaz Figura, como una página web o una aplicación de software. (Comunidad, Introduction to User-Centered Design, 2012). Entre algunas características que debería de contar un sistema usable deberían de ser:

- 1) Fácil de aprender.
- 2) Fácil de recordar cómo usarlo.
- 3) Efectivo para usar.
- 4) Eficiente para usar.
- 5) Ergonómico para usar.

c. **Factores dentro de HCI.** Bajo el contexto de HCI se deben de considerar distintos y numerosos factores para el análisis y el diseño de un sistema utilizando los principios de HCI. Entre los factores más importantes a considerar se encuentran los siguientes:

- 1) Ambiente: tales como ruido en el lugar, iluminación, ventilación, factores de salud y seguridad.
- 2) Tipo de usuario: tales como capacidades cognitivas, motivación, satisfacción, experiencia.
- 3) Interfaz de usuario: tales como entradas del sistema, estructura del sistema, colores, tipo de íconos, tipo de gráficos.
- 4) Funcionalidades del sistema: tales como tipo de hardware, tipo de software, tipo de aplicación.
- 5) Factores de productividad: tales como complejidad del desarrollo, habilidades del equipo, número de tareas.
- 6) Restricciones: tales como costos de producción, equipo físico con el que se cuenta, presupuesto. (Jones, 2011)

d. **Principios de HCI.** Como se ha mencionado anteriormente, HCI busca crear de la mejor manera, y siempre realizando un diseño centrado en el usuario, una interfaz intuitiva y fácil de utilizar para cualquier persona. Para cumplir con el objetivo principal de HCI se deben de tener en cuenta los siguientes principios de diseño (Green, 2008):

- 1) Enfoque en el usuario y las tareas: En este principio se debe de establecer el perfil del usuario apropiado para el uso de la aplicación y, asimismo, determinar si estos usuarios son capaces de realizar las tareas que serán asignadas al momento de ejecutar las pruebas sobre la misma. Además

de esto, se debe de definir estas tareas y el tiempo estimado en que se realicen cada una de ellas. (Green, 2008)

- 2) Establecer métricas para las pruebas: Se deben de probar la aplicación con usuarios que utilizarán la misma, si no todos los días, la mayoría de ellos. Asimismo, se deben de establecer métricas específicas y cuantificables para las pruebas. (Green, 2008)
- 3) Realizar Iteraciones: Luego de cumplir con los dos principios anteriores, se deben de seguir los siguientes pasos para un proceso exitoso:
  - a. Diseñar la interfaz a utilizar.
  - b. Realizar pruebas sobre la interfaz.
  - c. Analizar los resultados de las pruebas sobre la interfaz.
  - d. Repetir el proceso hasta obtener mejores resultados

Este último principio se debe de repetir hasta el punto de crear una interfaz intuitiva, amigable y fácil de utilizar para la aplicación que se esté desarrollando. (Green, 2008).

e. **Etapas del HCI.** Debido a que HCI es una metodología de diseño, debe contar con etapas específicas en las que se realizan diferentes actividades. Estas etapas son las siguientes:

- 1) Benchmarking: consiste en aprender de los otros. Identificarlos, estudiarlos y mejorar basándose en lo que se ha aprendido (Boxwell JR., 1995). Su objetivo principal es bucear en un enorme fondo de conocimientos de forma que el conocimiento, refiriéndose a la formación colectiva y la experiencia de otros, pueda ser utilizado por aquellos que desean mejorar sus propias organizaciones (Boxwell JR., 1995). Actualmente, el benchmarking se está practicando ampliamente por dos razones:
  - a. Dar una ventaja competitiva a la organización que realiza el estudio de benchmarking.
  - b. Superar la actuación de la competencia.
- 2) Benchmarking de usabilidad: En esta etapa se realizan pruebas de usabilidad con los productos o servicios que se lograron encontrar en la parte de Benchmarking. Asimismo, estas pruebas tienen que tener tareas a realizar con los productos y se realiza un análisis de los resultados. (Preece, 1994)
- 3) Prototipar: Este término se refiere al proceso del desarrollo de prototipos. Este es una etapa fundamental del diseño centrado en el usuario porque permite a los diseñadores obtener una retroalimentación de los diseños que han propuesto para alguna interfaz. (Preece, 1994)
- 4) Iteraciones de entrevistas: Esta etapa se basa en la realización de entrevistas a diferentes personas a modo de conseguir siempre una retroalimentación de las mismas. Las iteraciones se realizan de la siguiente manera:
  - a. Diseñar la interfaz a utilizar.
  - b. Realizar pruebas sobre la interfaz.
  - c. Analizar los resultados de las pruebas sobre la interfaz.

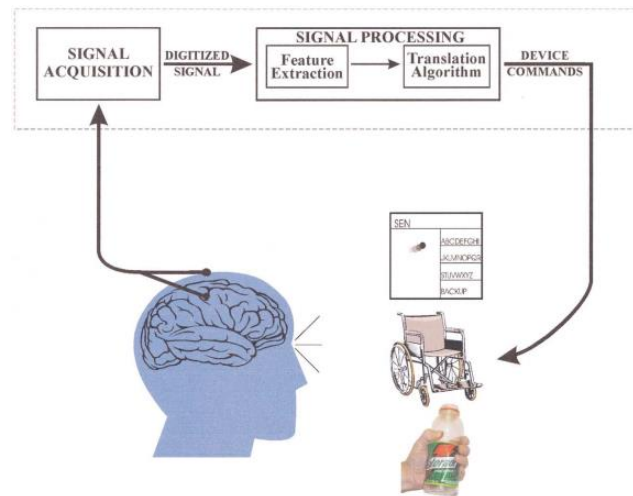
- d. Repetir el proceso hasta obtener mejores resultados para crear una interfaz intuitiva, amigable y fácil de utilizar para la aplicación que se esté desarrollando. (Green, 2008)
- 5) Pruebas de usabilidad: Esto se refiere a la evaluación de un producto o servicio poniendo sus funcionalidades a prueba con usuarios representativos. Normalmente, durante la prueba, los participantes de la prueba tratarán de realizar algunas tareas que el evaluador le pida, mientras que el evaluador observa y escucha a las personas, anotando cualquier observación que crea competente. El principal objetivo de estas pruebas es identificar cualquier problema de usabilidad y determinar la satisfacción de la persona con el producto o servicio. (Comunidad, Usability Testing, 2013). Para la realización de estas pruebas se debe generar un ambiente a continuación a describir:
- a. Un espacio en dónde uno o varios observadores puedan tomar nota sobre los comportamientos de las personas.
  - b. Este espacio debe de poseer toda la tecnología, dispositivos de hardware como de software, a utilizar para la prueba.

7. **Interfaz Cerebro-Computador.** La Interfaz Cerebro-Computador, o más bien conocida por sus siglas en inglés BCI (Brain Computer Interface) es un sistema en una computadora que recibe señales del cerebro, las analiza y las traslada por medio de comandos relacionados a las mismas a un dispositivo que realiza una operación deseada. (Wolpaw, Birbaumer, McFarland, Pfurtscheller, & Vaughan, 2002)

a. **Partes del BCI.** El objetivo principal de tecnología BCI es lograr establecer un sistema de control que provea un canal de comunicación entre el cerebro y la computadora. Debido a esto, este sistema de control debería de contar con lo siguiente:

- 1) Detección de la señal: En esta etapa, la detección de señales por medio de electrodos, amplificada y digitalizada.
- 2) Extracción de la información a utilizar: Esta etapa obtiene la información necesaria u competente de las señales recogidas.
- 3) Implementación del algoritmo de traducción de la señal: Esta etapa traduce las señales obtenidas en comandos de control en dispositivos para conveniencia del usuario.
- 4) El dispositivo de salida: Es el dispositivo, de software o hardware, que demuestran el análisis de las señales recibidas, luego de ser procesadas.
- 5) Protocolo de operaciones: Este protocolo define la forma en que funciona el sistema, ya sea que tenga comunicación o no, si la información transmitida será procesada por el usuario o la computadora, así como la retroalimentación que el usuario provea.

Figura 153. Esquema general de la tecnología BCI



b. **Aplicaciones de BCI.** Debido a la gran gama de aplicaciones en las que se pueda utilizar aplicaciones BCI, a continuación, se muestran algunas ramas en las que se podría utilizar BCI:

- 1) Medicina: En este campo se pueden utilizar aplicaciones BCI para la prevención de enfermedades crónicas, así como la detección y diagnóstico y la rehabilitación de enfermedades.
- 2) Ambientes inteligentes: se pueden desarrollar aplicaciones BCI para el control de casas inteligentes, lugares de trabajo o implementación de tendencias tecnológicas como el internet de las cosas con BCI.
- 3) Educación: Se podrían establecer nuevos métodos de estudio y aprendizaje para personas con problemas de aprendizaje.
- 4) Juegos y entretenimiento: Al combinar aplicaciones BCI con diferentes modalidades de juegos, se puede crear toda una nueva gama de juegos, así como una nueva tendencia de desarrollo de videojuegos utilizando BCI.

## B. Marco metodológico

1. **Investigación.** La primera etapa del Megaproyecto EMOTRIX se dio en los primeros meses del año 2016 en la que se realizó varias investigaciones y estudios acerca de la factibilidad del proyecto. Además de esto, estas investigaciones sirvieron para delimitar el alcance de la primera fase del proyecto y, asimismo, para ampliar los conocimientos que tenía el grupo sobre temas de reconocimiento y manipulación de emociones.

En base a esta investigación se lograron definir los entregables para esta versión, los cuales se mencionaron anteriormente. En esta parte, además, se determinó que el principal alcance del proyecto sería el reconocimiento de dos emociones fundamentales: felicidad y tristeza.

Luego de determinar el alcance principal y de tener definido los entregables, tanto físicos como de software, se distribuyeron los mismos en cada miembro del grupo para que se empezara una etapa más a fondo de los materiales, recursos y demás para la elaboración del proyecto. En esta fase se determinó que la mejor manera de realizar la interfaz para EMOTRIX sería utilizar un diseño centrado en el usuario. Este diseño se consigue con base en la metodología de diseño HCI, la cual se decidió utilizar para la realización del proyecto.

Bajo el primer principio de trabajo de HCI, se definió el perfil del usuario final para EMOTRIX. Luego de establecer los objetivos generales del proyecto, el perfil de usuario quedó establecido en “Aquella persona que desee reconocer dos distintos tipos de emociones: alegría y tristeza, utilizando un Software Open-Source fácil de entender e intuitivo al momento de utilizarlo”. Debido al alcance que tiene el proyecto en sí, no se especula que la persona interesada en la utilización de este Software sea directamente relacionada y/o involucrada en el área de Ciencias de la Computación. Esto se debe a que la mayoría de personas que tratan de utilizar softwares de este tipo están muy relacionados a estudios médicos, científicos y psicológicos, siempre relacionados con la identificación de las emociones de las personas en respuesta a diferentes estímulos que se le puedan inducir tanto de manera voluntaria como involuntaria.

Asimismo, en esta parte se realizó un estudio de Benchmarking para conocer tecnologías relacionadas a este proyecto. En el Cuadro 10 se muestran algunas de las tecnologías BCI más utilizadas para la detección de emociones humanas.

Cuadro 10. Algunas plataformas BCI existentes

<b>Nombre</b>	<b>Descripción</b>	<b>Precio</b>
Melon Headband	Headband para mediciones biométricas.	\$149
NeuroSky	pulsera que toma mediciones biométricas, pero también es capaz de medir las actividades del cerebro.	\$99.99
Emotive EPOC	Headset para medir señales eléctricas del cerebro.	\$399
MyndPlay BrainBand	Headband que detecta la actividad cerebral y lo transmite a la computadora para luego ser procesada. Niveles de concentración y relajación.	\$249
OpenBCI	Dispositivos bio-sensores programables (una placa) con los electrodos que pueden medir señales EEG. Se puede comprar por separado los electrodos: 10 electrodos compatibles con las tarjetas OpenBCI cuestan \$29.	Varía por el kit solicitado

Todas estas marcas ofrecen dispositivos de *hardware* que realizan la lectura y transmisión de señales electroencefalográficas a la computadora. Además, proveen un SDK para gestión y uso de las lecturas. A excepción de OpenBCI, estas plataformas ofrecen *software* libre de costo para implementación de sistemas que interactúan con el cerebro, pero todos son compatibles con sus respectivos dispositivos de *hardware* que en todos los casos tiene un precio mayor a \$99.00. El caso de OpenBCI es que es una plataforma abierta. Al

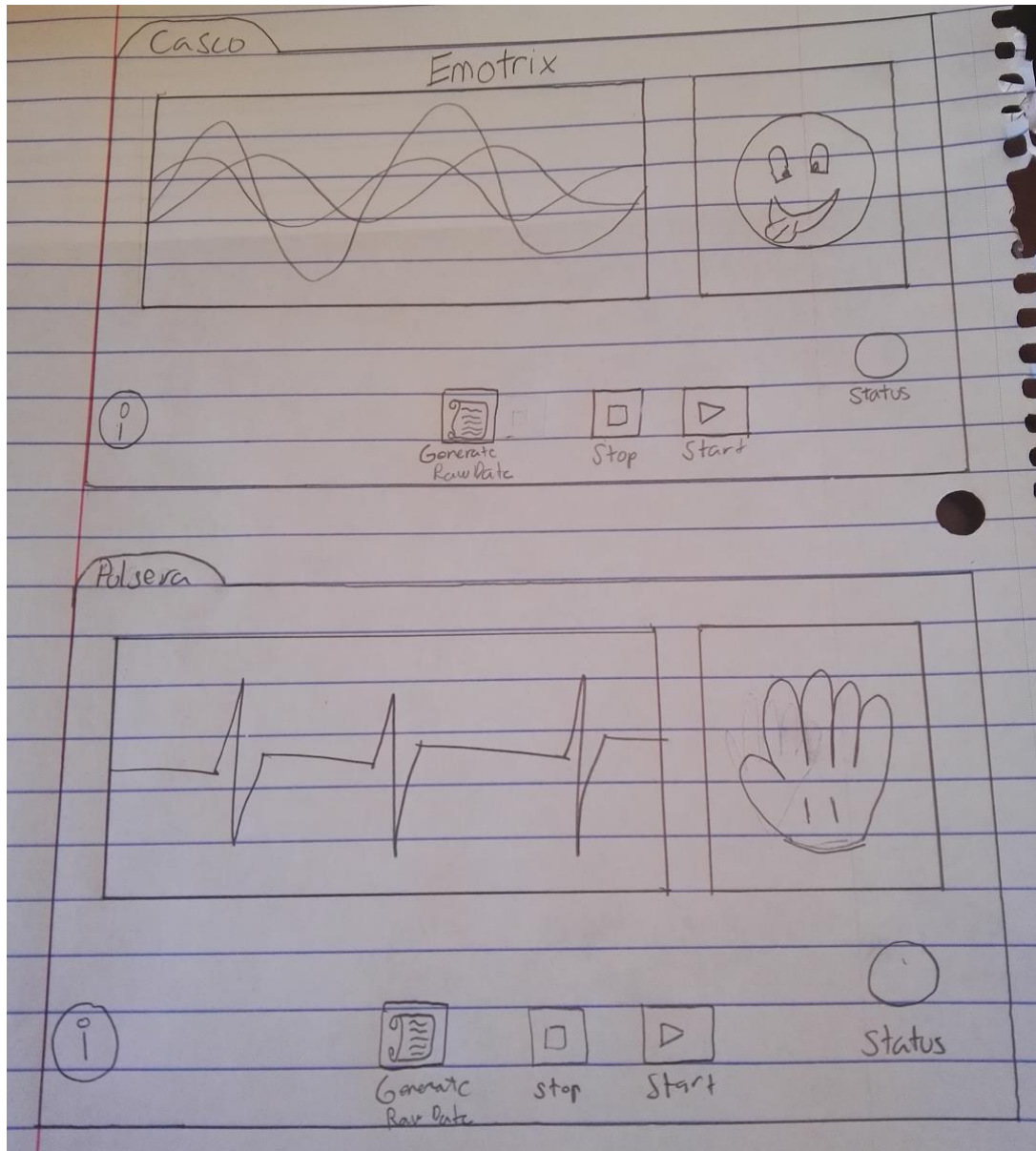
22 de abril de 2016, OpenBCI lanza su dispositivo de *hardware* a la venta, y ofrece el *software* para su uso (y código fuente del mismo) libre de costo. Así mismo permiten la descarga de archivos para diseño, impresión y ensamblado de un “casco” para lectura de señales EEG. Lo que OpenBCI hace es lo más parecido a lo que Emotrix pretende lograr.

2. **Diseño.** Para la etapa de diseño de la interfaz para EMOTRIX se utilizó el proceso de HCI para encontrar la mejor manera de desarrollar una interfaz al tener una serie de pruebas y análisis sobre las mismas, y así, poder lograr el diseño de una interfaz intuitiva y amigable que mostrara las funcionalidades del API de EMOTRIX.

a. **Primeras entrevistas.** Como primer paso se desarrolló un prototipo en papel de cómo se pensaba que sería la interfaz de EMOTRIX la cual se muestra en la Figura 154. Luego de realizar este prototipo, se prosiguió a la parte de las entrevistas al usuario. Como ya estaba definido el perfil de usuario, las entrevistas se centraron en personas que clasificaran como posibles usuarios finales de la aplicación. Se realizaron 20 entrevistas a diferentes tipos de personas, entre las cuales se encontraron diez personas relacionadas con el área de Psicología, ya que fueron tanto catedráticos de diferentes áreas de psicología como estudiantes de la carrera; y las otras diez personas eran estudiantes de diferentes carreras dentro del área de Ingeniería. En primer lugar, se le dio una explicación al entrevistado acerca del porqué de la entrevista y qué era lo que se esperaba del mismo. Las entrevistas tuvieron una duración de 10 minutos en la mayoría de los casos.

En estas entrevistas se buscaba principalmente la opinión de los usuarios con respecto a la organización de los componentes de la interfaz, los colores que podría tener la interfaz y la opinión acerca de las funcionalidades de la misma. Debido a esto, no se les solicitó a los entrevistados que realizaran alguna tarea en específico más que compartieran sus opiniones sobre la interfaz. En el anexo 1 se encuentran las opiniones de las personas en esta etapa.

Figura 154. Primer prototipo en papel de la interfaz de la aplicación



b. **Entrevistas de usabilidad en Mockup.** En esta parte del diseño se realizaron nuevamente 20 entrevistas a personas diferentes a las entrevistadas en la primera del diseño. Estas entrevistas fueron realizadas para que, por medio de usuarios finales de la aplicación, se pudiera determinar si la interfaz era entendible y, asimismo, intuitiva para los mismos. La realización de cada entrevista duró más que cualquiera de las entrevistas realizadas anteriormente ya que en estas al usuario final se le pasó un consentimiento informado, el cual se encuentra en el anexo 2. También se le pasó al entrevistado una serie de preguntas antes y después de que utilizaran la interfaz para conocer información general acerca de los entrevistados, sus conocimientos con proyectos similares y los comentarios acerca de la interfaz. Estas entrevistas se pueden encontrar en el anexo 3 y anexo 4, respectivamente.

Para estas entrevistas se le dio al entrevistado un listado de tareas a realizar en el Mockup con las que se probarían las funcionalidades del API de EMOTRIX. El listado de tareas se muestra a continuación:

- 1) Ingresar a la aplicación y realizar una medición de 30 segundos del casco.
- 2) Reproducir la medición del casco tomada anteriormente.
- 3) Ingresar a la aplicación y realizar una medición de 30 segundos de la pulsera.
- 4) Reproducir la medición de la pulsera tomada anteriormente.
- 5) Ingresar a la aplicación y realizar una medición de 30 segundos del casco, más alguna funcionalidad de la pulsera.
- 6) Generar la data obtenida de alguna medición.
- 7) Lea las “Instrucciones” y el “Acerca De” de la aplicación.
- 8) Utilizar la sección de “Demo” de la aplicación.

Asimismo, y según los principios de HCI, se tomaron los tiempos respectivos que tomó cada entrevistado en realizar cada tarea asignada. Esto con el fin de analizar el tiempo promedio en que un usuario final lograría realizar alguna acción dentro de la interfaz de la aplicación. El tiempo máximo establecido por actividad fue de un minuto. Aquellas tareas que solicitaban por lo menos 30 segundos de medición se les asignó un tiempo máximo de realización de minuto y medio.

Para estas entrevistas se utilizó el diseño del Mockup realizado para las entrevistas. Este diseño se puede observar en la Figura 155, Figura 156, Figura 157, Figura 158, Figura 159, Figura 160, Figura 161, Figura 162 y Figura 163.

Figura 155. Mockup de inicio de la aplicación

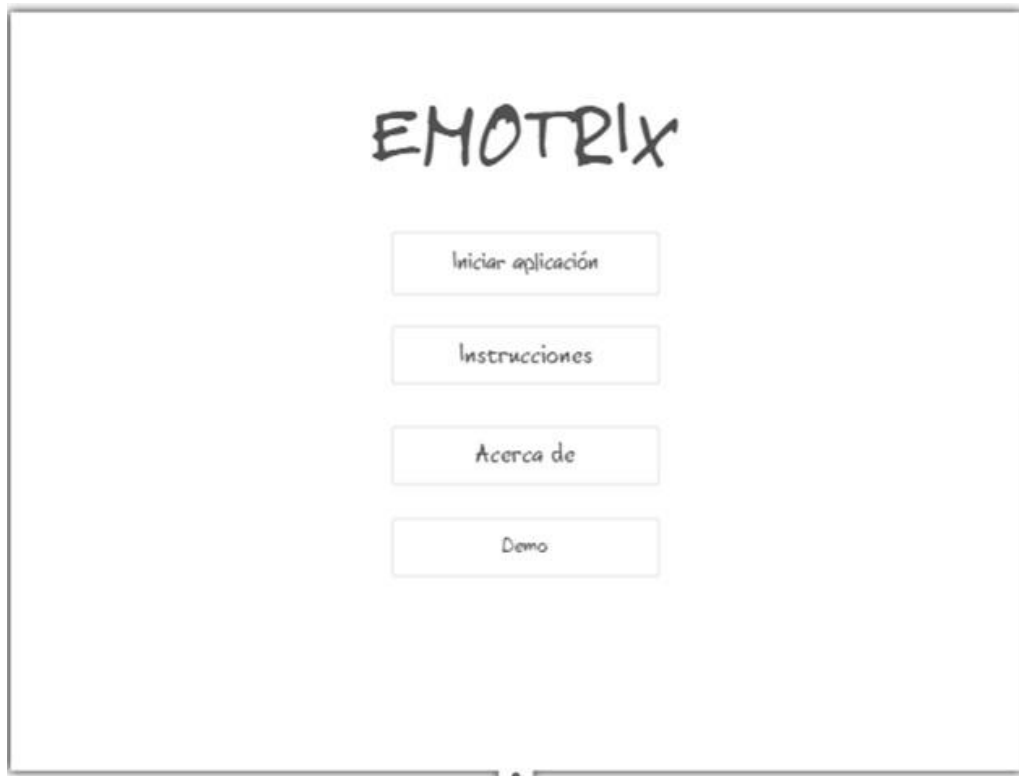


Figura 156. Mockup de la interfaz del casco



Figura 157. Mockup de interfaz de la pulsera

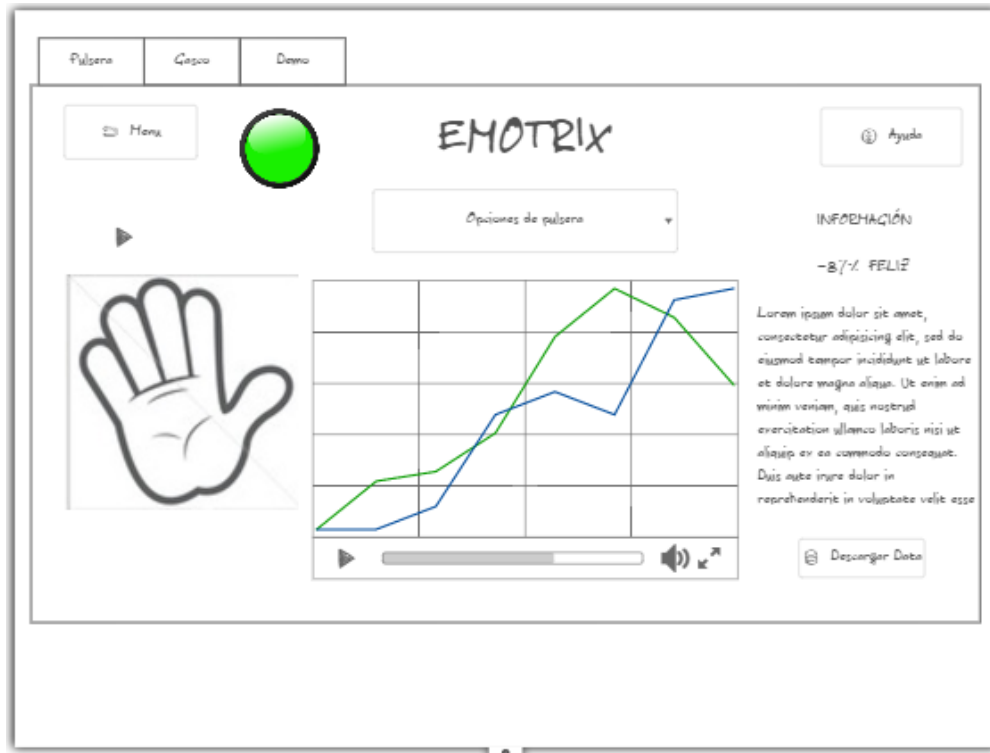


Figura 158. Mockup de la interfaz de la ventana de instrucciones



Figura 159. Mockup de la interfaz de instrucciones del casco



Figura 160. Mockup de la interfaz de instrucciones del casco



Figura 161. Mockup de la interfaz del demo de la aplicación

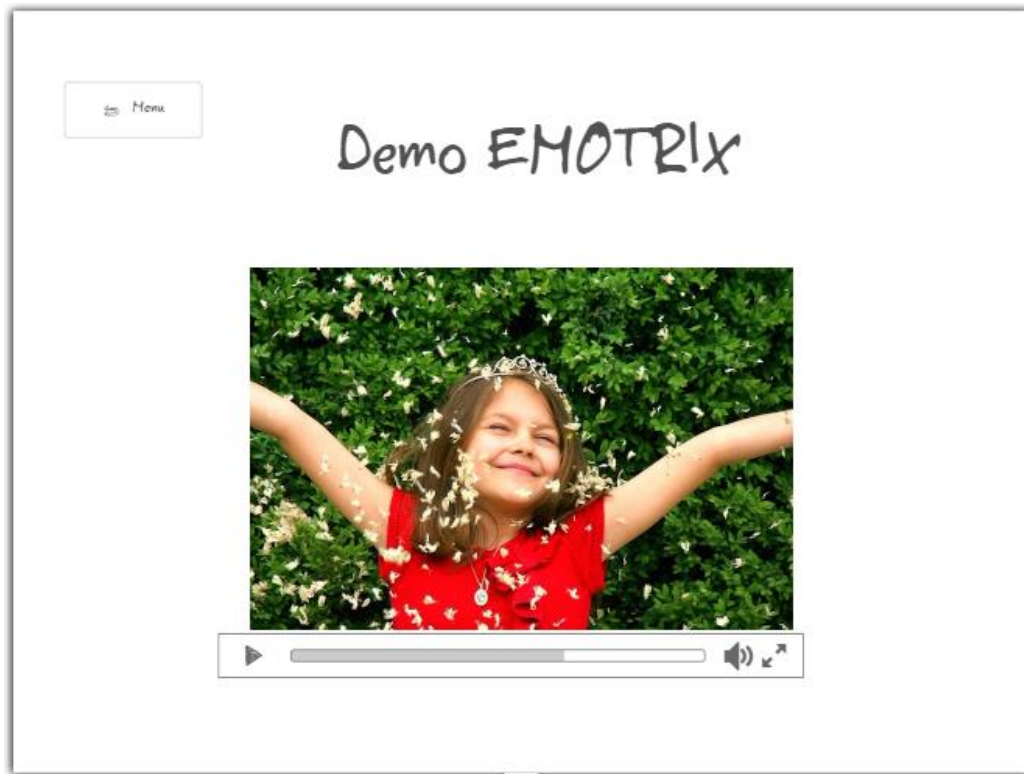


Figura 162. Mockup de la interfaz de la sección de ayuda

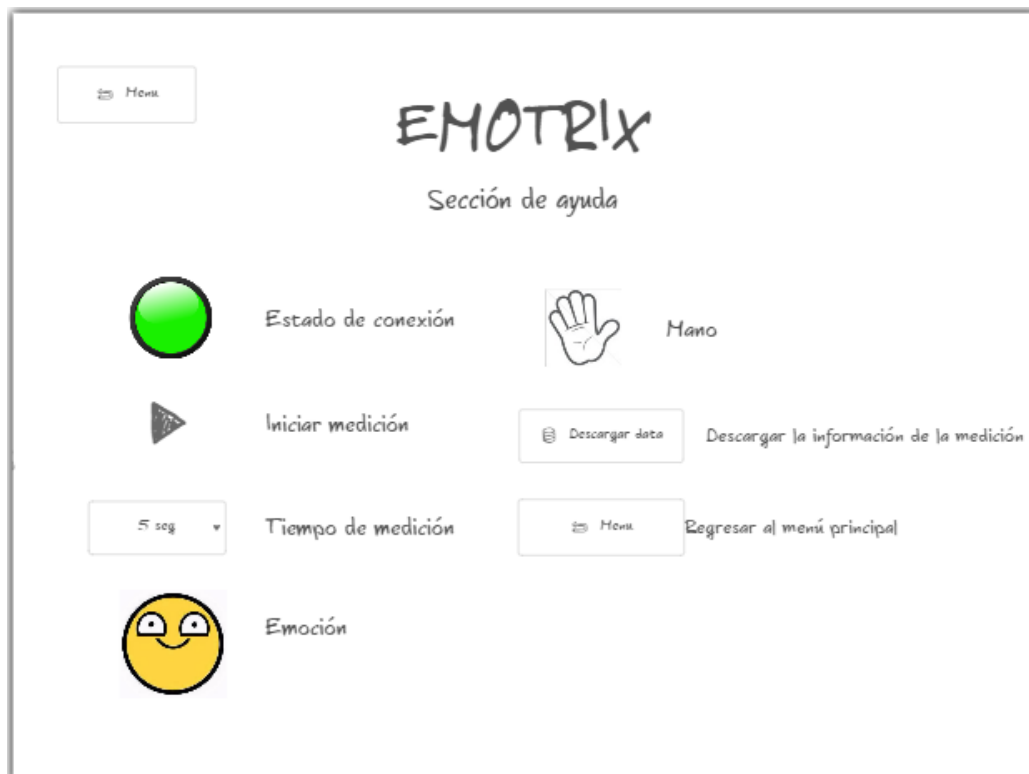


Figura 163. Mockup de la interfaz de la ventana acerca de Emotrix



c. **Entrevistas de usabilidad en aplicación.** Para estas entrevistas se utilizó la misma metodología que en las entrevistas de usabilidad con Mockup ya que se utilizaron las encuestas realizadas, juntamente con el consentimiento informado del Anexo 2. Además de esto, se utilizó una computadora personal para que las personas lograran utilizar la interfaz de la aplicación. Estas entrevistas se realizaron en un ambiente en el cual no se tenía ningún tipo de ruido que interrumpiera la atención de la persona. Este ambiente se seleccionó ya que para la obtención de datos que tendrá EMOTRIX, se necesita un ambiente tranquilo y sin ninguna molestia. Se realizaron ocho entrevistas a diferentes personas y las tareas a realizar fueron las mismas que en las entrevistas pasadas. Asimismo, el ambiente seleccionado para estas entrevistas fue un salón de clase, en el que las personas llegaban y realizaban lo que pedía las diferentes tareas solicitadas.

Para estas entrevistas se utilizó el diseño de la aplicación final de EMOTRIX. Este diseño se puede observar en la Figura 164, Figura 165, Figura 166, Figura 167, Figura 168, Figura 169, Figura 170, Figura 171 y Figura 172.

Figura 164. Ventana inicial de la aplicación

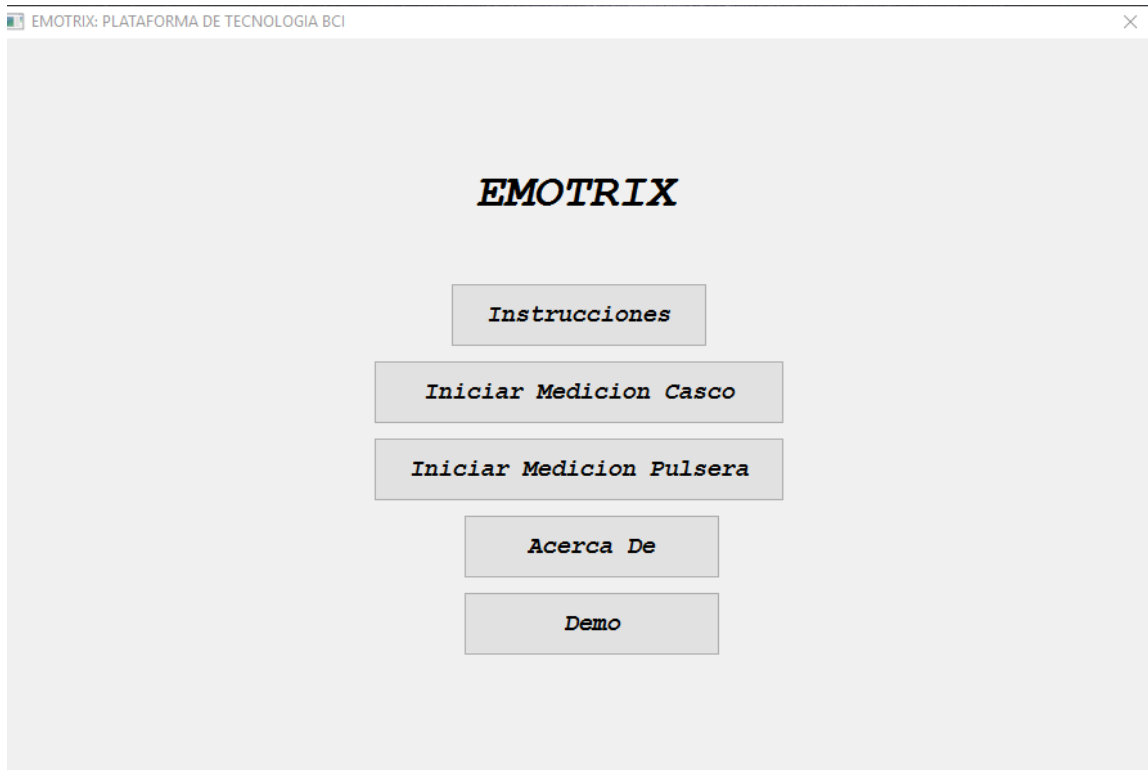


Figura 165. Ventana general de instrucciones

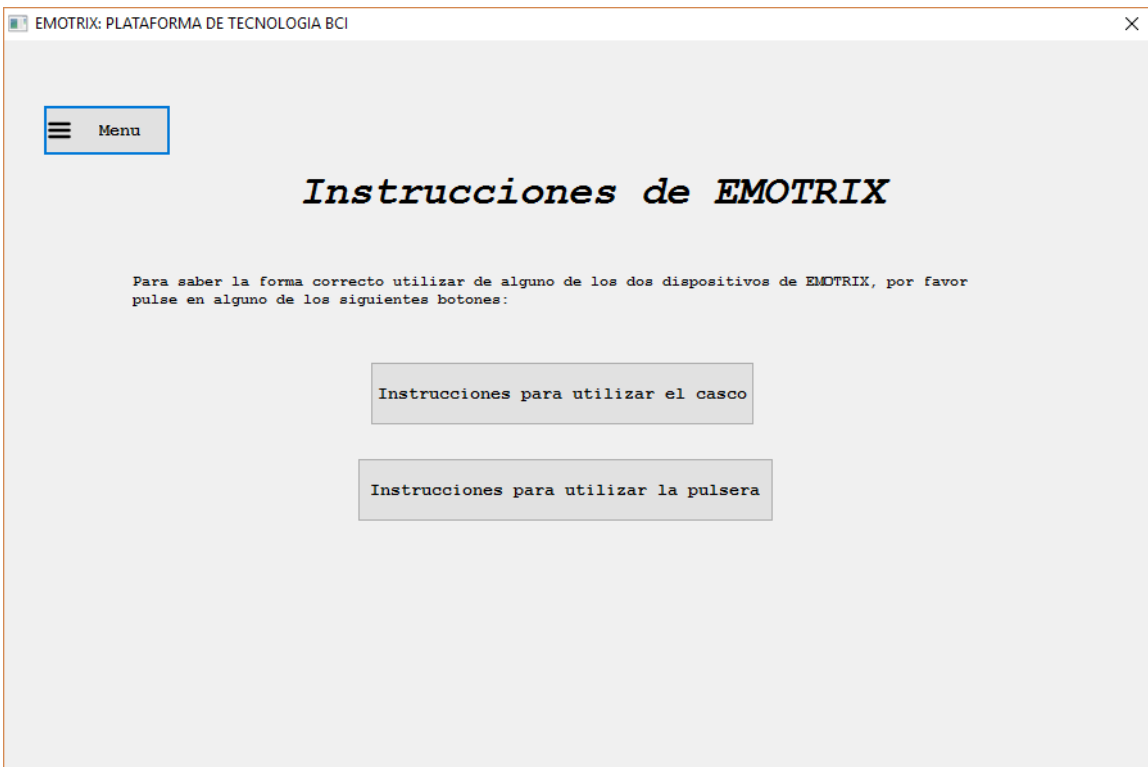


Figura 166. Ventana de instrucciones sobre el casco de EMOTRIX

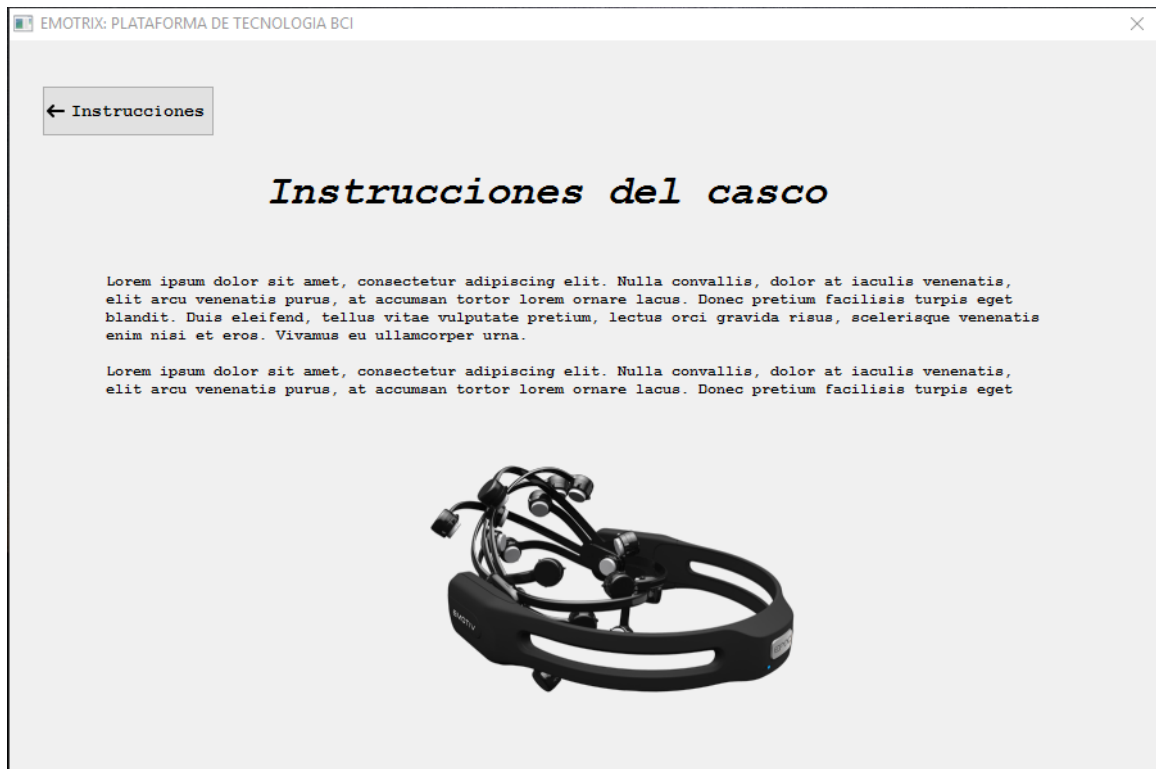


Figura 167. Ventana de instrucciones sobre la pulsera de EMOTRIX



Figura 168. Ventana del casco de EMOTRIX

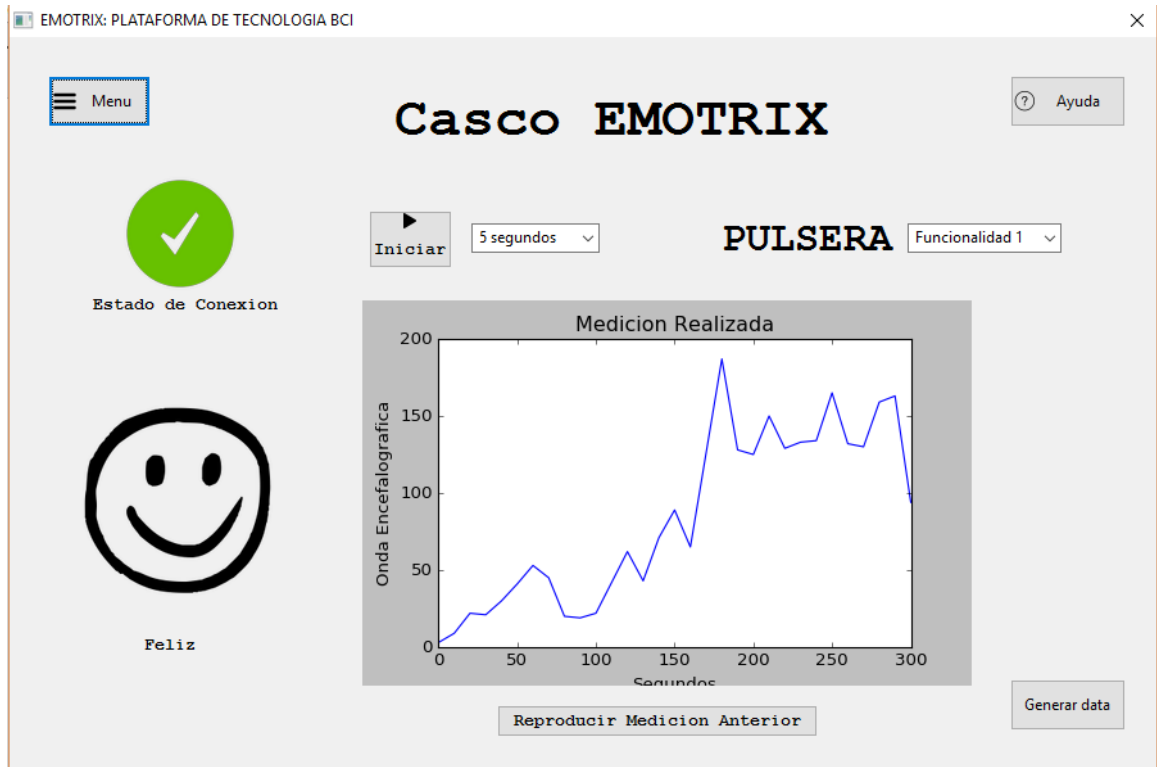


Figura 169. Ventana de la pulsera de EMOTRIX

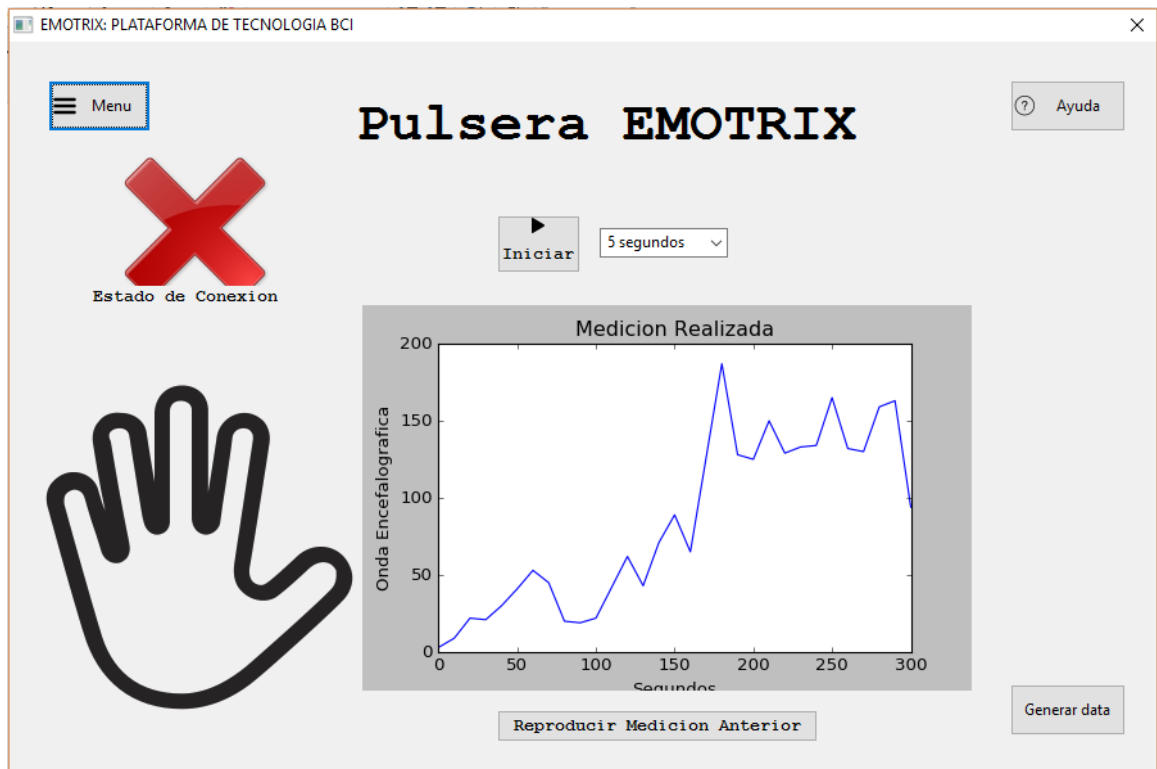


Figura 170. Ventana de información de ayuda al usuario

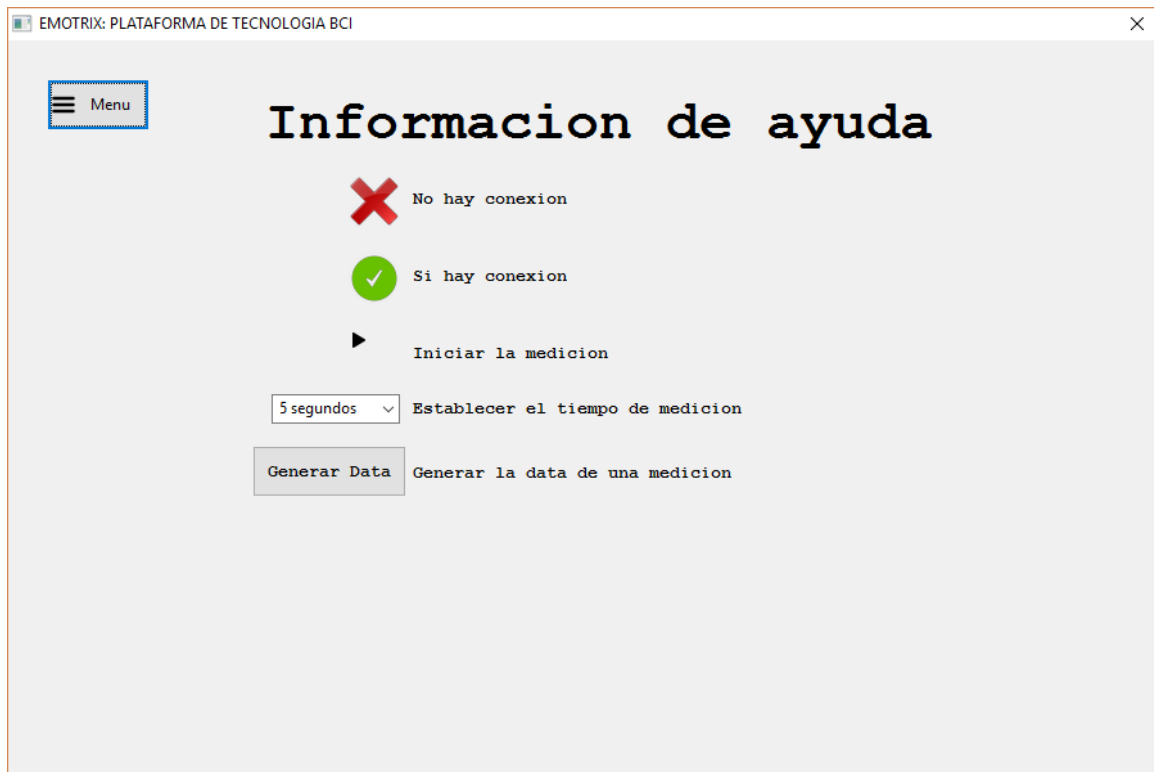


Figura 171. Ventana de información sobre EMOTRIX

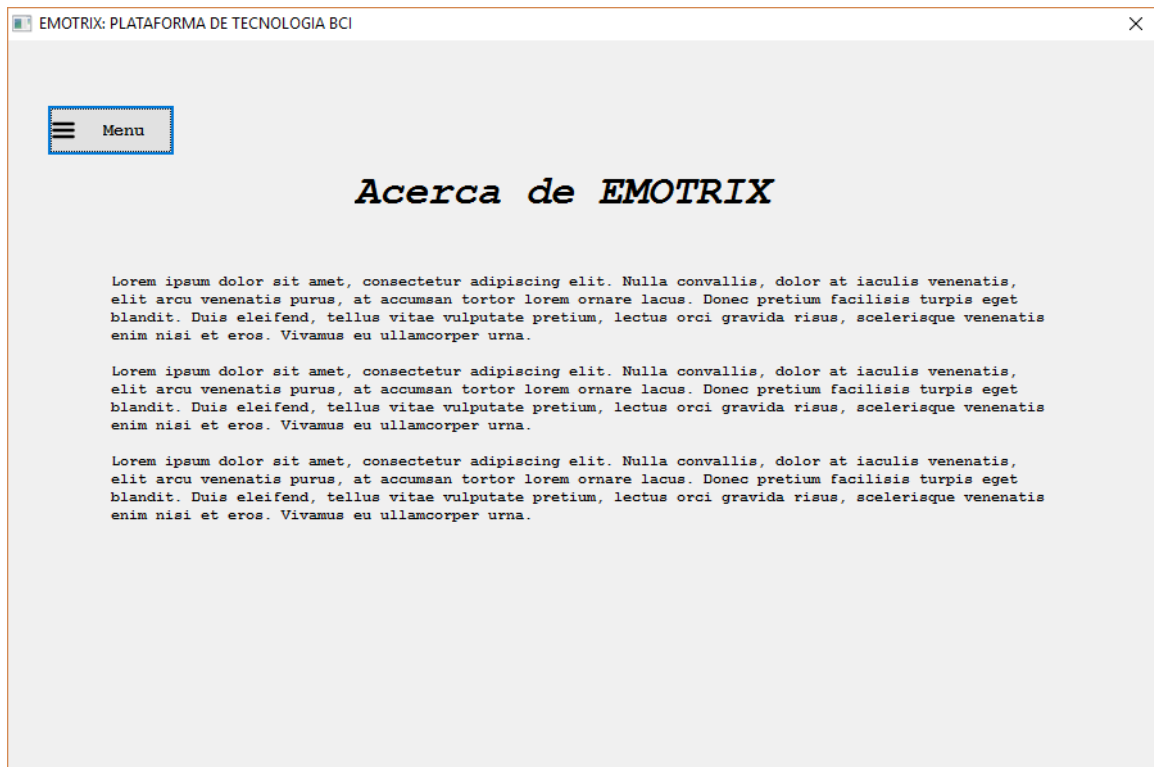


Figura 172. Ventana de demo de EMOTRIX



## C. Resultados

1. **Entrevistas con el prototipo de papel.** Los primeros resultados obtenidos se dieron en las entrevistas realizadas con el prototipo de papel para la interfaz. En estas entrevistas se recopilaban los comentarios más importantes y que tendrían un impacto mayor en la interfaz, y con base en los cuales se hicieron los cambios respectivos a la interfaz. Las primeras entrevistas se realizaron a 20 personas de diferentes edades y de ambos sexos. Entre estas personas se encontraron estudiantes de diferentes carreras universitarias, así como a personas graduadas de la Universidad. En esta etapa del proceso de iteración de entrevistas, se recopilaban los comentarios más significativos y con base en ellos se hicieron los cambios.

De los 20 entrevistados, más de 10 personas hicieron comentarios acerca de los colores que tendría la aplicación, ya que, con base en la explicación previa sobre el porqué de la entrevista, creían que los mismos podrían afectar en alguna manera la emoción que la persona estuviera sintiendo, sesgando así la medición y no obteniendo resultados efectivos o los resultados esperados. De la misma manera, varios de los entrevistados sugirieron un menú de inicio ya que en el prototipo en papel no se lograba entender cómo empezar a usar la interfaz o qué botón iniciaba alguna acción en la misma. Este tipo de comentarios influyó en la creación del menú principal de la interfaz. Nuevamente, de los 20 entrevistados, más de 10 personas sugirieron que se cambiara el orden de los botones y la ubicación de los mismos. Estas personas hacían énfasis en la forma en que las personas leen ya que en la mayoría de culturas y países se lee de izquierda a derecha y el

orden de los botones podría ser un problema. De la misma manera, un gran número de las personas entrevistadas sugirieron que se agregara alguna sección de ayuda y de información acerca del proyecto. Así de esta manera las personas podrían tener un conocimiento más amplio acerca de la aplicación y tendrían una herramienta para utilizar al momento de estar usando la aplicación. En las figuras 173, 174, 175 se pueden observar los cambios realizados.

Figura 173. Menú inicial agregado

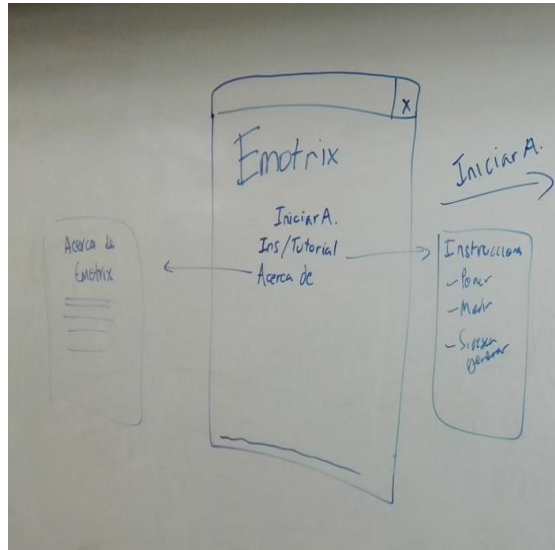


Figura 174. Cambios realizados a la interfaz del casco

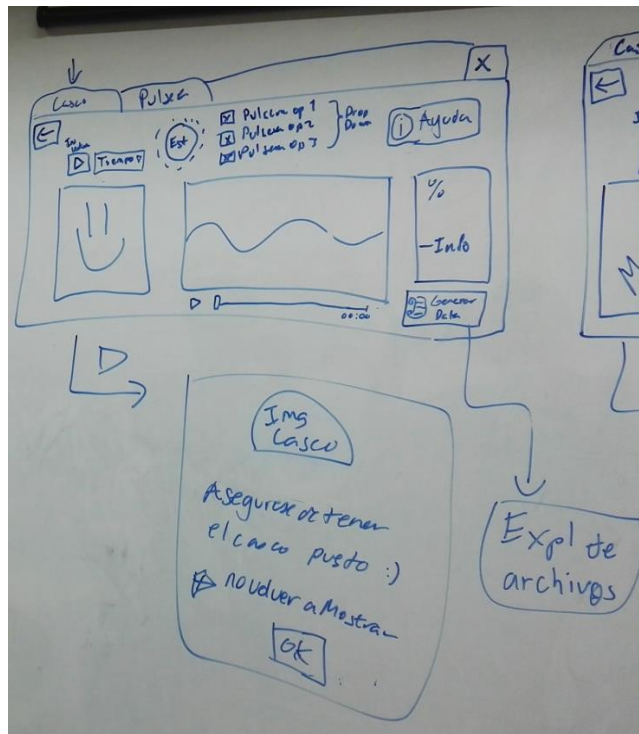
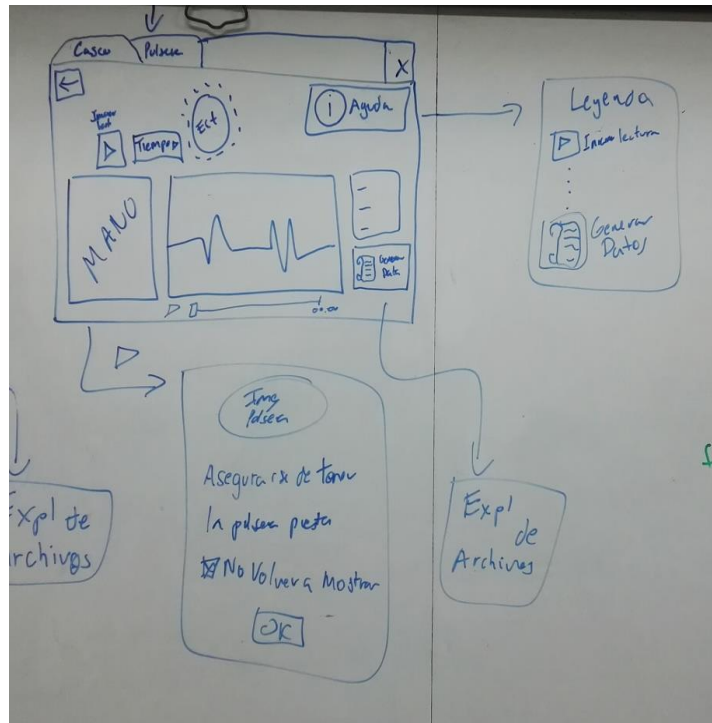


Figura 175. Cambios realizados a la Interfaz de la pulsera



2. Entrevistas de usabilidad con Mockup. El propósito principal de las entrevistas de usabilidad es comprobar que todas las funcionalidades de una aplicación sean lo suficientemente entendibles y así verificar que el usuario final para el cual está destinada la aplicación la pueda utilizar. Estas entrevistas se realizaron a 20 personas, en las que se encontraban diferentes estudiantes de carreras universitarias y a varios catedráticos de diferentes universidades. Para estas entrevistas se utilizó el Mockup de las figuras mencionadas anteriormente en la etapa de Diseño de la Metodología realizada. A continuación, se presentan los resultados obtenidos en cada tarea que cada entrevistado realizó en la entrevista, así como un resumen general de los mismos.

a. **Tarea no.1: Ingresar a la aplicación y realizar una medición de 30 segundos del casco**

Cuadro 11. Resultados obtenidos en la tarea no.1 en la entrevista con Mockup

Usuario	Tiempo (seg)
1	3
2	14
3	11
4	18
5	15
6	28
7	32
8	22
9	21

## Continuación Cuadro 11

Usuario	Tiempo (seg)
10	30
11	24
12	30
13	28
14	35
15	25
16	22
17	20
18	24
19	19
20	18

**b. Tarea no.2: Reproducir la medición del casco tomada anteriormente**

En el Cuadro 12 se presentan los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 12. Resultados obtenidos en la tarea no.2 en la entrevista con mockup

Usuario	Tiempo (seg)
1	14
2	42
3	9
4	3
5	3
6	4
7	7
8	19
9	8
10	9
11	2
12	16
13	5
14	15
15	10
16	15
17	18
18	16
19	13
20	19

**c. Tarea no. 3: Ingresar a la aplicación y realizar una medición de 30 segundos de la pulsera.** En el Cuadro 13 se presentan los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 13. Resultados obtenidos de la tarea no.3 en la entrevista con mockup

Usuario	Tiempo (seg)
1	42
2	50
3	49
4	34
5	40
6	26
7	36
8	39
9	13
10	21
11	31
12	14
13	15
14	18
15	20
16	24
17	26
18	18
19	20
20	19

d. **Tarea no. 4: Reproducir la medición de la pulsera tomada anteriormente.** En el Cuadro 14 se muestran los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 14. Resultados obtenidos de la tarea no.4 en la entrevista con mockup

Usuario	Tiempo (seg)
1	12
2	14
3	11
4	7
5	9
6	17
7	15
8	7
9	5
10	10
11	12
12	7
13	5
14	9
15	7
16	15
17	18
18	12

Cuadro Continuación 14

Usuario	Tiempo (seg)
19	14
20	18

e. **Tarea no. 5: Ingresar a la aplicación y realizar una medición del casco, más alguna funcionalidad de la pulsera.** En el Cuadro 15 se muestran los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 15. Resultados obtenidos de la tarea no.5 en la entrevista con mockup

Usuario	Tiempo (seg)
1	52
2	47
3	15
4	16
5	26
6	12
7	40
8	32
9	21
10	49
11	30
12	28
13	35
14	38
15	29
16	25
17	28
18	26
19	27
20	23

f. **Tarea no. 6: Generar la data obtenida de alguna medición.** En el Cuadro 16 se muestran los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 16. Resultados obtenidos de la tarea no.6 en la entrevista con mockup

Usuario	Tiempo
1	6
2	13
3	8
4	21
5	18
6	27
7	13
8	15

Continuación Cuadro 16

Usuario	Tiempo
9	6
10	11
11	10
12	4
13	5
14	10
15	8
16	12
17	19
18	14
19	16
20	13

g. **Tarea no.7: Lea las “Instrucciones” y el “Acerca De” de la aplicación.** En el Cuadro 17 se muestran los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 17. Resultados obtenidos de la tarea no.7 en la entrevista con mockup

Usuario	Tiempo (seg)
1	16
2	26
3	33
4	11
5	7
6	17
7	10
8	8
9	35
10	30
11	34
12	27
13	29
14	26
15	30
16	12
17	18
18	13
19	18
20	14

h. **Tarea no.8: Utilizar la sección de “Demo” de la aplicación.** En el Cuadro 18 se muestran los tiempos obtenidos que cada entrevistado realizó en esta tarea.

Cuadro 18. Resultados obtenidos de la tarea no.8 en la entrevista con mockup

Usuario	Tiempo (seg)
1	21
2	30
3	19
4	16
5	17
6	19
7	22
8	20
9	6
10	14
11	21
12	2
13	13
14	15
15	12
16	17
17	14
18	20
19	12
20	18

i. **Resumen de las entrevistas de usabilidad con Mockup.** Para fines prácticos, se muestra un resumen de los resultados obtenidos en las entrevistas de Usabilidad realizadas a cada entrevistado. En el Cuadro 19 se muestra la información general recabada de los entrevistados. En ella se muestra la edad promedio de los entrevistados, así como la cantidad de hombres y mujeres entrevistadas.

Cuadro 19. Información general de los entrevistados en la entrevista con mockup

Información general	
Edad promedio (años)	28.5
Cantidad hombres	9
Cantidad mujeres	11

En el Cuadro 20 se muestra la cantidad de personas que pertenecen a los diferentes grados académicos que se especificaban en la encuesta pre-entrevista.

Cuadro 20. Cantidad de personas según su grado académico en la entrevista con mockup

Grado académico	
Diversificado	4
Universitario	14
Licenciatura	2

En el Cuadro 21 se muestran los porcentajes de la cantidad de personas que han contestado de manera afirmativa a la pregunta que se les hacía en la encuesta pre-entrevista.

Cuadro 21. Experiencia de los entrevistados en sistemas relacionados en la entrevista con mockup

Experiencia en Sistemas Relacionados	
¿Alguna vez ha oído acerca de tecnología BCI?	20%
Ha oído acerca del dispositivo EMOTIV EPOC	25%
¿Sabía que su cerebro produce señales Encefalográficas?	20%
Sabía que hay seis emociones fundamentales	55%

En el Cuadro 22 se muestra el tiempo promedio obtenido por los entrevistados al realizar cada una de las tareas en las que se les pidió durante la entrevista.

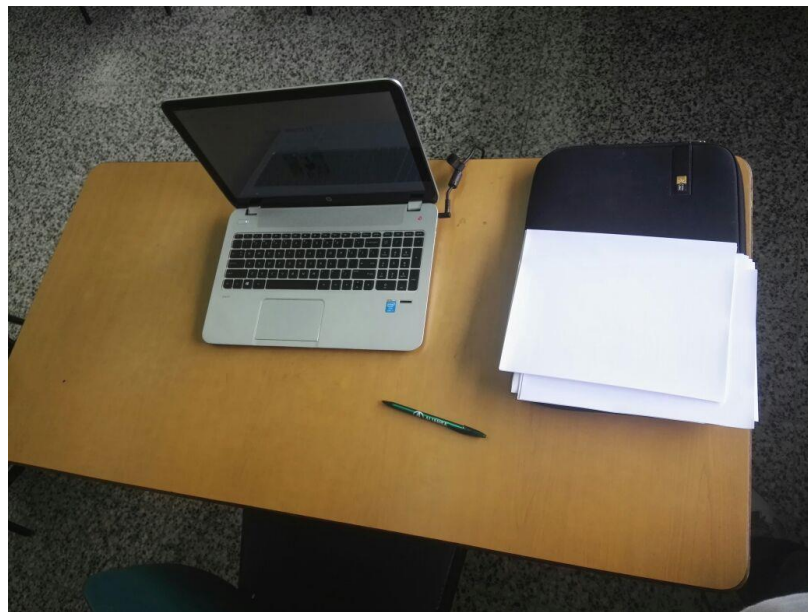
Cuadro 22. Tiempo promedio de los entrevistados por cada tarea realizada en la entrevista con mockup

Tarea	Promedio (seg)
Ingresar a la aplicación y realizar una medición de 30 segundos del casco	21.88
Reproducir la medición del casco tomada anteriormente	12.37
Ingresar a la aplicación y realizar una medición de 30 segundos de la pulsera	27.76
Reproducir la medición de la pulsera anteriormente.	11.22
Ingresar a la aplicación y realizar una medición del casco, más alguna funcionalidad de la pulsera.	30.01
Generar la data obtenida de alguna medición	12.43
Lea las “Instrucciones” y el “Acerca De” de la aplicación	20.78
Utilizar la sección de “Demo” de la aplicación	16.36

Figura 176. Salón utilizado para entrevistas de Usability Testing



Figura 177. Materiales utilizados para entrevistas de Usability Testing



**a. Tarea no. 1: Ingresar a la aplicación y realizar una medición de 30 segundos del casco.**

En el Cuadro 23 se presentan los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 23. Resultados obtenidos en la tarea no.1 en la entrevista con la aplicación

Usuario	Tiempo
1	17
2	20
3	24
4	31
5	25
6	35
7	32
8	24

b. **Tarea no. 2: Reproducir la medición del casco tomada anteriormente.** En el Cuadro 24 se presentan los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 24. Resultados obtenidos en la tarea no.2 en la entrevista con la aplicación

Usuario	Tiempo (seg)
1	21
2	11
3	23
4	22
5	17
6	20
7	18
8	19

c. **Tarea no.3: Ingresar a la aplicación y realizar una medición de 30 segundos de la pulsera.** En el Cuadro 25 se presentan los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 25. Resultados obtenidos de la tarea no.3 en la entrevista con la aplicación

Usuario	Tiempo (seg)
1	30
2	30
3	48
4	30
5	18
6	46
7	21
8	19

d. **Tarea no. 4: Reproducir la medición de la pulsera tomada anteriormente.** En el Cuadro 26 se muestran los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 26. Resultados obtenidos de la tarea no.4 en la entrevista con la aplicación

Usuario	Tiempo (seg)
1	12
2	12
3	13
4	15
5	18
6	33
7	21
8	19

e. **Tarea no. 5: Ingresar a la aplicación y realizar una medición del casco, más alguna funcionalidad de la pulsera.** En el Cuadro 27 se muestran los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 27. Resultados obtenidos de la tarea no.5 en la entrevista con la aplicación

Usuario	Tiempo
1	19
2	31
3	31
4	24
5	26
6	12
7	40
8	32

f. **Tarea no. 6: Generar la data obtenida de alguna medición.** En el Cuadro 28 se muestran los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 28. Resultados obtenidos de la tarea no.6 en la entrevista con la aplicación

Usuario	Tiempo
1	21
2	10
3	33
4	20
5	30
6	14
7	10
8	10

g. **Tarea no.7: Lea las “Instrucciones” y el “Acerca De” de la aplicación.** En el Cuadro 29 se muestran los tiempos que cada entrevistado realizó en esta tarea.

Cuadro 29. Resultados obtenidos de la tarea no.7 en la entrevista con la aplicación

Usuario	Tiempo (seg)
1	45
2	29
3	21
4	25
5	39
6	21
7	21
8	47

h. **Tarea no.8: Utilizar la sección de “Demo” de la aplicación.** En el Cuadro 30 se muestran los tiempos obtenidos que cada entrevistado realizó en esta tarea.

Cuadro 30. Resultados obtenidos de la tarea no.8 en la entrevista con la aplicación

Usuario	Tiempo (seg)
1	14
2	16
3	16
4	15
5	17
6	19
7	19
8	10

i. **Resumen de las entrevistas de usabilidad con Mockup.** Para fines prácticos, se muestra un resumen de los resultados obtenidos en las entrevistas de Usabilidad realizadas a cada entrevistado. En el Cuadro 31 se muestra la información general recabada de los entrevistados. En ella se muestra la edad promedio de los entrevistados, así como la cantidad de hombres y mujeres entrevistadas.

Cuadro 31. Información general de los entrevistados en la entrevista con la aplicación

Información general	
Edad promedio (años)	20.5
Cantidad hombres	6
Cantidad mujeres	2

En el Cuadro 32 se muestra la cantidad de personas que pertenecen a los diferentes grados académicos que se especificaban en la encuesta pre-entrevista.

Cuadro 32. Cantidad de personas según su grado académico en la entrevista con la aplicación

Grado académico	
Diversificado	0
Universitario	8
Licenciatura	0

En el Cuadro 33 se muestran los porcentajes de la cantidad de personas que han contestado de manera afirmativa a la pregunta que se les hacía en la encuesta pre-entrevista.

Cuadro 33. Experiencia de los entrevistados en sistemas relacionados en la entrevista con la aplicación

Experiencia en sistemas relacionados	
¿Alguna vez ha oído acerca de tecnología BCI?	0%
Ha oído acerca del dispositivo EMOTIV EPOC	38%
¿Sabía que su cerebro produce señales Encefalográficas?	38%
Sabía que hay 6 emociones fundamentales	38%

En el Cuadro 34 se muestra el tiempo promedio obtenido por los entrevistados al realizar cada una de las tareas en las que se les pidió durante la entrevista.

Cuadro 34. Tiempo promedio de los entrevistados por cada tarea realizada en la entrevista con la aplicación

Tarea	Promedio (seg)
Ingresar a la aplicación y medir durante 30 seg solamente la pulsera	26.00
Reproducir la medición anterior del casco	18.88
Ingresar a la aplicación y medir durante 30 seg solamente la pulsera	30.25
Ingresar a la aplicación y medir durante 30 seg el casco más alguna medición de la pulsera	27.00
Generar la data obtenida de una medición	18.50
Reproducir la medición de la pulsera	17.90
Lea las “Instrucciones” y el “Acerca De” de la aplicación	31.00
Utilizar el Demo de la aplicación	15.75

4. Vídeo de las entrevistas. En el siguiente link se pueden observar las entrevistas grabadas que se realizaron en las pruebas de usabilidad de la aplicación:

[https://www.youtube.com/playlist?list=PLmb7D2uqbh\\_wdNuo9KvV0whkOMFc2\\_88j](https://www.youtube.com/playlist?list=PLmb7D2uqbh_wdNuo9KvV0whkOMFc2_88j)

## D. Discusión

La metodología de trabajo utilizada fue HCI. A pesar de esto, no se realizó la etapa de Benchmarking de Usabilidad ya que estas pruebas se realizan con dispositivos similares al del proyecto, pero estos dispositivos tenían un alto costo para su obtención y por esta razón no se realizó esta etapa de HCI. Las primeras entrevistas se realizaron en papel y por lo mismo les fue más difícil a las personas entrevistadas tratar de comprender la finalidad de la aplicación. Esta es la razón por la que se obtuvieron los comentarios antes mencionados en las entrevistas con el prototipo de Papel. Asimismo, se debe de tener en cuenta que las personas entrevistadas, según el Cuadro 32, en su gran mayoría contaban con un nivel de estudios universitarios. Este factor pudo haber influido en los comentarios finales e inclusive, en la interpretación que se le diera al prototipo. De la misma manera, los resultados obtenidos de las entrevistas con el prototipo de papel fueron de gran ayuda ya que al momento de realizar este prototipo no se tomaron en cuenta algunas funcionalidades que son necesarias en la Interfaz. Estos comentarios influyeron en la creación de las ventanas del Menú Principal, ventana de Instrucciones, ventana de información acerca del proyecto llamada Acerca De, así como una ventana de ayuda que explicara la información de cada botón y parte de la interfaz del proyecto. Cabe mencionar que la opinión de los entrevistados fue de gran importancia en la siguiente fase del diseño de la Interfaz de la aplicación ya que, como se ha mencionado anteriormente, la metodología de trabajo utilizada fue HCI centrada y enfocada en la opinión del usuario final de la aplicación.

En las entrevistas de Usabilidad se obtuvieron resultados muy satisfactorios ya que las personas lograron entender el propósito final de la Interfaz. Además, los entrevistados dieron comentarios muy amigables con respecto a la organización de la Interfaz, los colores de la misma y la funcionalidad final de cada ventana de la aplicación. Los comentarios obtenidos se dieron luego de que los entrevistados utilizarán la interfaz y se pudieran responder cualquier duda o sugerencia que tuvieran acerca de la misma.

En base a el Cuadro 19, la edad promedio de los entrevistados fue de 28.5 años, siendo un total de 9 hombres y 11 mujeres entrevistadas. Estos datos demuestran que las entrevistas realizadas fueron hechas a usuarios que tienen el perfil descrito del usuario final. A la misma vez, en el Cuadro 21 se pueden observar que las personas no contaban con un conocimiento amplio acerca de tecnología BCI, o relacionada a ella. Asimismo, podemos notar según la misma Cuadro que la mayoría de las personas, un 55%, sí conoce que las personas poseen 6 emociones fundamentales. Caso contrario a las demás preguntas relacionadas con esta sección ya que un porcentaje muy alto, según el Cuadro 21, muestra que las personas no tienen conocimiento sobre señales Encefalográficas o del dispositivo EMOTIV EPOC.

La primera pregunta de la encuesta post-entrevista era “¿La organización de la interfaz le parece la correcta?” y para la cual todos los entrevistados respondieron de forma afirmativa a ella. Esto indica que los cambios efectuados en la interfaz luego de realizar las entrevistas en el prototipo de papel fueron muy acertados ya que los nuevos entrevistados entendieron el orden de cada parte de la interfaz. Con respecto a la información que proveía la interfaz las personas dieron respuestas muy variadas ya que, para el 70% de los entrevistados la información que proveía la interfaz fue la necesaria, mientras que para el 30% de los

entrevistados, la información era muy poca o no lograban entender la misma. Esto pudo haber sido influido debido al nivel académico que poseían los entrevistados ya que los que se encontraban en Diversificado, cuatro personas, y los que tienen un grado en Licenciatura, dos personas, no se encuentran en una búsqueda de nuevas tecnologías día con día. Esto se debe a que en la Universidad normalmente se utilizan las últimas tecnologías que se presentan en las diferentes ramas que se imparten en la misma, y el diseño que se realizó para la interfaz de la aplicación fue hecho por un estudiante universitario.

Con la siguiente pregunta de la encuesta referente al flujo de la interfaz, el 10% de los entrevistados tuvieron algún inconveniente con el mismo y este porcentaje sugirió que se le agregara el nombre a los botones que poseía la interfaz. Referente a los colores de la interfaz, el 80% de los entrevistados no tuvieron ningún problema con ellos. El otro 20% sugirió el uso de colores más llamativos o con un tono más claro. Esta sugerencia no se tomó en cuenta ya que, con base en la psicología del color, los colores vivos o fuertes pueden inducir a un estado de felicidad, siendo este un posible factor que altere la medición que se esté realizando. Por último, los entrevistados tuvieron la oportunidad de comentar alguna mejora o problema que hayan tenido durante el uso de la aplicación. De esta última pregunta se los comentarios recibidos hacían referencia al uso de algún tipo de información que posean los botones de la aplicación.

Con base en el Cuadro 22 se puede observar que el tiempo promedio para realizar alguna funcionalidad de la aplicación no fue mayor a 30.01 segundos, obteniéndose en la tarea 5 al momento de solicitar que se realizara una medición del casco más una funcionalidad de la pulsera. Esto indicó que la aplicación es intuitiva, amigable para la vista del usuario y, asimismo, fácil de utilizar para el público en general.

En la fase final de este proyecto se desarrolló la interfaz de la aplicación en la cual se mostraba las funcionalidades que tendrá EMOTRIX. Como podemos observar en el Cuadro 34 se observa que el tiempo promedio por tarea fue mayor que el obtenido en el Cuadro 22. Esto se debe a que, en la interfaz ya terminada, se agregaron ventanas que no se encontraban en el Mockup, como, por ejemplo, las interfaces de las instrucciones de cada dispositivo, así como la obtención de gráficas al presionar cierto botón en la interfaz. Estas entrevistas se realizaron a diferentes personas, y con base en el Cuadro 32, todos eran estudiantes universitarios, con un promedio de edad de 20.5 años, así como lo indica el Cuadro 31, siendo un total de ocho personas entrevistadas de las cuales seis eran hombre y dos mujeres. En estas entrevistas el facilitador les proveyó de toda la información necesaria previa del porqué de la entrevista, así como los fines educativos que tendría la misma. El único problema que surgió en el uso de las entrevistas, y de los cuales la totalidad de los entrevistados tuvo una confusión fue al momento de realizar la tarea no.6 en la que se pedía que se generara la data de una medición antes realizada. Esto se debe a que este botón no mostraba ningún tipo de mensaje o confirmación acerca de las acciones que realizaba, dando la sensación de que no servía o no realizaba ninguna acción.

Por último, cabe mencionar que, aunque las personas no contaban con un amplio conocimiento en el área de tecnología BCI, según el Cuadro 33 en la que se muestra que sólo el 38% de los entrevistados tenían conocimientos previos, las personas no tuvieron problema en entender la justificación del proyecto.

## E. Conclusiones

1. Se utilizó la metodología de diseño HCI para la elaboración de la Interfaz exitosamente ya que se obtuvieron los resultados esperados referentes a la evaluación con usuarios finales de la aplicación.
2. Se diseñó la interfaz para EMOTRIX en la que se demuestran las funcionalidades que ofrece el proyecto.

## F. Recomendaciones

1. Se recomienda agregar algún tipo de animación o un módulo que implemente el sonido para hacer más amplio el alcance de EMOTRIX al tener más métodos para inducir las emociones.
2. Se recomienda hacer más iteraciones del proceso de HCI sobre la interfaz del proyecto para obtener más y mejores comentarios acerca de la misma.
3. Se recomienda que las entrevistas de Usabilidad con la aplicación sean realizadas en un ambiente dedicado a pruebas de este tipo.

**IX. MÓDULO 6: ADMINISTRAR LOS RECURSOS EN EL DESARROLLO DE LA PLATAFORMA DE TECNOLOGÍA BRAIN COMPUTER INTERFACE, REGISTRAR SISTEMÁTICAMENTE CON MÉTRICAS EL AVANCE DEL PROYECTO Y PROPORCIONAR LA INFORMACIÓN MEDIANTE UN ANÁLISIS DE COSTOS PARA FUNDAMENTAR LA TOMA DE DECISIONES EN LAS SIGUIENTES FASES DE EMOTRIX**

**A. Marco teórico**

1. Gestión de proyectos. Un proyecto es un conjunto de actividades no repetitivas claramente definidas para alcanzar uno o varios objetivos, en un tiempo definido y con recursos limitados. Esta definición de proyecto es general, justifica por qué las actividades del plan informático se han definido como proyectos. Además, nos ayuda a identificar y verificar cuando nos encontramos ante un proyecto, lo cual se logra mediante la relación de los objetivos con las características que definen un proyecto. (Rodríguez & William, 2006)

a. Ciclo de vida predictivo. Es de utilidad cuando el producto a entregar está definido de forma clara, o este se debe entregar de una manera específica que le dará valor a los involucrados o interesados. Por ello, el equipo del proyecto en el inicio del ciclo de vida lo más antes posible deberá enfocarse en definir el alcance y desarrollar un plan para entregar. Las actividades realizadas en cada fase serán diferentes a las fases anteriores y como requerimiento, también serán diferentes las habilidades requeridas y la composición del proyecto. (Ameijide L., 2016)

b. Ciclo de vida adaptativo. Busca la participación continua de las partes interesadas y también responder rápidamente a los altos niveles de cambio. En este método las iteraciones son muy rápidas, de duración y costes fijos. El alcance se conoce como trabajo pendiente asociado al producto y estará descompuesto en un conjunto de requisitos y trabajos a realizar. Al inicio de cada iteración se definen los pendientes que son de alta prioridad y son posibles entregar en la siguiente iteración. El cliente deberá revisar el producto al final cada iteración para garantizar que se está siguiendo el camino para el resultado esperado por el cliente y el patrocinador. (Ameijide L., 2016)

## 2. Gestión de proyectos de software

a. **Planificación informática.** Es el proceso sistematizado necesario para definir y desarrollar la estrategia tecnológica de información, que la organización debe seguir. Un proyecto informático se caracteriza por la intensa participación de recurso humano de distintas áreas durante su desarrollo. Cuando se define un proyecto también se deben aclarar los siguientes puntos (Bedini A., 2007):

- i. Cliente: A quien va dirigido el resultado del proyecto
- ii. Usuario: Quienes utilizaran el sistema o parte de este.
- iii. Término: Momento en que se cumple el resultado definido tanto en costo, oportunidad, calidad o desempeño técnico.
- iv. Costo: Recurso o insumo
- v. Tiempo: Recurso que origina una secuencia y luego un programa, es transformable en costo.
- vi. Jefe del proyecto: Encargado de la dirección del proyecto, su planificación y el control de todos los costos, recursos, programas y de la satisfacción del cliente.

b. **Metodologías ágiles.** Este tipo de metodologías son una serie de técnicas para la gestión de proyectos de desarrollo. El desarrollo ágil de software, se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones que evolucionan con el tiempo según la necesidad del proyecto. El término de Metodologías ágiles se hizo popular cuando diecisiete especialistas en desarrollo de software, presentaron los métodos Extreme Programming (XP), Scrum, Feature Driven Development (FDD), entre otros, establecieron principios comunes compartidos por todos. El resultado fue la creación de la Alianza Ágil (Agile Alliance) y el establecimiento del Manifiesto Ágil o Agile Manifiesto, en el 2004. Las metodologías ágiles varían en sus prácticas y en sus fases, sin embargo, comparten algunas características, tales como: desarrollo iterativo e incremental, comunicación y reducción de productos intermediarios y de la documentación extensiva. (Martinez, 2015)

1) **Manifiesto ágil.** El manifiesto ágil es un documento que brinda métodos para desarrollar software y se valoran (Amejjide L., 2016):

- Los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

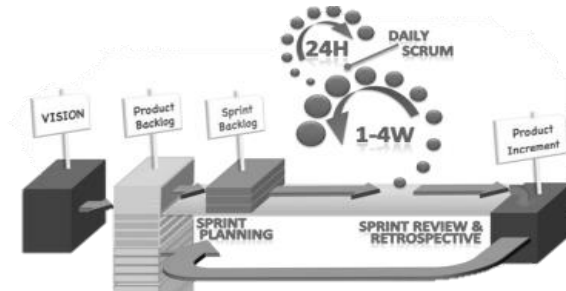
### 2) Metodologías Ágiles

1. Scrum. Sus características fundamentales son:

- Iteraciones cada mes y hasta dos semanas, proporciona un resultado completo.

- Se planifica la ejecución cada día y tiene una duración máxima de quince minutos.
- El director del proyecto se encarga de eliminar obstáculos y evaluar productividad.
- Se refina con el cliente requisitos buscando maximizar el retorno de la inversión.

Figura 178. Iteraciones de Scrum

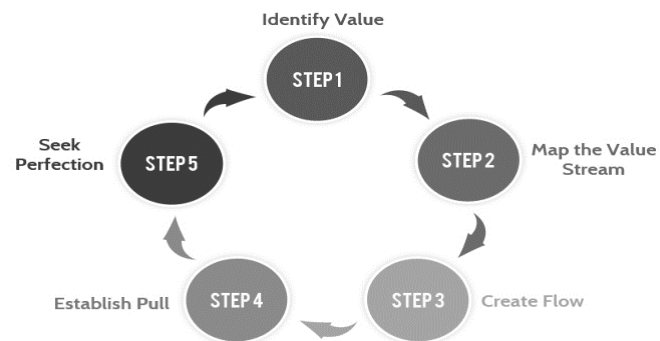


(TatvaSoft, 2015)

## 2. Lean Development. Sus características fundamentales son:

- Optimización del proceso, para desarrollar más rápido.
- Aplicación de los principios de lean de eliminación del desperdicio, identifica cualquier cosa que no añada valor al producto para eliminarla.
- Entre las actividades que no añaden valor al producto está la sobre documentación.

Figura 179. Desarrollo de Lean Development

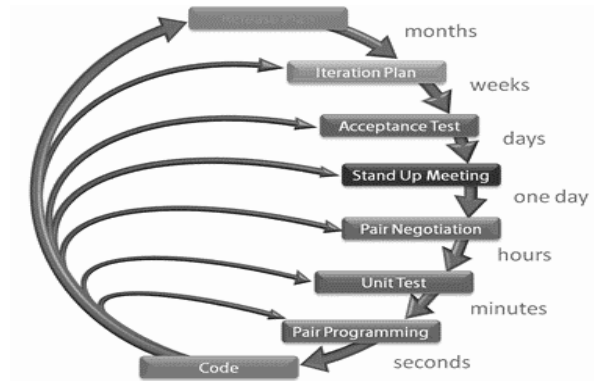


(TatvaSoft, 2015)

## 3. Extreme Programming (XP)

- Programación en parejas, debido a la discusión y revisión que se le da al código mientras es desarrollado.
- Frecuente integración del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de añadir nueva funcionalidad.

Figura 180. Planificación de Extreme Programming

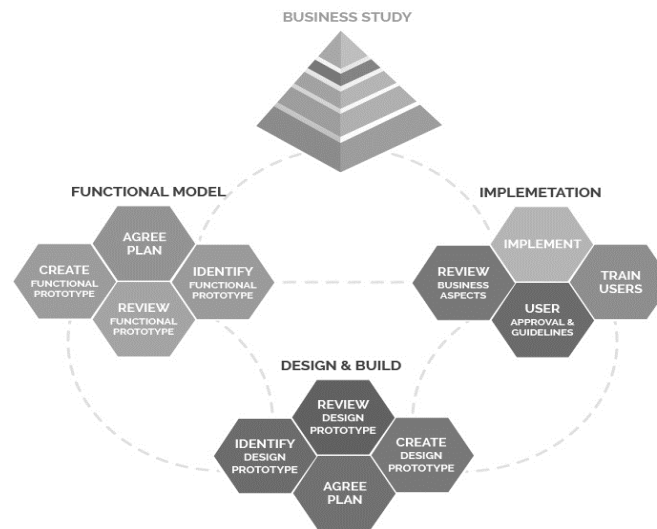


(TatvaSoft, 2015)

#### 4. Dynamic System Development Method

- Proceso iterativo e incremental
- Equipo de desarrollo y el usuario trabajan juntos.
- Fases para la evaluación del negocio y modelado funcional.
- Retroalimentación a todas las fases.

Figura 181. DSDM metodología

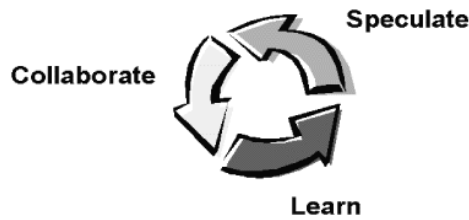


(TatvaSoft, 2015)

#### 5. Adaptive Software Development. Sus características fundamentales son:

- Orientado a los componentes del software más que a tareas del proyecto.
- Se realiza en colaboración del equipo el desarrollo de características.
- Revisión constante de componentes para retroalimentación de errores.

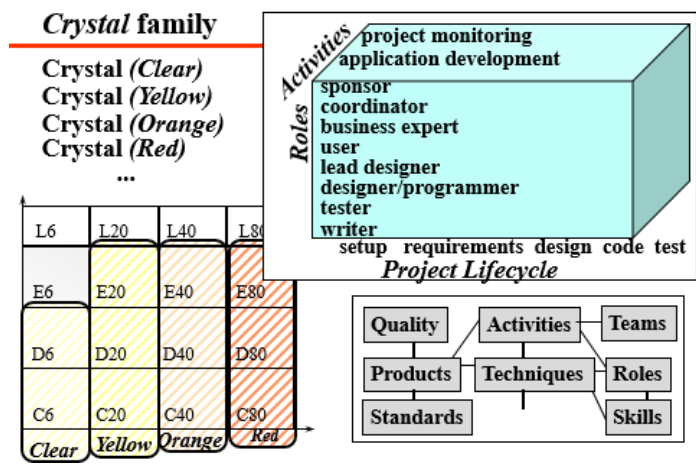
Figura 182. Desarrollo con Adaptive Software Development



(TatvaSoft, 2015)

6. Crystal Method. Son una familia de metodologías que fueron desarrollados por Alistair Cockburn a mediados de 1990. Se utiliza la palabra proveniente de la piedra preciosa Crystal, porque sus lados son una representación de técnicas, herramientas, normas y roles. Un proyecto cuando es pequeño utiliza la metodología Crystal Clear, Crystal Orange o Crystal. Si el proyecto implicaba una misión crítica se usaban los métodos Crystal Diamond o Crystal Sapphire.
  - Mindset. El desarrollo de software como recurso limitado es un juego de invenciones y comunicación.
  - Principles. Aligerar los productos del trabajo, mejorar la comunicación informal y ajustarse a circunstancias.
  - Specific. Crystal Clear, Orange, OrangeWeb. Hacer las bases y el proceso de sembrar. Las personas trabajan mejor utilizando una base que inventando algo desde cero.

Figura 183. Adecuar el método correcto de Crystal family



(TatvaSoft, 2015)

En las columnas se muestra una evaluación de parámetros para la definición del rango en el que se encuentra un proyecto: Vidas (L), Dinero Esencial (E), Dinero Discrecional (D), y Comodidad (D). Establece que pérdidas ocasionaría la falla de un sistema, en el caso de que ocasione incomodidades, según este criterio,

se encontraría en la columna Comodidad(C). Los colores en el cuadro implican la complejidad de la metodología, entre más crítico es un sistema se le asigna un color más oscuro, ya que el método es más pesado.

a. **Crystal Clear:** Hitos y productos. Secuencia de entregas, lista de metas por colaborador, diseño de bocetos y notas, modelos de objeto común, código que se ejecuta, copias de código, pruebas y manual de usuario. Se debe declarar si se tienen los requisitos suficientemente estables para diseñar, para documentar y una aplicación correcta para entregar. Políticas: Incrementos de entrega en cada  $2 \pm 1$  por mes, seguimiento de hitos y no por productos, requisitos de escenarios de uso, pruebas de regresión, revisiones de código de pares, participación directa del usuario y modelo de productos.

b. **Crystal Orange:** Son obligaciones las políticas, pero se permite la sustitución equivalente. Se acepta la tolerancia en la semántica del modelado y las sesiones. Políticas: Las entregas incrementan cada  $3 \pm 1$  por mes, seguimiento de hitos, se involucra al usuario directo, pruebas de regresión obligatorias, las actividades empiezan tan pronto como sea suficiente para discutir sobre ellas.

3. **Calendarización del proyecto.** Las fechas límites son irreales, pero también son un hecho en la vida de los desarrolladores de software. La calendarización de un proyecto de software no difiere de las demás de ingeniería multitarea. Se pueden utilizar el método de evaluación y la técnica de revisión PERT, para obtener herramientas cuantitativas que permitan determinar la ruta crítica y establecer estimaciones de tiempo. (Rodríguez N. y William M., 2006).

#### 4. Administración del riesgo

a. **Riesgos del software.** Para identificar los riesgos se debe crear una lista de verificación que se puede usar para enfocarse sobre un subconjunto de riesgos conocidos y predecibles en las siguientes subcategorías genéricas (Rodríguez José, et al., 2005):

- Tamaño del producto
- Impacto empresarial, restricciones impuestas por la administración o el mercado
- Características de los participantes, habilidades comunicativas
- Definición del proceso
- Entorno de desarrollo, riesgos asociados con la disponibilidad
- Tecnología por construir, riesgos asociados con la complejidad del sistema
- Tamaño y experiencia del personal

#### 5. Documentación de proyectos

a. **Sistemas de documentación.** El documentador debe conocer las tareas que estarán bajo su responsabilidad y las particularidades del proyecto. En las industrias de servicios es común que la documentación se realice bajo las normas del cliente y su aprobación. La dirección de la información puede influir en la dirección del usuario de la documentación, ya que, el tiempo que se lleve el implantar un sistema

podría hacer que la dirección se plantee si es necesaria la documentación o no. Las características que más influyen en la documentación del proyecto son: el tiempo del desarrollo y el equipo destinado al mismo. (Bedini A., 2007)

En el caso de un proyecto de larga duración, se exige un nivel de documentación más alto y mayor será la necesidad de un nivel adecuado de información para facilitar la comunicación, por lo tanto, estos proyectos necesitan aún más documentación. Un sistema que sólo se utilice una vez, o durante un período de tiempo muy corto, normalmente no requerirá el mismo nivel de documentación que uno que se utilice durante mucho tiempo. (Bedini A., 2007).

b. **Documentación informática.** El grado de detalle que se alcanza en estos documentos dependerá de la complejidad del producto y de la cantidad de trabajo de nuevo desarrollo que se necesite. Flujo de proceso tradicional en la utilización de la documentación (Gray y Keith R., 1976):

- Discusión formal entre el utilizador y el servicio de proceso de datos.
- Envío de una petición de utilizador por parte del usuario, preparada con la ayuda del analista de sistemas.
- Revisión inicial de la petición del utilizador y del esquema de análisis para resolver el problema.
- Elaboración de la propuesta del sistema y valoración de las posibilidades de realización del proyecto.
- Informe analítico basado en el análisis del sistema y cubriendo el área especificada en la petición del utilizador.

Las metodologías ágiles de desarrollo requieren que su documentación de manera general implique: el catálogo de requerimientos, diccionario de datos, diagramas UML, el plan de trabajo, el plan de implementación o implantación, el plan de capacitación y el control de cambios. El análisis del sistema de documentación y el proceso de diseño pueden definirse en función de los documentos de entrada recibidos y los documentos que se producen. La producción de la documentación debe ser planeada cuidadosamente tanto como la forma de presentar el material. (Bedini A., 2007).

## 6. Mediciones en producto y proceso de software

a. **Principios de medición.** Las métricas serán útiles cuando se caractericen efectivamente y se validen. Para esto, deberá tener propiedades matemáticas en una escala racional, es decir, en un rango significativo por ejemplo de 0 a 1, donde 0 representa ausencia, 1 representa valor máximo y 0.5 representa el punto medio. Cuando la métrica representa una característica que aumenta con rasgos positivos y disminuye con rasgos indeseables, el valor de la métrica aumentará y disminuirá de la misma forma. Cada métrica debe validarse de manera experimental antes de utilizarse para la toma de decisiones. (Bedini A., 2007).

b. **Modelos de costeo y desarrollo de software.** En la estimación de costo y esfuerzo del software existen distintas variables que pueden afectar el costo final de un proyecto, tales como las variables humanas, técnicas, ambientales, políticas. Es de importancia saber que la estimación de costo del software será tan buena como los datos históricos usados para generar la estimación. Sin embargo, la estimación puede darse en una serie de pasos sistemáticos con un riesgo aceptable. En proyectos de software las “estimaciones de coste” son valoraciones, con un cierto error, del esfuerzo esperado para el desarrollo del proyecto y de los plazos de tiempo requeridos para completarlo. Los componentes principales de costos son Hardware, Entrenamiento y Esfuerzo. Para utilizar un modelo de costeo es necesario saber la magnitud del producto final, midiéndolo por el tamaño del proyecto en líneas de código principalmente (Bedini A., 2007):

Cuadro 35. Dificultad de un proyecto en función de sus líneas de código

Categoría	Programadores	Duración	Líneas de Código	Ejemplo
Trivial	1	0 – 4 sem	< 1k	Utilidad de ordenación
Pequeño	1	1 – 6 sem	1k – 3k	Biblioteca de funciones
Media	2 – 5	0.5 – 2 año	3k – 50k	Compilador
Grande	5 – 20	2 – 3 año	50k – 100k	Sistema Operativo
Muy Grande	100 – 1000	4 – 5 año	100k – 1 M	Grande SO
Gigante	1000– 5000	5 – 10 año	>1M	Sistema de Distribución

(Bedini, 2007)

### c. Modelos para estimaciones

1. Modelo COCOMO. Estos modelos están definidos para tres modos de desarrollo (Bedini A., 2007):

- Proyectos orgánicos: El equipo de desarrollo es pequeño y experimentado, en un ambiente familiar y con aplicaciones conocidas. En general no manejan más de 50,000 Líneas de Código. El esfuerzo se define por:

$$E_D = 2,4 \times (KLDC)^{1,05}$$

- Proyectos semiconectados: El equipo de desarrollo está conformado por personal experimentado y personas con poca experiencia, tienen interacción con otros sistemas y sus líneas de código no sobrepasan las 300,000. El esfuerzo se define por:

$$E_D = 3,0 \times$$

$$(KLDC)^{1,12}$$

- Proyectos integrados: Un equipo grande de desarrollo poco experimentado y corresponden a proyectos que presentan un fuerte acoplamiento entre el Hardware, Software y los procedimientos operacionales. El esfuerzo se define por:

$$E_D = 3,6 \times (KLDC)^{1,20}$$

Donde **KLDC** son kilo líneas de código.

Cuadro 36. Ecuaciones básicas del modelo integrado

Estimación	Proyecto integrado
Esfuerzo estimado por mes	$E_D = 3.6 \times (KLDC)^{1.20}$
Tiempo de desarrollo en meses	$T_D = 2.5 \times (E_D)^{0.32}$
Productividad	$P_R = LDC \times (E_D)$
Número de personas en promedio	$P = \frac{E_D}{T_D}$

(Bedini, 2007)

El número promedio de personas resultado en el Cuadro 36 regularmente el número de personas empleadas varía a lo largo del desarrollo del proyecto, por lo tanto, se debe considerar únicamente como una aproximación y no se debe tomar como una cifra significativa. Las constantes añadidas para el Modelo Básico para un proyecto integrado permiten la estimación más acertada y teórica de la realización de un proyecto.

Los modelos COCOMO manejan una jerarquía que se da según el nivel de detalle que se necesite, estas estimaciones de esfuerzo son expresadas en hombre-mes, considerando que la dedicación de una persona es de 152 horas al mes (Bedini, 2007):

- Modelo COCOMO básico: es un modelo univariable estático que calcula el esfuerzo y el tiempo del desarrollo de software en función del tamaño del programa, expresado en Líneas de Código (LOC).
- Modelo COCOMO intermedio: es un modelo univariable estático que calcula el esfuerzo del desarrollo de software en función del tamaño del programa y otros aspectos considerados como “conductores de coste” como la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto. A estos atributos se les identifica el impacto que les produce en entorno del desarrollo del proyecto. Entre los que menciona Boehm se encuentra:
  - Rely: Consecuencia para el usuario de los defectos del producto y sus niveles de efectos de los fallos.
  - CPLX: Si el módulo se compone de expresiones matemáticas que utilizan muchos recursos de tiempo en planificación.
  - MODP: Las prácticas modernas de programación utilizadas.
- Modelo COCOMO avanzado (o detallado): es un modelo que incorpora todas las características de la versión intermedia y realiza una evaluación del impacto de los conductores de coste en cada fase desde su análisis, diseño, etc., del proceso en el proyecto. Este modelo implica dos características principales:
  - Multiplicadores de esfuerzo sensitivos a la fase: Los multiplicadores de este modelo se asignan según el esfuerzo representado por los atributos. Es útil para la asignación de los recursos de personal para cada fase del proyecto.

- Jerarquía del producto a tres niveles: Se asigna una cuantificación según los niveles de susceptibilidad entre el módulo, subsistema y sistema.

## B. Marco metodológico

Debido a que el proyecto implicó distintos recursos para el desarrollo de la plataforma, el recurso humano fue una de los principales y también los recursos tecnológicos y herramientas para la ejecución y finalización del proyecto. Con el apoyo del Coordinador del proyecto, quién representó a la Universidad del Valle de Guatemala y actuó como el cliente y el equipo de Emotrix para el desarrollo del Trabajo de Graduación.

El equipo estaba conformado por una estudiante de Ingeniería Industrial, encargada de la administración, métricas y estimaciones de costos, un estudiante de Ingeniería Mecatrónica, encargado del procesamiento y clasificación de señales para la interfaz Brazo Computador, un estudiante de Ingeniería Electrónica, encargado del procesamiento de señales encefalográficas para interfaz Cerebro Computador, tres estudiantes de Ingeniería en Ciencias de la Computación y Tecnologías de la Información, encargados del desarrollo del Software Development Kit para el procesamiento de las señales, recepción y almacenamiento de los datos del casco y la pulsera, y la implementación de la plataforma Figura para Emotrix.

El Megaproyecto inició en enero de 2016 y finalizó en noviembre de 2016 y las cinco etapas del proyecto fueron las siguientes:

1. **Iniciación.** Se inició concibiendo la oportunidad de desarrollar una plataforma BCI enfocada en la identificación y manipulación de emociones humanas que fuera *open source* y más económicas que las existentes, y explorando su potencial para ser realizado como Trabajo de Graduación, también comprendiendo las características generales y específicas sobre el tipo de plataforma que se debía desarrollar y las herramientas y materiales que se necesitarían. Así como se investigó sobre las propuestas actuales en el mercado, su potencial en distintas áreas de investigación y se escudriñó información sobre los recursos y competencias que serían necesarios para la ejecución de cada actividad. En base a esto, se propusieron los roles que debían desempeñarse en el equipo para la ejecución de la plataforma.

2. **Planificación.** Se definió el objetivo principal de cada rol para lograr el éxito del desarrollo de la plataforma. Se establecieron los requisitos del proyecto para que se permitiera definir los entregables de cada miembro del equipo y se plasmó en un Gantt. Se estableció la metodología Crystal Clear, la cual se adaptó mejor a las características y condiciones de desarrollo de Emotrix ya que permitió que las entregas incrementales se dieran 2 por mes y el seguimiento fue en relación a los hitos establecidos y no a los productos.

3. **Ejecución.** Se llevaron a cabo reuniones de forma recurrente apegándose a la metodología Crystal Clear, para la revisión de requerimientos y determinación de los nuevos objetivos. Se realizó la estimación concerniente al costo del proyecto final, esta medida permitió visualizar el esfuerzo que se llevaría el desarrollo de la plataforma, para esto se definió la estimación que es más adaptable a este tipo de proyectos según sus criterios y características, Emotrix utilizó una estimación Constructiva de Costos Básico para un equipo Integrado. Se realizó la documentación de los entregables con sus respectivas métricas del avance del proyecto.

4. **Seguimiento y control.** Se empleó la herramienta en línea Asana (<http://asana.com/>) para el seguimiento, asignación y coordinación al trabajo en función de los objetivos alcanzados. También se utilizó la plataforma de desarrollo colaborativo de software GitHub, la cual permitió hacer un repositorio del código y ayudó a visualizar la frecuencia de desarrollo de código. La comunicación se mantuvo a través del grupo de Whatsapp del equipo de trabajo. Las reuniones fueron presenciales y vía Hangouts de Google en cada una se llevaron minutas para la determinación de resultados de las estrategias utilizadas, fallos y avances evaluados con las métricas.

5. **Cierre.** El cierre del proyecto se realizó con las pruebas finales de la implementación y unificación de todos los módulos y prototipos del Megaproyecto Emotrix. La fase de cierre inicia con la entrega de los reportes individuales en los cuales se recopiló el desarrollo del proyecto y el reporte consolidado del equipo. En esta fase se oficializaron los alcances finalizados del proyecto.

## C. Resultados

1. **Administración de Emotrix.** La organización y gestión del proyecto se realizó aplicando e integrando la metodología Crystal Clear para el desarrollo de proyectos de software. Para poder visualizar y darles seguimiento a los hitos y al avance del proyecto se generó un Gantt general que incluye los seis módulos en los cuales está dividido el proyecto. Para generar el Gantt se definió como entrega final la Plataforma desarrollada, el casco de electrodos, la pulsera, la comunicación entre hardware y software, el seguimiento y las estimaciones, y la plataforma Figura, a partir de esto se definieron las entregas cada  $2 \pm 1$  por mes en la etapa de ejecución.

Se desarrolló un plan para la gestión del proyecto el contenía distintas actividades necesarias para la preparación e integración del mismo.

Estimación de recursos necesarios para el desarrollo de cada actividad implícita para la realización del proyecto.



El Gantt detallado en cada etapa y con todas las tareas de los integrantes del equipo de Emotrix se puede encontrar en los anexos, así como los Gantt definidos para cada actividad designada a cada miembro del equipo, los cuales tomaron como línea guía el Gantt ilustrado en el Cuadro 38. Ver en anexos el seguimiento y control al equipo desarrollado vía GitHub en la parte de “Frecuencia de códigos”.

## 2. Estimación de costos

Cuadro 39. Selección del modelo constructivo de costos

Categoría	Programador	Duración	LDC	Desarrollado (Ejemplo)
Pequeño	1	1-6 mes	1k – 3k	Biblioteca de funciones
Media	2 – 5	0.5 – 2 años	3k -50k	Compilador

Se estimó en la etapa de iniciación la cantidad de líneas de código que tendría la plataforma según historiales de desarrollo del tipo de plataforma y la duración que se podría tener, para poder categorizar el proyecto y hacer estimaciones más acertadas sobre el esfuerzo y el tiempo realmente necesario para el desarrollo del proyecto y así tener métricas representativas del desarrollo del proyecto para la toma de decisiones en cada seguimiento del mismo. En este caso la columna representativa para la selección del modelo es el tamaño del desarrollo debido a que este cuadro representa un trabajo que se realiza a tiempo completo en un total de 152 horas al mes. Por lo tanto, Emotrix se categoriza como pequeño, aunque el período de duración y número de programadores es distinto, se adaptó así debido a la división de tareas para lograr la finalización de los objetivos planteados del mismo. Inclusive se seleccionó como un proyecto integrado debido al acoplamiento que existió entre el software y hardware, además de ser un equipo de 3 programadores poco experimentados.

Cuadro 40. Estimaciones del modelo constructivo de costos básico

Estimación de Emotrix	Proyecto integrado	Resultados
Esfuerzo estimado por mes	$E_D = 3.6 \times (2.5KDC)^{1.20}$	10.81
Tiempo de desarrollo en meses	$T_D = 2.5 \times (10.81)^{0.32}$	5.35
Programadores en promedio	$P = \frac{13.45}{5.74}$	2.01

Según el modelo constructivo de costos básico para proyectos integrados del Cuadro 40 el tiempo de desarrollo sería un aproximado de 5.3 meses y con dos programadores en promedio, esta estimación implica que el programador haya desarrollado las líneas de código todos los días hábiles durante un mes de trabajo a tiempo completo de siete horas diarias. Este tiempo se considera para un trabajo en el cual los miembros del equipo del proyecto trabajan en paralelo.

Cuadro 41. Tiempo de desarrollo estimado total

Horas totales para finalizar desarrollo	Horas reales utilizadas
$5.35 \text{ mes} \times \frac{152 \text{ hora}}{\text{mes}} = 813 \text{ hora}$	$813 \frac{\text{hora}}{\text{persona}} \times 2 \text{ persona} = 1626 \text{ horas}$

## 1) Salarios de desarrolladores en Guatemala

Figura 184. Salarios para programadores de aplicaciones sin experiencia en Guatemala

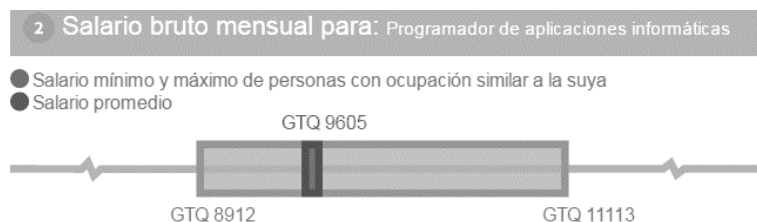
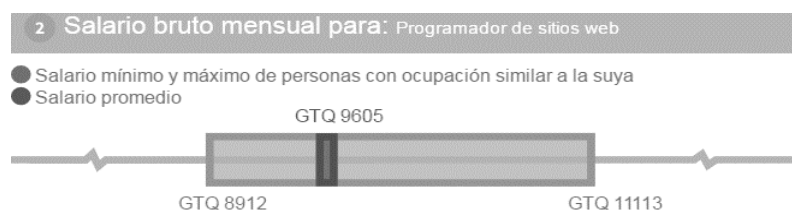


Figura 185. Salarios para programador de sitios web sin experiencia en Guatemala



(WageIndicator, 2016)

Según las Figuras 184 y 185, el sueldo promedio de un programador de distintas índoles en Guatemala es de GTQ 9,605.00.

Cuadro 42. Tipo de cambio de (Dólares de EE.UU. a Quetzales de enero de 2016

Moneda	TCR/
Dólares de EE.UU.	7. 63237 Q

(Guatemala, 2016)

Cuadro 43. Costo de mano de obra del desarrollo de la plataforma de Emotrix

Horas regulares de trabajo en Guatemala	Sueldo por hora regular en Guatemala	Costos resultantes del desarrollo de Emotrix
$22 \text{ día hábil} \times \frac{8 \text{ hora}}{\text{día hábil}}$ $= 176 \text{ hora}$	$\frac{Q9,605.00}{\text{mes}} \times \frac{1 \text{ mes}}{176 \text{ hora}}$ $= \frac{Q 54.57}{\text{hora}}$	$1626 \text{ hora} \times \frac{Q 54.57}{\text{hora}}$ $= \mathbf{Q88,730.82}$

Según el Cuadro 43 la estimación de los costos de mano de obra del desarrollo de Emotrix, cada programador con un sueldo de Q 54.57 la hora, ascienden a Q 88, 730.82 en total.

Cuadro 44. Costo de materiales utilizados en el desarrollo de prototipos de Emotrix

<b>Recurso o elemento</b>	<b>Beneficiario del recurso</b>	<b>Precio</b>
Material PLA para estructura	casco Emotrix	Q 152.65
Microcontrolador	casco Emotrix	Q 53.42
Amplificadores operacionales	casco Emotrix	Q 73.27
Resistencias	casco Emotrix	Q15.26
Capacitores	casco Emotrix	Q 7.63
Potenciómetros	casco Emotrix	Q30.52
Módulo UART	casco Emotrix	Q 61.05
Cable mini USB a USB	casco Emotrix	Q 76.32
Electrodos	casco Emotrix	Q 41.97
Electrodos para pruebas	casco Emotiv	Q 381.61
Tachuelas	pulsera o Brazalete	Q 5.00
Elástico	pulsera o Brazalete	Q 5.00
Velcro	pulsera o Brazalete	Q 10.00
<b>Total</b>		<b>Q 1,425.88</b>

Según el Cuadro 44 El total en materiales y recursos utilizados en Emotrix asciende a Q1,425.88, incluyendo los recursos utilizados para hacer pruebas de funcionamiento del Software Development Kit desarrollado con el casco del Emotiv Epc el cual le pertenece a la Universidad del Valle de Guatemala.

Cuadro 45. Consolidación de los costos incurridos y estimados para Emotrix

Costos de mano de obra	Q 88,730.82
Costos de materiales y requerimientos	Q 1,425.88
<b>Total</b>	<b>Q 90,156.7</b>

En el Cuadro 10 Se reflejan los costos estimados para el desarrollo de la plataforma de Emotrix y los costos de todos los materiales utilizados para los prototipos y pruebas del equipo.

### 3. Manejo de riesgos

a. Proyección del riesgo. Se identificaron distintos riesgos que podrían obstaculizar el cierre del proyecto, se identificó su impacto en el caso ocurriesen en el transcurso de la ejecución del desarrollo de la plataforma. Para ello se categorizaron y ponderaron de forma equitativa los cuatro componentes que sufren el impacto de los riesgos y son necesarios tomar en cuenta para la ejecución de Emotrix (rendimiento, costo, calendarización y alcance). Fue definida una ponderación equitativa debido a que tanto el coordinador, como

el equipo no priorizó los componentes y los objetivos específicos de todos los módulos son de igual importancia. En el caso de la probabilidad de ocurrencia se asignó una escala cualitativa entre *Moderada* y *Alta*, debido a que las formas teóricas de evaluar estos riesgos están orientados a otro tipo de proyectos más extensos, así que el equipo llegó a este consenso por sus propias experiencias en otros proyectos.

Cuadro 46. Identificación, probabilidad y ponderación de los riesgos

Riesgo	Probabilidad	Ponderación de impacto
No contar con todo el equipo en momentos críticos	Moderada	50%
Falta de materiales o atrasos en los pedidos	Moderada	75%
Rediseñar	Alta	50%
Subestimar el tiempo de desarrollo de la plataforma	Alta	75%
No contar con pruebas para prototipos	Moderada	50%

Cuadro 47. Plan de respuesta a los riesgos de desarrollo

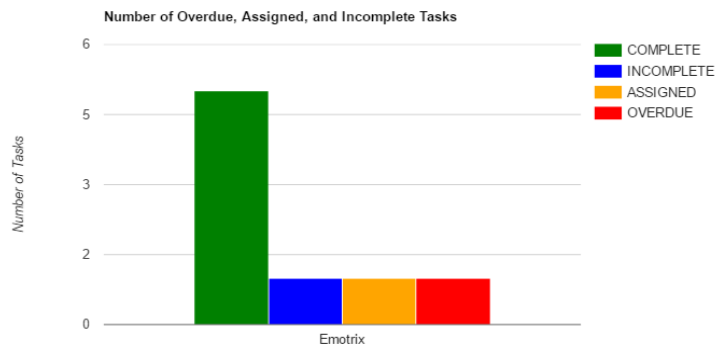
Riesgo	Respuesta
No contar con todo el equipo en momentos críticos	Coordinar del recurso humano del proyecto mediante una metodología ágil de desarrollo que implique una comunicación fluida entre el equipo y que se caracterice por un seguimiento de actividades individuales.
Falta de materiales o atrasos en los pedidos	Gestionar el equipo y los recursos permitiendo una metodología que se caracterice por recursos limitados y permita una comunicación automática dentro del equipo para establecer el tiempo requerido para hacer pedidos.
Rediseñar	Coordinar los entregables del proyecto por medio de una metodología que se enfoque en el desarrollo de software y sus constantes iteraciones y cambios en requerimientos por el cliente o directos del proyecto.
Subestimar el tiempo de desarrollo de la plataforma	Gestionar mediante una metodología ágil que permita la organización por medio de entregables, seguimientos de hitos de tal forma que los retrasos causados no sean mayores a los que sean permisibles para el proyecto.
No contar con pruebas para prototipos	Organizar por medio de una metodología ágil adaptada al tipo del proyecto, que permita un desarrollo iterativo corto que permitan varias entregas en promedio $2 \pm 1$ por mes.

4. **Métricas.** Se analizaron los módulos según su avance en relación a lo establecido por cada Gantt con el propósito de darle un seguimiento a la secuencia de entregas con respecto al tiempo que se había asignado, esta medición se realizó desde el punto de vista del desarrollador, el coordinador y el administrador del proyecto en el contexto del seguimiento de hitos.

Cuadro 48. Definición de las métricas para cada módulo

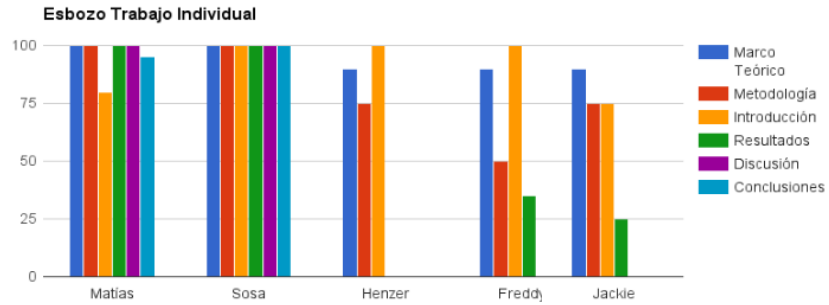
Módulo	Objetivo
Desarrollo de SDK enfocado en el procesamiento de señales EEG para la clasificación y el reconocimiento de emociones	Desarrollo de plataforma
Diseño e implementación de la plataforma Figura para Emotrix	Desarrollo del GUI
Procesamiento y clasificación de señales para interfaz brazo - computadora	Procesar y clasificar las señales humano - computador
Comunicación con casco y pulsera para recepción y almacenamiento de datos	Comunicación entre el software y hardware
Procesamiento de señales encefalográficas para interfaz cerebro - computador	Procesar y clasificar las señales cerebro - computador
Administrar los recursos en el desarrollo de la plataforma de tecnología Brain Computer Interface	Gestión, estimaciones y documentación para la entrega final de Emotrix

Figura 186. Métricas extraídas de Asana



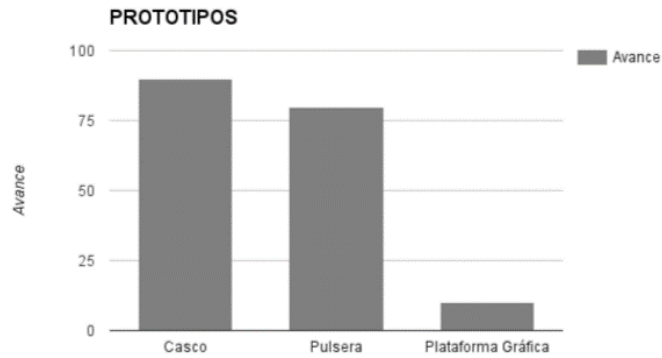
En la Figura 186 se muestran las métricas que se extraían de Asana, el cual era alimentado por cada miembro del equipo según las tareas que ya les había asignado el Administrador del proyecto o el coordinador. El color verde representaba las tareas que ya estaban completas, el azul las incompletas, el naranja tareas ya tenía asignadas y el rojo crítico que representa que tiene actividades tardías.

Figura 187. Métricas modificables en tiempo real extraídas del 5 de octubre



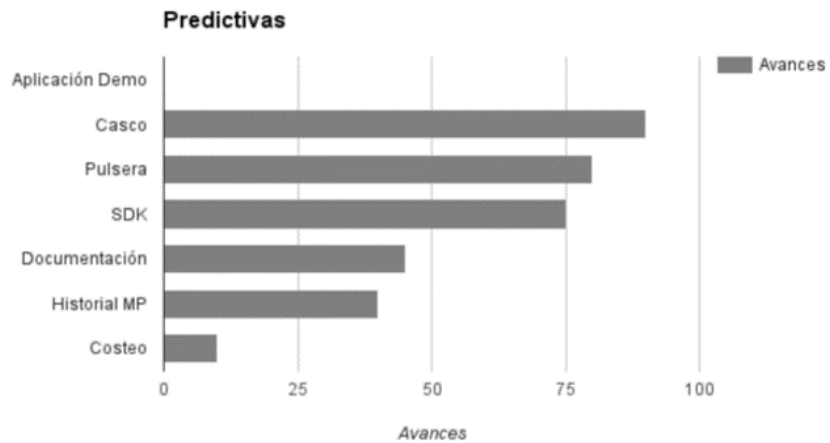
La Figura 187 muestra el estado del esbozo del trabajo individual y los porcentajes que faltaban por completar de cada parte del reporte individual del proyecto.

Figura 188. Desarrollo de métricas de los prototipos



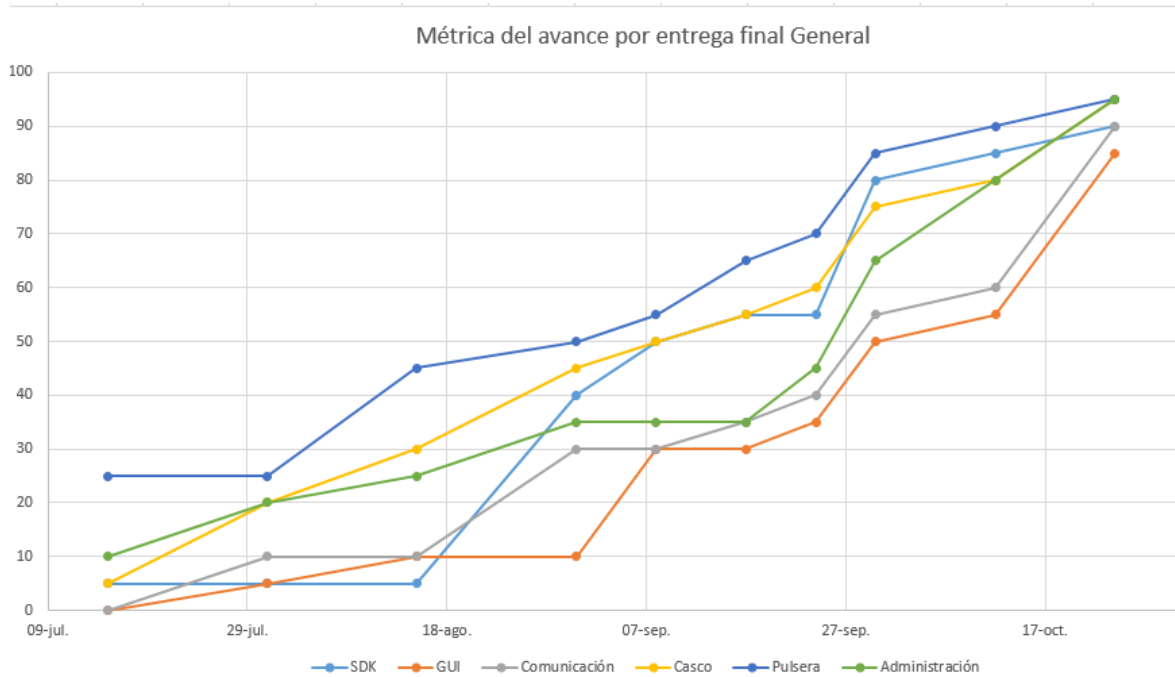
La Figura 188 refleja el estado de los prototipos que tendrán interacción con el usuario, como ejemplo del seguimiento de estas actividades.

Figura 189. Métricas de los avances del proyecto por objetivos



La Figura 189 Muestra una de las gráficas que se le mostraba al coordinador del proyector para visualizar más fácil los avances de cada módulo y fase del proyecto. El 100% representó la meta final que se estableció al inicio en el Gantt.

Figura 190. Detalle de los avances hasta el 20 de octubre



Según la Figura 190 hasta el 20 de octubre no se había dado por finalizada la plataforma debido a que aún no se había integrado todos los módulos, como un solo objetivo, el cual buscaba ver el funcionamiento completo de Emotrix, hasta esa fecha se finalizaron los objetivos generales por módulo por lo tanto se considera como el cierre general del proyecto, el cierre completo de Emotrix se dio con las pruebas finales.

## D. Conclusiones

1. El presente trabajo de graduación demostró que la metodología implementada fue la más adecuada para la administración de los recursos en el desarrollo de Emotrix ya que, los registros sistemáticos como las estimaciones de costos permitieron el seguimiento del desarrollo y facilitaron la toma de decisiones en la implementación de la plataforma.
2. Se pudo crear e implementar una estructura estándar para la gestión de proyecto para el desarrollo de Emotrix, haciendo uso de la metodología ágil Crystal Clear, que facilitó la gestión de proyectos de desarrollo de software para llegar a la etapa de cierre o finalización tuvo el seguimiento vía Asana.
3. Se registraron y documentaron las métricas de las actividades desarrolladas y facilitaron la toma de decisiones en la etapa de ejecución del proyecto.
4. Se determinó el método para la estimación de costos con el modelo constructivo de costos y se adaptó el modelo que tenía los criterios y parámetros de Emotrix.

## E. Recomendaciones

1. Se recomienda que en las siguientes fases se realice una estimación previa de esfuerzo y tiempo para poder determinar si los recursos disponibles serán los necesarios para la finalización de las mismas.
2. Continuar con otra fase del Megaproyecto en la Universidad del Valle de Guatemala partiendo de los resultados finales que se obtuvieron del Trabajo de Graduación “Emotrix: plataforma para el desarrollo de tecnología BCI enfocada en la identificación y manipulación de emociones humanas.” Se podrá implementar la identificación de más emociones y también la creación de estímulos para obtener las respuestas de las mismas.
3. Se recomienda utilizar las estimaciones de costos de este Megaproyecto y adaptarlo a las necesidades que pueda surgir en sus siguientes fases y a partir de esto se recomienda iniciar con el plan de mercadeo.
4. Se recomienda una inclusión de estudiantes del Departamento de Psicología desde el inicio del Megaproyecto o de Asesoría de parte del departamento mencionado, debido a que ellos tienen el conocimiento preciso o el *know how* para entender el funcionamiento y estímulos que resulta en una emoción.

## X. CONCLUSIONES GENERALES

- A. El mejor porcentaje de clasificación, cuando se comparó: tristeza y felicidad fue del 74%. Lo cual se logró implementando la densidad espectral de potencia (PSD), una ventana con traslape de 256 muestras (dos segundos) y un filtro que permitiese solo la utilización de aquellas muestras con una calidad en la lectura mayor al 70%.
- B. Dada la información, se logró demostrar patrones en las señales generadas por los músculos del antebrazo y clasificarlas en las distintas posiciones por medio de transformaciones al dominio de frecuencia.
- C. Se permitió a través de los dispositivos de entrada y el API de Emotrix, verificar la conexión con los dispositivos, verificar la calidad de la señal, leer los datos que estos transmiten hacia el puerto USB y almacenar la información en una base de datos MongoDB. Las funciones de conectividad que provee, permite la comunicación con cualquier dispositivo que siga el protocolo de Emotrix.
- D. La metodología implementada para la administración del desarrollo de la plataforma Emotrix se adecuó a las necesidades que implicó la gestión del proyecto, las estimaciones realizadas permitieron el seguimiento del proyecto y la estructura estándar del seguimiento permitió su documentación y métricas del proyecto.

## **XI. RECOMENDACIONES GENERALES**

- A. Se recomienda utilizar estímulos estandarizados y validados por una mayor cantidad de personas, como por ejemplo el IAPS (International affective picture system) y el IADS (International digitized sounds) ampliamente utilizados en este tipo de investigaciones. Así como un entorno mucho más controlado durante la toma de datos, para el entrenamiento del algoritmo de clasificación.
- B. Realizar pruebas con más de un individuo para asegurarse que el dispositivo funcione de forma general y no solo con una persona en específico.
- C. Para mejorar la velocidad de comunicación entre los dispositivos de entrada y el API de Emotrix, se recomienda transmitir únicamente los valores de lectura separados por comas y con algún carácter especial de finalización que indique el fin de una muestra. El proceso actual que realiza el casco convirtiendo los valores de lectura en caracteres ASCII agrega complejidad al algoritmo de lectura ya que necesita decodificar dichos caracteres para luego obtener el valor real de la lectura.

## XII. BIBLIOGRAFÍA

1. Arduino. (2016). *Arduino Nano*. Obtenido de <https://www.arduino.cc/en/Main/ArduinoBoardNano>
2. Arduino. (2016). *What is Arduino?* Obtenido de <https://www.arduino.cc/en/Guide/Introduction>
3. Baraniuk, R. (2003). *Singals ans Systems*. Houston Tx: The Connexions Project.
4. Barea, R. (2002). *Instrumentacion Biomedica: Electroencefalografia*. España: Universidad de alcala.
5. Bauer, J. (1996). *An Introduction to Python*. Linux Journal.
6. Bedini, A. (2007). *Gestión de proyectos de software*. Santiago: Departamento de Informático de Universidad Técnica Federico Santa María.
7. Boxwell JR., R. (1995). *Benchmarking para competir con ventaja*. Madrid: Ed.McGraw Hill.
8. Bradley, M. a. (2007). *The International Affective Digitized Sounds (IADS-2): Affective Ratings of Sounds and Instruction Manual*. Gainvesville: University of Florida.
9. Bradley, P., & Lang, P. a. (2005). *International Affective Picture System (IAPS): Digitized Photographs, Instruction Manual and Affective Ratings*. Gainasville: University of Florida.
10. BrainWave Laboratories. (s.f.). *Las ondas cerebrales*. Recuperado el 19 de Septiembre de 2016, de BrainWave: <https://brainwavelaboratories.com/las-ondas-cerebrales-vision-general/>
11. Brownlee, J. (03 de 08 de 2016). *Machine Learning Mastery*. Obtenido de <http://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/>
12. Comunidad. (2012). *Introduction to User-Centered Design*. Obtenido de Usability First: <http://www.usabilityfirst.com/about-usability/introduction-to-user-centered-design/>
13. Comunidad. (13 de Noviembre de 2013). *Usability Testing*. Obtenido de usability.gov: <https://www.usability.gov/how-to-and-tools/methods/usability-testing.html#>
14. Comunidad. (2014). *Definición de Aplicación*. Obtenido de ConceptoDefinicion.DE: <http://conceptodefinition.de/aplicacion/>
15. Cortés, J. C. (2007). *Del análisis de Fourier al análisis Wavelets: transformada continua Wavelet (TCW)*. Scientia et Technica.
16. Creative Commons. (2016). *Sobre las licencias*. Obtenido de <https://creativecommons.org/licenses/?lang=es>
17. Drake, J. M. (2005). *Instrumentacion electrónica de comunicaciones* . España: Universidad de Cantabria.
18. Dunn, R. (1990). *What is wxPython*. Obtenido de wxpython.org: <https://wxpython.org/what.php>
19. EEGinfo. (s.f.). *EEG electrodos y votaje galvánico*. Recuperado el 18 de Septiembre de 2016, de eeginfo: <http://www.eeginfo-neurofeedback.es/neurofeedback/support/informacion-tecnica-faq/eeg-electrodos-y-voltaje-galvanico.html>
20. Escobar, M. (2006). *Sistema Nervioso*. Cali: Editorial Universidad del Valle.

21. Eveliux. (s.f.). *Conversión Análogo-Digital*. Recuperado el 2 de Octubre de 2016, de eveliux: <http://www.eveliux.com/mx/Conversion-Analogico-Digital-ADC.html>
22. Francisco, T. L. (2013). *Administración de proyectos de informática*. Santiago: ECOE.
23. Fredrickson, B. L. (2001). *Positive Emotions*. Nueva York: Guilford Press.
24. Fredrickson, B., & Levenson, R. (2001). *The Role of Positive Emotions in Positive Psychology*. Nueva York: Guilford Press.
25. Friesen, P. E. (1982). "Measuring facial movement with the facial action coding system," in *Emotion in the Human Face*. New York: Cambridge University Press.
26. Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., . . . Stanley, H. E. (2000). Physiobank: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*, e215 -- e220.
27. Goleman, D. (1993). *La inteligencia Emocional*. Barcelona: Kairos. Obtenido de <http://www.psicoactiva.com/emocion.htm>.
28. Gonzáles, E. (2011). Megaproyecto INNOVA. Módulo de Interfaz Cerebro - Máquina. Control no convencional e implementación de un prototipo de una silla de ruedas eléctrica a través de la electroencefalografía. *Universidad del Valle de Guatemala*.
29. Gonzáles, V. (2015). *Muestreo de señales*. Recuperado el 2 de Octubre de 2016, de UAL: <http://www.ual.es/~vruiz/Docencia/Apuntes/Signals/Sampling/index.html>
30. Green, P. (4 de Febrero de 2008). Iterative Design. *Industrial and Operations Engineering*, pág. 436.
31. Guatemala, B. d. (10 de septiembre de 2016). *Tipo de Cambio*. Obtenido de Banguat: <http://banguat.gob.gt>
32. Gunn, S. (1998). *Support Vector Machine for classification and regression*. Houston: IEEE.
33. Guohua L., B. J. (2009). *Msol*. (ElSavier) Recuperado el 2 de Agosto de 2016, de <http://msol.people.uic.edu/ECE516/papers/Removing%>
34. Gupta, H. (2013). Power Spectrum using Welch Method for various Window Techniques. *International Journal of Scientific Research & Technology*, 389-392.
35. Hermes J., F. B. (2006). *Seniam*. Recuperado el 28 de Junio de 2016, de <http://www.seniam.org/>
36. Horlings, R. (2008). *Emotion Recognition Using Brain Activity, Department of Mediamatics*. Delft University of Technology.
37. Instituto nacional del cáncer. (s.f.). *Diccionario del cáncer:Electrodo*. Recuperado el 18 de Septiembre de 2016, de [Diccionario del cáncer: https://www.cancer.gov/espanol/publicaciones/diccionario?cdrid=454755](https://www.cancer.gov/espanol/publicaciones/diccionario?cdrid=454755)
38. Johnson, S. (2011). *Notes on FFT-based differentiation*. Massachusetts.
39. Jones, M. (2011). *Introduction to HCI*. Obtenido de School of Computer Science University of Birmingham: <https://www.cs.bham.ac.uk/~rx/Teaching/HCI%20II/intro.html>

40. Jurcak, V., & Tsuzuki, D. (2005). *10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning system*. Japon: Sensory and Cognitive Food science Laboratory.
41. Jurican, J. (s.f.). *Filtros Activos, Conceptos Basicos y Diseño*. . Recuperado el 30 de Septiembre de 2016, de Quidel: [http://quidel.inele.ufro.cl/~jhuircan/PDF\\_CTOSII/ieeefact.pdf](http://quidel.inele.ufro.cl/~jhuircan/PDF_CTOSII/ieeefact.pdf)
42. Klem, G., & Elger, H. L. (1999). *The ten-twenty electrode system of the international Federation*. Estados Unidos: Elsevier Science.
43. Konrad, P. (2005). *Hermanwallace*. (Noraxon Inc.) Recuperado el 17 de Junio de 2016, de [https://hermanwallace.com/download/The\\_ABC\\_of\\_EMG\\_by\\_Peter\\_Konrad.pdf](https://hermanwallace.com/download/The_ABC_of_EMG_by_Peter_Konrad.pdf)
44. Kotsiantis, S. (2007). Supervised machine learning: a review of classification techniques. *Informática*, 31.
45. Kotsos, T. (19 de septiembre de 2016). *Las ondas cerebrales y los estados de la mente*. Obtenido de BibliotecaPleyades: [http://www.bibliotecapleyades.net/ciencia/ciencia\\_brain69.htm](http://www.bibliotecapleyades.net/ciencia/ciencia_brain69.htm)
46. L., A. (2016). *Gestión de proyectos según el PMI*. Catalunya: Universitat Oberta de Catalunya.
47. Liechti, C. (2016). *pySerial's Documentation*. Obtenido de <https://pyserial.readthedocs.io/en/latest/index.html>
48. Lindström LH, M. R. (1977). *Interpretation of myoelectric power spectra: a model and its applications*. USA: IEE.
49. Luca., C. d. (2006). *Delsys*. (E. o. Instrumentation, Productor) Recuperado el 17 de Junio de 2016, de [https://www.delsys.com/Attachments\\_pdf/Electromyography%20Medical%20Encyclopedia.pdf](https://www.delsys.com/Attachments_pdf/Electromyography%20Medical%20Encyclopedia.pdf)
50. Mallat, S. (1989). A theory for multiresolution signal decomposition: The Wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. *IEEE*, 11.
51. Martinez, R. (2015). *Curso de Ingeniería de Software*. Madrid: IT Capus Academy.
52. Mathworks. (2016). *Machine Learning: Train Classification Models in Classification Learner App*. Obtenido de [http://www.mathworks.com/tagteam/88253\\_92995\\_ML\\_Train\\_Classification\\_Models\\_App.pdf](http://www.mathworks.com/tagteam/88253_92995_ML_Train_Classification_Models_App.pdf)
53. McClellan, J. H. (1998). *Computer-Based Exercises for Signal Processing Using MATLAB 5*. Prentice-Hall.
54. Mendoza, R. (2004). *JAVAFilters: Calculo y diseño de filtros analogicos y digitales*. Mexico: UDLAP.
55. Microchip. (2004). *dsPIC30F3014, dsPIC30F4013 Data Sheet*. Recuperado el 4 de Octubre de 2016, de Microchip: <http://ww1.microchip.com/downloads/en/devicedoc/70138c.pdf>
56. Milvojevic, Z. (2009). *Digital Filter Design*. Mikroelektronika.
57. MongoDB, Inc. (2016). *Reinventando la gestión de datos*. Obtenido de <https://www.mongodb.com/es>
58. Morillo, L. (s.f.). *Análisis visual del electroencefalograma*. Recuperado el 19 de Septiembre de 2016, de acnweb: <http://www.acnweb.org/guia/g7cap17.pdf>

59. Morley, D. a. (2016). *Understanding Computers: Today and Tomorrow*. Comprehensive: Cengage Learning.
60. Moya, R. (03 de 08 de 2016). *Jarroba*. Obtenido de <http://jarroba.com/que-es-el-machine-learning/>
61. Murphy, K. (2012). *Machine Learning, A Probabilistic Perspective*. London, England: Massachusetts Institute of Technology.
62. National Instrument. (s.f.). *Aliasing*. Recuperado el 2 de Octubre de 2016, de Analysisconcepts: <http://zone.ni.com/reference/en-XX/help/370051M-01/cvi/libref/analysisconcepts/aliasing/>
63. Noppadon, J. S. (2013). Real-Time EEG-Based Happiness Detection System. *The Scientific World Journal*, 12.
64. Núñez Peña, I. C. (2004). Potenciales evocados cerebrales en el contexto de la investigación psicológica: una actualización. *Anuario de Psicología*, 35.
65. Oostenveld, R. (2002). *High-density EEG electrode placement*. Recuperado el 28 de Septiembre de 2016, de Electrode: <http://robertoostenveld.nl/electrode/>
66. Parker, R. M. (2004). *Electromyography: Physiology, Engineering, and Noninvasive Applications*. USA: EMB.
67. Preece, J. e. (1994). Prototyping. En J. e. Preece, *Human Computer Interaction* (págs. 537-565). Addison-Wesley.
68. Ramírez, R. (2008). Detecting emotion from EEG signals using the emotiv epoc device. *University Pompeu Fabra*, 11.
69. Reitermanová, Z. (2010). *Data Splitting*. Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.
70. Richiert, W. a. (2016). *Building Machine Learning System with Python*. Packt Publishing.
71. Rodríguez, J. (2005). *Gestión de proyectos informáticos*. San José: Editorial.
72. Rodríguez, N., & William, M. (2006). *Planificación y evaluación de proyectos informáticos*. San José: EUDNED.
73. Ruiz, G. (2001). Tema 8: El amplificador Operacional. En G. Ruiz, *Electronica basica para ingenieros* (págs. 139-156). España: Universidad de Cantabria.
74. Russell, J. (1980). "A circumplex model of affect," *Journal of Personality and Social Psychology*. Google Scholar.
75. Sánchez, L. y. (2003). *Metodologías ágiles en el Desarrollo de Software*. Valencia: Grupo ISSI de Alicante.
76. Sapsanis, C., Georgoulas, A., Tzes, A., & Lymberopoulos, D. (July de 2013). *Improving EMG based classification of basic hand movements using EMD*. Obtenido de UCI: 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 13 (EMBC 13): <https://archive.ics.uci.edu/ml/datasets/sEMG+for+Basic+Hand+movements>
77. Sherwood, J. &. (2009). On classification ability of wavelet features for EEG based brain-computer interfaces. *IEEE*, 23.

78. Silva, L. (2008). *Respuesta en frecuencia*. Venezuela: USM.
79. Sörnmo, L., & Laguna, P. (2005). *Bioelectrical signal processing in cardiac and neurological applications* (1st. ed.). San Diego, California, USA: Elsevier Academic Press.
80. Suárez, L. (2005). Visor de señales bioédicas para el fisiógrafo mk-iii-p de narco-scientific biosystem division. En L. Suárez, *Principios de procesamiento de señales* (págs. 55-76). Colombia: Universidad Pontificia Bolivariana.
81. T, H. C., M, D. S., & Foster, G. (2007). *Contabilidad de costos: Un enfoque gerencial*. México: Pearson Educación.
82. TatvaSoft. (22 de septiembre de 2015). *Top 12 Software Development Methodologies and Its*. Obtenido de TatvaSoft: <http://www.tatvasoft.com/blog/>
83. Tecnología . (s.f.). *Impresora 3D*. Recuperado el 5 de Octubre de 2016, de areatec: <http://www.areatecnologia.com/informatica/impresoras-3d.html>
84. Thevenet, D. (2008). *Instrumentación Biomédica con Seguridad*. Montevideo: Universidad de la República de Uruguay.
85. Trans Cranial Technologies . (2012). *10/20 System Positioning Manual*. China: Wachei.
86. Universidad de Navarra. (s.f.). *Muestreo y Cuantización*. Recuperado el 2 de Octubre de 2016 , de TECNU: <http://www4.tecnun.es/asignaturas/tratamiento%20digital/tema5.pdf>
87. Universidad del país Vasco. (2005). *Amplificador Operacional*. Recuperado el 3 de Octubre de 2016, de SC: [http://www.sc.ehu.es/acwamurc/Transparencias/\(2\)OPAMP.pdf](http://www.sc.ehu.es/acwamurc/Transparencias/(2)OPAMP.pdf)
88. Universidad del país Vasco. (30 de Septiembre de 2016). *Filtros activos*. Obtenido de Ftp: [ftp://ftp.ehu.es/cidira/dptos/depjt/Instrumentacion/BK-ANGEL/02\\_Filtros\\_Activos/Filtros.PDF](ftp://ftp.ehu.es/cidira/dptos/depjt/Instrumentacion/BK-ANGEL/02_Filtros_Activos/Filtros.PDF)
89. Universidad del país Vasco. (30 de Septiembre de 2016). *Filtros activos*. Obtenido de Ftp: [ftp://ftp.ehu.es/cidira/dptos/depjt/Instrumentacion/BK-ANGEL/02\\_Filtros\\_Activos/Filtros.PDF](ftp://ftp.ehu.es/cidira/dptos/depjt/Instrumentacion/BK-ANGEL/02_Filtros_Activos/Filtros.PDF)
90. University of Utah. (s.f.). *Anatomía del cerebro*. Recuperado el 9 de Septiembre de 2016, de HealthCare: <http://healthcare.utah.edu/healthlibrary/related/doc.php?type=85&id=P03866>
91. Valderrama, C. (2011). Análisis espectral de parámetros fisiológicos para. *Revista S&T*, 20.
92. Van Rossum, G., & Drake, F. (2000). *An Introduction to Python*. Nueva York: Network Theory LTD.
93. WageIndicator. (20 de septiembre de 2016). *Compara tu salario*. Obtenido de Tusalario.org: <http://www.tusalario.org/guatemala/>
94. Williams, J., Sochats, K., & Morse, E. (1995). "Visualization". Annual Review of Information Science and Technology.
95. Wolpaw, J., Birbaumer, N., McFarland, D., Pfurtscheller, G., & Vaughan, T. (2002). Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 767-791.
96. Yanowitz, G. F. (2006). *PHS Institute*. (University of Utah School of Medicine Medical Director) Recuperado el 17 de Junio de 2016, de <http://www.phsinstitute.com/ECG>

## XIII. ANEXOS

### A. Módulo: Comunicación con casco y pulsera para recepción y almacenamiento de datos

Figura 191. Código de Arduino utilizado para pruebas de comunicación por bluetooth

```
/*
 Software serial multiple serial test.
 modified 25 May 2012 by Tom Igoe based on Mikal Hart's example
 This example code is in the public domain.
*/
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX

void setup() {
 // Open serial communications and wait for port to open:
 Serial.begin(57600);
 while (!Serial) {
 ; // wait for serial port to connect. Needed for native USB port only
 }

 Serial.println("Goodnight moon!");

 // set the data rate for the SoftwareSerial port
 mySerial.begin(4800);
}

void loop() { // run over and over
 if (mySerial.available()) {
 mySerial.write("Hola mundo!");
 }
}
```

Figura 192. Código utilizado para buscar dispositivos de bluetooth con PyBluez

```
# file: inquiry.py
# auth: Albert Huang <albert@csail.mit.edu>
# desc: performs a simple device inquiry followed by a remote name request of
#       each discovered device
# $Id: inquiry.py 401 2006-05-05 19:07:48Z albert $
#

import bluetooth

print("performing inquiry...")

nearby_devices = bluetooth.discover_devices(
    duration=8, lookup_names=True, flush_cache=True, lookup_class=False)

print("found %d devices" % len(nearby_devices))

for addr, name in nearby_devices:
    try:
        print(" %s - %s" % (addr, name))
    except UnicodeEncodeError:
        print(" %s - %s" % (addr, name.encode('utf-8', 'replace')))
```

Figura 193. Programa utilizado para pruebas de comunicación por bluetooth con PyBluez

```
# Descripción: Programa que usa PyBluez para intentar establecer
# comunicación con el dispositivo de bluetooth 20:15:03:19:27:02.
# Autor: Albert Huang
# Obtenido de: https://people.csail.mit.edu/albert/bluez-intro/x232.html

import bluetooth
import sys
bd_addr = "20:15:03:19:27:02"

port = 1
sock=bluetooth.BluetoothSocket( bluetooth.RFCOMM )
sock.connect((bd_addr, port))
print 'Connected'
sock.settimeout(1.0)
sock.send("x")
print 'Sent data'

data = sock.recv(1)
print 'received [%s]'%data

sock.close()
```

## B. Módulo: Diseño y desarrollo de la plataforma Figura para EMOTRIX

Anexo 1: Comentarios obtenidos en las entrevistas sobre el prototipo en papel

Ocupación	Comentarios obtenidos
Psicóloga / Catedrática	Sus observaciones estuvieron dirigidas principalmente al color del API como tal. Según su opinión, la aplicación debería de llevar colores neutros. La razón por la cual recomendó estos colores es para disminuir la cantidad de posibles estímulos al usuario al usar la aplicación y que no influya en su estado emocional al momento de realizar la medición.
Estudiante Ing. Industrial	Es difícil comprender qué es lo primero que se debe de hacer pues no hay ningún botón de ayuda o algún tutorial que explique el funcionamiento de la misma. Si está dirigida para todo público, debería de haber algún tipo de explicación del proyecto y sus integrantes. Sin embargo, no hay nada de información sobre la app y sobre el proyecto.
Estudiante Ing. Industrial	También comentó respecto al color. Sugirió que para que al usuario le quedara claro su emoción, un buen aspecto de la aplicación podría ser que fuera cambiando de color con respecto a la emoción que fuera sintiendo. Es decir, si después de realizar la medición el usuario se sentía feliz, la aplicación podría ponerse con colores fuertes y brillantes que indiquen felicidad y colores oscuros y opacos con la tristeza.
Estudiante Ing. CC y TI	Tuvo dudas acerca del formato de la data que se va a dar como output (a partir del Generate Raw Data): la gente que esté interesada en el proyecto y en seguir investigando más el tema debería de poder contar con esta información de manera clara y precisa
Estudiante Ing. en Informática	Comentó sobre la mala comunicación de la app con el usuario. En otras palabras, la app no tiene forma de decirle al usuario que está realizando una medición (ni por cuánto tiempo) por lo que el usuario no sabe si está haciendo las cosas bien. Por otro lado, comentó que las gráficas no tienen ningún tipo de escala ni nombres de eje, por lo que no se sabe exactamente qué es lo que se está midiendo.
Psicóloga / Catedrática USAC	Las observaciones respecto de la falta de instrucciones fue lo principal, el usuario nuevo, o que no esté familiarizado con el software y el concepto no tiene la más mínima idea de que hacer al empezar a utilizar la aplicación.

Ocupación	Comentarios obtenidos
Ing. Ciencias de la Computación	Se observó que existe un botón de play y stop, pero que a la hora del análisis de la medición tal vez sea necesario una pausa, o inclusive botones para adelantar o atrasar la medición, sugirió el uso de un “reproductor” para ayudar al análisis de la medición.
Mercadóloga	Preguntó si se iba a tener un logo, ya que el nombre en letras puede ser muy burdo, sugirió el uso de un logo para no ocupar demasiado espacio en la pantalla, además de crear una identidad para Emotrix, ya que será Open Source y utilizado por personas que desconocen su origen.
Estudiante Ing. Industrial	Preguntó cuál era la manera correcta de usar el casco para la medición, y si el programa tenía alguna manera de mostrar si el contacto del casco y el usuario era el correcto antes y durante la medición.
Estudiante	Le gustó mucho el concepto, pregunto si había un manual de usuario o algo por el estilo y si al extraer la información había algún tipo de descripción que ayudará al usuario a saber qué hacer con ella, que significaba cada cosa y así.
Pedagoga	Refiere en cuanto al orden de los botones para dar inicio debe de ir primero Play, colocar el botón de información en la esquina superior derecha. Además de color verde al botón play y rojo al stop.
Psicólogo	Sugirió el uso de botón de información con un signo de interrogación, en la esquina superior derecha. El énfasis es en el uso del idioma español para hacerlo más fácil para la población guatemalteca. Colocar la mano/cara del lado izquierdo y la lectura en lado derecho.
Psicóloga	Indicó que asocia más la tristeza al color azul y el color amarillo a la alegría. Además de sugerir colores neutros para la aplicación con el fin de evitar la influencia de otro tipo de estímulo cuando se evalúa uno directo.
Psico pedagoga	Sugiere que se coloquen debajo de cada botón una instrucción en español para hacerlo más amigable al usuario. Además del cambio de mano/cara en lado izquierdo y en derecha la lectura, pues así lo hacen casi todas las aplicaciones (YouTube).
Maestra	Indica que los botones lleven un orden específico que sea similar al encontrado en aparatos, primero play, stop y pausa y en el lado superior. Sugiere cambio de cara/mano y lectura al revés. Indicó acerca de los colores que se usen colores que no sean chillones para no hacerla agresiva.
Psicóloga	Indica que, si le parece bien la posición de las imágenes y botones, solo le parece que la idea de el botón de información sea parecido al de Office, que es un “?” para generalizar y hacer más amigable a lo ya utilizado. Al igual que mover al lado izquierdo el modo de ver la emociones pues es más parecido a YouTube.
Psicóloga	Refiere que las pestañas lo hacen un poco aburrido y motiva a que se prueben otro tipo de metodología para que llame la atención, específicamente para gente de la generación Millenials, donde lo nuevo y tecnológico automáticamente atrae. La cara de felicidad le pareció demasiado exagerada, sino hacer un rostro más de caricatura.
Secretaria	Su forma le parece familiar, aunque preferiría una aplicación bastante neutra en cuanto a colores, pues le parece que si se van a medir emociones sea de forma que no influya otro estímulo. El botón indica que tengan un orden más parecido al utilizado primero play, pausa y luego stop. Las instrucciones hacerlas con la palabra debajo en español.
Administrador de empresas	Su impresión es que la aplicación es algo muy fácil de usar, añade que le parece que en un futuro próximo si es para población de nuestro país añadir algún idioma maya, pero que el español e inglés se le hace que es un inicio bueno. Añadiría otra pestaña de instrucciones, menú, objetivos y finalidad de la aplicación, pues le gusta leer todo tipo de antecedentes para este proyecto tan interesante.

Ocupación	Comentarios obtenidos
Administradora de empresas	Incluir menú principal, pues le parece que necesita antes de iniciar algo un poco de guía para no arruinarlo en el primer intento, pues al conocerlo podría participar en la prueba piloto de la aplicación. La lectura de emociones en situaciones de estrés o enojo le parece que debemos añadir pues es cuando menos el ser humano puede predecir su estado e intentar calmarse.

Anexo 2: Consentimiento informado utilizado en las entrevistas de Usabilidad

### Consentimiento Informado

En primer lugar, se le agradece su participación en esta entrevista perteneciente al Megaproyecto “EMOTRIX: Plataforma para el desarrollo de tecnología BCI enfocada en la identificación y manipulación de emociones humanas” desarrollado por estudiantes de la Universidad del Valle de Guatemala pertenecientes al último año de las carreras de Ingeniería en Ciencias de la Computación y Tecnologías de la Información; Ingeniería Mecatrónica, Ingeniería Electrónica, Ingeniería Industrial; y estudiantes de tercer año de Licenciatura en Psicología. Dicho proyecto es asesorado por el Ingeniero Sergio Izquierdo y el Director del Megaproyecto Tomás Gálvez.

El objetivo general del proyecto es diseñar e implementar herramientas tecnológicas “open-source” que permitan detectar emociones humanas básicas, y adaptar su funcionamiento a ellas. Debido a que el proyecto lo podrá utilizar cualquier persona, se debe de crear una interfaz intuitiva y fácil de usar para las mismas. Dado lo anterior, su participación en esta prueba determinará la mejor forma de diseñar la interfaz gráfica de EMOTRIX. Se mantendrá completa confidencialidad en todos los resultados obtenidos durante este estudio, y su anonimato está asegurado. Los datos que se pretenden recaudar serán utilizados expresa y únicamente para fines de esta investigación, por lo tanto, no se extraerán conclusiones de forma individual. Finalmente hacemos de su conocimiento que usted está en su derecho de retirarse de la realización de la entrevista en cualquier momento.

Yo: \_\_\_\_\_

Fecha de nacimiento: \_\_\_\_\_

El día de hoy:

1. Acepto participar en el proyecto “Emotrix” para fines educativos en la Universidad del Valle.
2. Comprendo la naturaleza y propósito del procedimiento.
3. He tomado la oportunidad de aclarar todas mis dudas.
4. Comprendo que la información que proporcione será anónima.
5. Estoy consciente de los posibles riesgos y beneficios que este proyecto representa para mí.
6. Entiendo que puedo retirarme en cualquier momento.
7. Estoy satisfecho(a) con la información que se proporcionó en este consentimiento informado.

Firma: \_\_\_\_\_

Fecha de hoy: \_\_\_\_\_

## Recopilación de Datos Generales

Fecha: \_\_\_\_\_ Prueba No: \_\_\_\_\_ Usuario No: \_\_\_\_\_

### *Información general*

Edad: \_\_\_\_\_ Sexo: M F

### *Educación*

- 1) Por favor indique el grado académico que posee
  - a. Secundaria
  - b. Diversificado
  - c. Universitario
  - d. Licenciatura
  - e. Maestría y demás
  
- 2) Si cursa algún curso universitario o mayor, favor de indicar el nombre y el año que cursa:  
\_\_\_\_\_

### *Experiencia en sistemas similares*

- 1) ¿Alguna vez ha oído acerca de tecnología BCI? \_\_\_\_\_
- 2) ¿Alguna vez ha oído acerca del dispositivo EMOTIV EPOC? \_\_\_\_\_
- 3) ¿Sabía que su cerebro produce 4 tipos de señales eléctricas llamadas “Señales Encefalográficas”? \_\_\_\_\_
- 4) ¿Sabía que existen 6 emociones fundamentales? \_\_\_\_\_
- 5) Si su respuesta anterior es sí, indique cuáles conoce:  
\_\_\_\_\_

### Questionario post-entrevista

1. ¿La organización de la interfaz le parece correcta?

---

---

---

2. ¿La estructura de la información es consistente con el resto de la interfaz?

---

---

---

3. ¿Tuvo problemas con el flujo de la interfaz?

---

---

---

4. ¿Los colores de la interfaz le parecen los adecuados?

---

---

---

5. Podría indicar algún problema o inconveniente que le sucedió utilizando la interfaz

---

---

---

C. Módulo Administrar los recursos en el desarrollo de la plataforma de tecnología Brain Computer Interface, registrar sistemáticamente con métricas el avance del proyecto y proporcionar la información mediante un análisis de costos para fundamentar la toma de decisiones en las siguientes fases de Emotrix

1. Seguimiento y control de las actividades. Los Gantt general y desglosados para las etapas de planificación, ejecución y control fueron los siguientes:

Figura 194. Gantt Megaproyecto para la fase de planificación

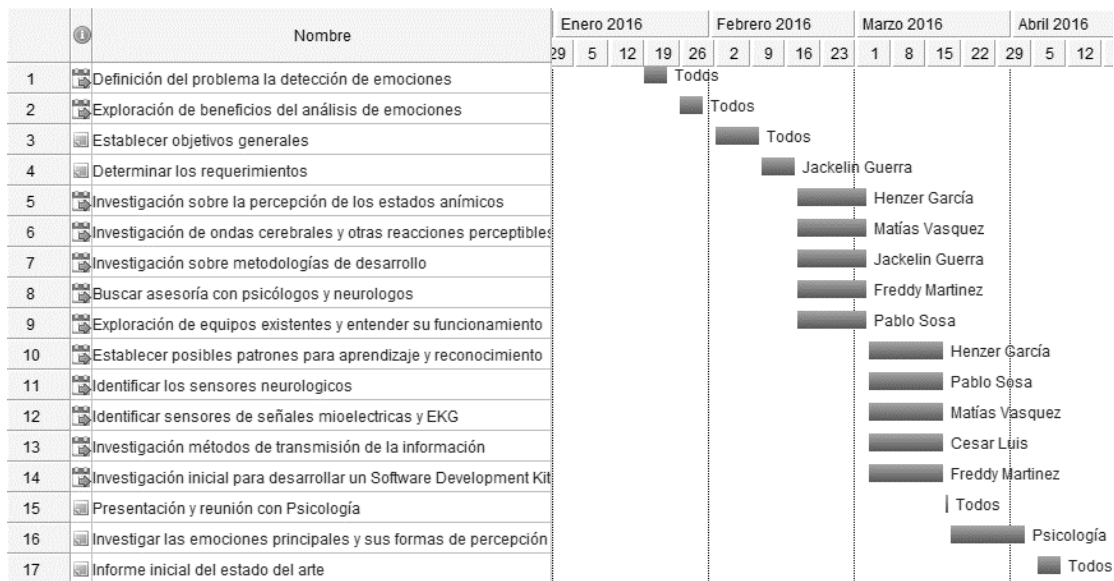


Figura 195. Gantt del módulo del desarrollo del software Development Kit

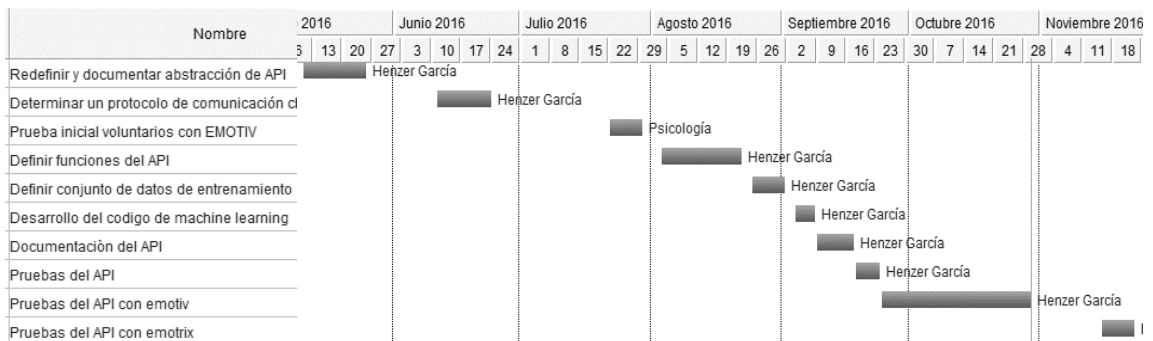


Figura 196. Gantt del módulo del casco de electrodos

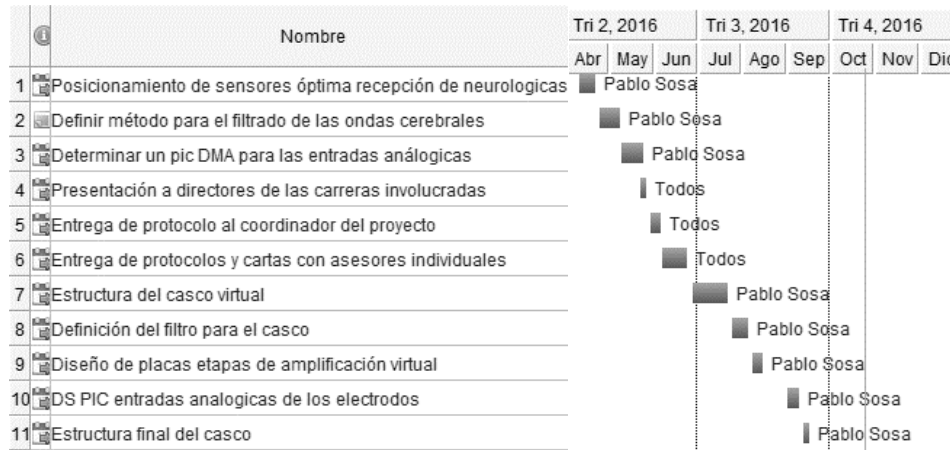


Figura 197. Gantt del módulo de la pulsera de señales electromiográficas

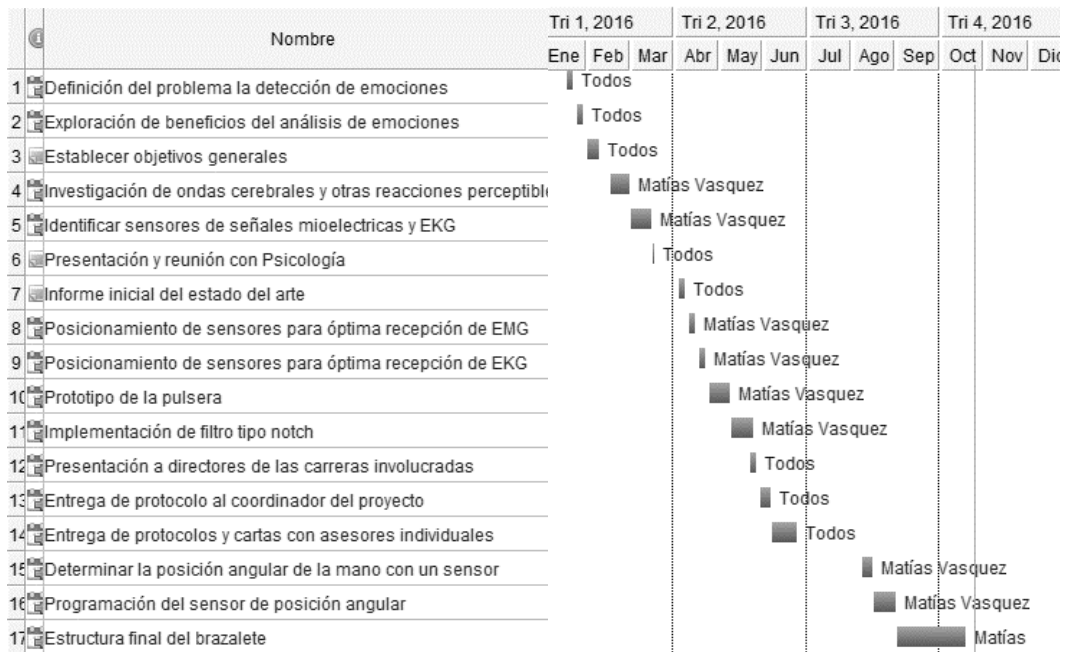


Figura 198. Gantt del módulo de la comunicación



Cuadro 2. Gantt módulo de interfaz figura para el usuario

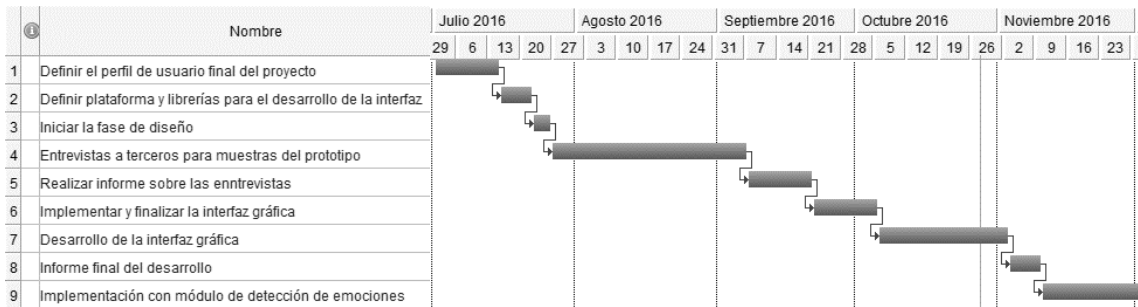
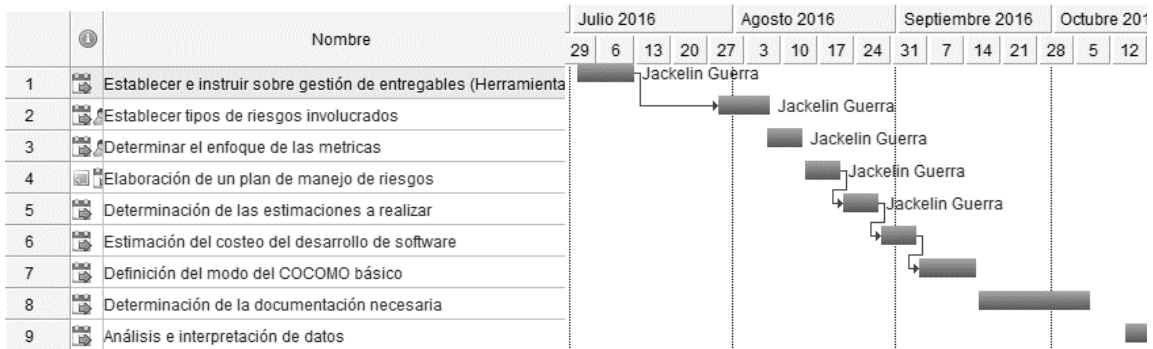


Figura 199. Gantt módulo de gestión del proyecto



## 2. Seguimiento vía Asana

Figura 200. Gestión y seguimiento de las actividades en la plataforma de Asana

### Emotrix

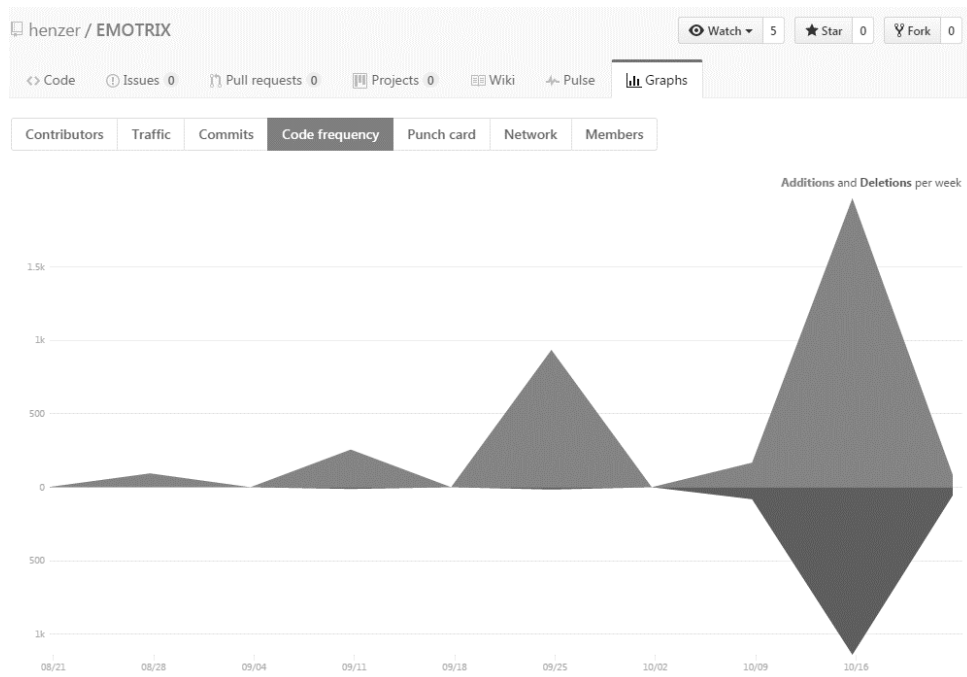
*Printed from Asana*

- Jackelin Guerra: Administración**
  - Jackelin Guerra: Establecer propuesta de calendarización** due May 4
  - Jackelin Guerra: Research y establecer herramientas** due May 27  
Cimentar nuestras herramientas y estructuras.
  - Jackelin Guerra: Inducción sobre ASANA** due April 23
  - Tomás: Enviar formatos de protocolo de MP** due March 14
  - Jackelin Guerra: Consentimiento Informado** due June 18
- Jackelin Guerra: Diseño de pruebas de Psicología**
  - Avances en el diseño de las pruebas
  - Francisco C.: Identificar de emociones a través de herramientas del software** due April 30
  - Mariana L.: Escala de depresión, inventario de estado mental** due April 30
  - Jackelin Guerra: Diseño de las pruebas** due July 30
  - Susset Fernandez: Diseño de prueba inicial uso de EMOTIV** due August 6
  - Jackelin Guerra: Consentimiento Informado** due March 10  
Task complete: cuando se adjunte el archivo pdf
  - Mariana L.: Borrador** due March 10
- Matias: Pulsera**
  - Henzer Ottoniel García Cruz: Investigación (estándares y formato de definición)** due May 14
  - Pablo Sosa: Mapeo de posición de sensores EMG** due June 11
  - Matias: Diseño inicial de pulsera** due June 25
  - Checha: Entradas y salidas** due August 13
  - Matias: Implementación de filtro tipo Notch** due August 20
  - Matias: Entregable: definición inicial (RFG)** due August 27
  - Henzer Ottoniel García Cruz: Pruebas iniciales de protocolo** due September 24
- Pablo Sosa: Gaseo**
  - Henzer Ottoniel García Cruz: Investigación (estándares y formatos de definición)** due June 15
  - Pablo Sosa: Research electrodos a conseguir y posicionamiento** due March 17  
Identificar los electrodos a utilizar para el emotrixC, el posicionamiento de los mismo.
  - Pablo Sosa: Identificación de electrodos a conseguir** due March 5
  - Francisco C.: Identificar posicionamiento de electrodos** due March 12
  - Pablo Sosa: Variables a leer: decidir cuáles serán las frecuencias que nos interesan** due March 17
  - Francisco C.: Investigar opciones adicionales a leer** due March 17
  - Pablo Sosa: Elegir microcontrolador para procesamiento de señal** due August 20
  - Checha: Entradas y salidas** due September 17

Figura 201. Gestión y seguimiento de las actividades en la plataforma de Asana

<input checked="" type="checkbox"/> Pablo Sosa: <del>Protocolo de comunicación casco-compu</del>	due April 1
<input checked="" type="checkbox"/> Pablo Sosa: <del>Escoger módulos de envío y recepción de señales</del>	due August 27
<input checked="" type="checkbox"/> Henzer Ottoniel García Cruz: <del>Pruebas iniciales de protocolo</del>	due September 30
<input checked="" type="checkbox"/> Pablo Sosa: <del>Entregable: definición inicial (request for comments RFC)</del>	due July 16
<input checked="" type="checkbox"/> Henzer Ottoniel García Cruz: <del>SDK</del>	
<input checked="" type="checkbox"/> Henzer Ottoniel García Cruz: <del>Research de herramientas de desarrollo de SDK</del>	due May 21
<input checked="" type="checkbox"/> Pablo Sosa: <del>Re-definición de protocolo</del>	due June 25
<input checked="" type="checkbox"/> Freddy: <del>Re-definición y documentación de Abstracción de API</del>	due July 23
<input checked="" type="checkbox"/> Pablo Sosa: <del>Identificación de los electrodos a utilizar</del>	due July 30
<input checked="" type="checkbox"/> Checha: <del>Comunicación y GUI</del>	
<input checked="" type="checkbox"/> Henzer Ottoniel García Cruz: <del>Propuesta inicial de protocolos de comunicación</del> Protocolos de comunicación hacia/desde computadora a dispositivos de hardware	due June 25
<input checked="" type="checkbox"/> Matias: <del>Protocolo de comunicación pulsera-compu</del>	due August 20
<input checked="" type="checkbox"/> Checha: <del>Definir codificación y de codificación de envío a compu</del>	due July 16
<input checked="" type="checkbox"/> Freddy: <del>Descarga, instalación de EEG</del>	due March 18
<input checked="" type="checkbox"/> gar12538@uvg.edu.gt: <del>Descarga e instalación de EEG</del>	due March 5
<input checked="" type="checkbox"/> gar12538@uvg.edu.gt: <del>Prueba de EEG (MATLAB)</del>	due March 5
<input checked="" type="checkbox"/> Tomás: <del>Pruebas adicionales: hacerlo funcionar</del>	due September 24
<input checked="" type="checkbox"/> Checha: <del>Prueba de EEG (MATLAB)</del>	due August 6

Figura 202. Líneas de código desarrolladas y eliminadas extraído de GitHub



3. Plantillas para minutas

<b>PLANTILLA PARA LAS MINUTAS</b>	
Fecha:	
Miembros del equipo presentes:	
Miembros equipo de apoyo presentes:	
Objetivos logrados desde la última reunión:	
Objetivos de la reunión:	
<b>OBJETIVOS POR:</b>	
Hardware casco - computadora	
Hardware pulsera - computadora	
General del software	
Administración	