

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

Maestría En Ingeniería Electrónica

DISEÑO DE UN DISPOSITIVO PARA LA ADQUISICIÓN DE UNA SEÑAL EEG QUE
INCLUYA LA REDUCCIÓN DE ARTEFACTOS OCULARES.

Ángela Patricia Guerrero Castillo



Bogotá, D. C. Junio de 2017

DISEÑO DE UN DISPOSITIVO PARA LA ADQUISICIÓN DE UNA SEÑAL EEG QUE
INCLUYA LA REDUCCIÓN DE ARTEFACTOS OCULARES.

Trabajo de grado para optar al título de Magíster en Ingeniería Electrónica con énfasis en Ingeniería
Biomédica

Ing. Javier Alberto Chaparro Preciado PhD.

Director del Trabajo de Grado

Jurado: _____

Jurado: _____

Nota de Aceptación:

El trabajo de grado titulado “DISEÑO DE UN DISPOSITIVO PARA LA ADQUISICIÓN DE UNA SEÑAL EEG QUE INCLUYA LA REDUCCIÓN DE ARTEFACTOS OCULARES” presentada por Ángela Patricia Guerrero Castillo, cumple con los requisitos establecidos para optar al título de Magíster en Ingeniería Electrónica con énfasis en Ingeniería Biomédica.

Jurado

Jurado

Bogotá, D.C. Junio de 2017.

*Dedico el presente trabajo a mi esposo y
a mi familia quienes lo son todo para mí.*

AGRADECIMIENTOS

A mi esposo Jonathan

Quien me apoya y me acompaña siempre en todos mis proyectos, dándome la fortaleza y brindándome su ayuda en todo momento.

A toda mi familia

En particular a mi mamá y a mi tía por haberme brindado la oportunidad de realizar mis estudios y estar pendientes de mí.

A mis profesores y a la universidad

Especialmente al ingeniero Javier Chaparro por la guía y el tiempo invertido en mí. Y al ingeniero Alexander Pérez por haberme brindado el espacio y tiempo necesarios para culminar mi trabajo de grado.

A mis compañeros

Por todas sus palabras de aliento ante las dificultades. A Nicolás por su gran ayuda y todo el tiempo dedicado en este trabajo. Y a Alexander por aportarme conocimiento tanto profesional como personal.

ÍNDICE GENERAL

INTRODUCCIÓN	11
OBJETIVOS	16
Objetivo General	16
Objetivos Específicos.....	16
MARCO REFERENCIAL	17
Bio-potenciales.....	17
Electrocardiografía	18
Electromiografía.....	19
Electroencefalografía	20
Sistema Internacional 10-20.....	21
Bandas Cerebrales	23
Potenciales evocados.....	24
Artefactos en las señales EEG.....	25
Artefactos Oculares	26
Artefactos Musculares.....	26
Artefactos por Movimiento	27
Artefactos por ECG.....	27
Otros artefactos	27
Procesamiento de señales EEG	28
Filtrado de la señal EEG.....	28
Reducción de artefactos oculares	29
Métodos basados en regresión lineal.....	29
Métodos de separación ciega de fuentes	30
ICA.....	30

CCA.....	31
EMD.....	31
Métodos de Filtrado Adaptativo.....	32
Estado del arte actual.....	32
METODOLOGÍA	35
Fase 1. Procesamiento de las señales de la base de datos de Physionet.....	35
Señales de la base de datos de Physionet.	35
Procesamiento de las señales de la base de datos de Physionet	36
Fase 2. Adquisición y procesamiento de señales EEG.....	37
Tarjeta de evaluación y demostración ADS1299EEG-FE	38
Protocolo de adquisición de las señales EEG.....	40
Procesamiento de las señales adquiridas	42
Fase 3. Implementación del dispositivo	42
Fase 4: Pruebas de funcionamiento	42
RESULTADOS Y CONTRIBUCIÓN	43
Fase 1. Procesamiento de las señales de la base de datos de Physionet.....	43
Adquisición de las señales de la base de datos de Physionet	43
Procesamiento de las señales con 64 canales de EEG.....	43
Procesamiento de las señales con 6 canales de EEG.....	46
Fase 2. Adquisición de señales EEG.....	51
Montaje y desarrollo del experimento.....	51
Procesamiento de las señales adquiridas	53
Fase 3. Implementación del dispositivo	57
Adquisición de la señal	57
Procesamiento de la señal	59
Visualización de la señal	66

Placa PCB.....	67
Circuito de Batería	67
Circuito de monitoreo de voltajes	68
Circuito del integrado ADS1299.....	69
Circuito del microcontrolador	70
Fase 4: Pruebas de funcionamiento.....	73
Contribución.....	78
CONCLUSIONES Y RECOMENDACIONES	80
REFERENCIAS	83

ÍNDICE DE FIGURAS

Figura 1. ECG normal [40]	18
Figura 2. EMG normal [41].....	19
Figura 3. Homúnculo de Peinfield [46].....	21
Figura 4. Localización de electrodos para EEG mediante el sistema 10-20 [19].....	22
Figura 5. Señal EEG [48]	23
Figura 6. Bandas cerebrales [46].....	24
Figura 7. Artefactos de la señal EEG [58].....	25
Figura 8. Localización de electrodos para EOG [59]	26
Figura 9. Diagrama del método de filtrado adaptativo [62]	32
Figura 10. Interfaz gráfica de la Tarjeta de Evaluación	38
Figura 11. Tarjeta de Evaluación del integrado ADS1299	39
Figura 12. Diagrama de bloques de la Tarjeta de Evaluación	40
Figura 13. Montaje de los electrodos para captar las señales.....	41
Figura 14. Señal Original 64 Canales.....	44
Figura 15. Elementos del vector de correlaciones 64 canales	45
Figura 16. Señal Reconstruida 64 Canales	45
Figura 17. Señal Final 64 Canales.....	46
Figura 18. Señal Original 6 Canales.....	47
Figura 19. Componentes de la señal 6 canales.....	48
Figura 20. Correlación de las componentes 6 canales.....	48
Figura 21. Señal Reconstruida 6 Canales	49
Figura 22. Señal Final 6 Canales.....	49
Figura 23. Señal original vs señal reconstruida canal 5.	50
Figura 24. Montaje del experimento en la tarjeta de evaluación.....	52
Figura 25. Señal adquirida con la tarjeta de evaluación.....	53
Figura 26. Señal Original Adquisición con Tarjeta de Evaluación	54
Figura 27. Componentes de la señal adquirida con la Tarjeta de Evaluación.....	54
Figura 28. Elementos del vector de correlación	55
Figura 29. Señal reconstruida de la adquirida con la Tarjeta de Evaluación	55
Figura 30. Señal Reconstruida del canal 5.	56
Figura 31. Señal filtrada de la adquirida por la Tarjeta de Evaluación	56

Figura 32. Tarjeta que contiene el integrado ADS1299 (izquierda) y Tarjeta Madre (derecha).....	58
Figura 33. Conexión entre la tarjeta que contiene el integrado y la tarjeta madre.	58
Figura 34. Comunicación SPI del integrado ADS1299.	59
Figura 35. Conexiones entre el microcontrolador y el integrado ADS1299.	60
Figura 36. Montaje completo de adquisición con el microcontrolador.....	61
Figura 37. Circuito de batería y carga de baterías.....	68
Figura 38. Fuente de 5V y de 3.3V regulados.....	68
Figura 39. Protección de voltajes y unión de tierras del dispositivo.....	69
Figura 40. Circuito del integrado ADS1299.	70
Figura 41. Circuito del microcontrolador y su programador.....	71
Figura 42. Placa PCB con sus componentes integrados.....	72
Figura 43. Tarjeta diseñada en modelo 3D.....	72
Figura 44. Señal original tomada por el dispositivo diseñado	73
Figura 45. Señal reconstruida después del procesamiento en Matlab®.....	74
Figura 46. Señal reconstruida por el procesamiento en el microcontrolador	74
Figura 47. Señal Original Vs Señal Reconstruida por Segmentos Canal 1.....	75
Figura 48. Señal Original Vs Señal Reconstruida por Segmentos Canal 2.....	76
Figura 49. Señal Original Vs Señal Reconstruida por Segmentos Canal 3.....	76
Figura 50. Señal Original Vs Señal Reconstruida por Segmentos Canal 4.....	77
Figura 51. Señal Original Vs Señal Reconstruida por Segmentos Canal 5.....	77
Figura 52. Señal Original Vs Señal Reconstruida por Segmentos Canal 6.....	78

INTRODUCCIÓN

Se estima que alrededor del 15% de la población mundial (más de mil millones de personas) vive con algún tipo de discapacidad [1]. La discapacidad es “un término general que abarca las deficiencias, las limitaciones de la actividad y las restricciones de participación. Las deficiencias son problemas que afectan a una estructura o función corporal; las limitaciones de la actividad son dificultades para ejecutar acciones o tareas, y las restricciones de la participación son problemas para involucrarse en situaciones vitales” [2]. La discapacidad más allá de ser una deficiencia física o mental es una situación que comprende la relación entre las personas discapacitadas y su entorno [3]. De tal manera que se trata de un “fenómeno que refleja la interacción entre las características del organismo humano y las características de la sociedad en la que vive” [2].

El número de personas con discapacidad va en aumento, debido a diferentes factores de salud, ambientales, accidentes, catástrofes naturales, violencia, hábitos alimentarios y abuso de sustancias. Algunas de las desventajas sociales que se generan por la no inclusión de las personas discapacitadas son: bajos resultados sanitarios, bajos resultados académicos y menor participación económica que conlleva tasas más altas de pobreza, mayor dependencia y participación limitada [1]. Existen también algunas áreas problemáticas como la educación, el empleo, la accesibilidad y movilidad, la asistencia médica y la legislación [4].

Se estima que en Colombia hay 4.992.000 personas con algún tipo de discapacidad y las principales causas son enfermedades adquiridas, lesiones causadas por accidentes de tránsito, lesiones causadas por accidentes laborales y violencia [4]. Según el informe del Departamento Administrativo Nacional de Estadísticas - DANE realizado en el año 2010, el 20% de las personas discapacitadas presentan una deficiencia en el movimiento de cuerpo, manos, brazos o piernas. [5]. Una de las causas más influyentes en el aumento de personas con discapacidades motoras es la presencia de minas antipersonas; en el periodo comprendido entre el año 1990 y 2015 se registraron un total de 11.073 víctimas de minas antipersona en Colombia, del cual el 80% de las víctimas resultaron heridas y el 20% murió en el lugar del accidente. 1.125 víctimas fueron menores de edad (22% niñas y 77% niños). Se identificó a Antioquia, Meta, Caquetá, Nariño y Norte de Santander como los departamentos con el mayor número de víctimas [6].

Es importante proponer soluciones que contribuyan al mejoramiento de la situación actual; es el caso de las políticas públicas que el Ministerio de Salud y Protección Social ha desarrollado para contrarrestar las desventajas sociales y las áreas problemáticas, en donde se expresa el “adecuado acceso a bienes y servicios, procesos de elección colectiva, la garantía plena de los derechos de los ciudadanos y la eliminación de toda práctica que conlleve a marginación y segregación de cualquier tipo” [7].

Para suplir algunas de las necesidades y mejorar la calidad de vida de las personas discapacitadas de miembros superiores y/o inferiores utilizan prótesis, definidas como “una extensión artificial que reemplaza una parte faltante del cuerpo” [8]. Las prótesis de miembro superior e inferior tienen finalidades específicas, por ejemplo, la función de una prótesis de miembro superior es la de agarrar y alcanzar objetos y la función de una prótesis de miembro inferior es realizar el apoyo de la bipedestación y su contribución en actividades de marcha. Tanto en el diseño de las prótesis de miembro superior como inferior, se buscan ciertas características que hagan que sean útiles y cumplan a cabalidad sus funciones tales como: condiciones de comodidad, poco peso, resistencia a impactos, confort, entre otros [9].

Hoy por hoy, las prótesis que se encuentran en el mercado son fabricadas de diferentes materiales y tienen diversos precios. Sin embargo, las prótesis robóticas (o biónicas) son las que están a la vanguardia. El término biónica se define como “el análisis del funcionamiento real de los sistemas vivos y, una vez descubiertos sus secretos, materializados en los aparatos” [10]. Así, lo que se busca es una prótesis robótica que imite de manera casi exacta el funcionamiento de la extremidad haciendo uso de la tecnología [9].

Existen especificaciones que la prótesis robótica debe satisfacer para que emule la dinámica del miembro en interés, éstas especificaciones son tamaño y masa, velocidad y torque, baterías y realimentación al usuario [8]. Ciertas prótesis robóticas que se han desarrollado tienen algunas de las especificaciones anteriormente mencionadas, en cuanto a prótesis de miembro superior se encuentra la mano mio-eléctrica (Otto Bock), la mano electrónica (Otto Bock), el codo – antebrazo ErgoArm (Otto Bock), la prótesis biónica 1 – Limb, el UTAH arm 3 y el Brazo de Luke, entre otros. En cuanto a las prótesis de miembro inferior se han desarrollado Power Knee (Ossur) y Propio Foot (Ossur) [8].

Sin embargo, es muy común que existan ciertas limitaciones para que estos sistemas funcionen correctamente. La batería, el tamaño o la localización de sistemas mecánicos hacen que la prótesis

no sea la adecuada para una persona [10]. Por tanto, se desarrollaron sistemas que controlan la prótesis robótica de manera remota, ahorrando así espacios y mejorando el procesamiento de las señales de realimentación con el usuario [11]. Ejemplo de estos sistemas son los conocidos como BCI (por sus siglas en inglés *Brain Computer Interfaces*), éstos traducen las señales provenientes del cerebro en señales de control para diferentes dispositivos tales como cursores de computador [12] [13], sistemas de deletreo [14], sillas de ruedas [15] [16] o prótesis robóticas [11]. El objetivo de un sistema BCI es adquirir la señal de cerebro que contiene un comando de control de movimiento (por ejemplo extensión y flexión de rodilla), realizar un procesamiento adecuado a la señal y entregar como salida órdenes específicas que controlen un dispositivo [12].

Para controlar los sistemas BCI se requiere de la adquisición de las señales del cerebro que se realiza por medio del electroencefalograma. Un electroencefalograma es el registro de las señales que se obtienen del cerebro mediante el uso de electrodos colocados de una forma específica en la superficie del cráneo, los electrodos captan la actividad eléctrica del encéfalo y lo reflejan en forma de ondas cerebrales [17]; éstas no tienen una forma definida pero pueden ser clasificadas en bandas conocidas como bandas cerebrales, cada una de ellas tienen unas características muy específicas, éstas bandas son: Delta, Theta, Alfa y Beta [18]. La configuración estándar para la ubicación de los electrodos en la corteza cerebral se conoce como sistema 10-20. Este sistema se basa en la relación entre la localización de un electrodo y el área subyacente de la corteza cerebral [19]. Recibe su nombre debido a que la medición total de la corteza cerebral está dividida en segmentos de 5, 10 o 20% [20]. Los electrodos para el registro superficial de bio-potenciales son fabricados en plata-cloruro de plata, platino, oro, grafito u óxidos semiconductores [21].

Los electroencefalógrafos en la actualidad tienen diferentes características dependiendo de su aplicación. Algunas de las características de electroencefalógrafos clínicos reúnen mecanismos de reducción de ruido, amplificadores, monitoreo constante del individuo, conexión a un computador a través de puertos USB, entre otros [22]. Sin embargo, en ocasiones el monitoreo al individuo debe ser a lo largo del día o desarrollando actividades en particular; para éste fin, existen algunos electroencefalógrafos ambulatorios, conocidos también como Holter EEG [23].

Para traducir de manera apropiada la señal EEG adquirida, se debe realizar un adecuado tratamiento de la misma, debido a que no solamente se captura la señal de interés sino que se captura toda la información de la actividad cerebral cercana al punto de registro [24]. A esta información (interferencia) se le conoce como artefactos y son alteraciones que cambian la forma de la señal

adquirida [25]. Los artefactos pueden producirse debido a movimientos involuntarios de la persona (latidos del corazón, cambio en la respiración, etc) como también por movimientos voluntarios de extremidades o del cuerpo [26]. No obstante, uno de los mayores inconvenientes con las señales EEG es que contienen artefactos producidos por actividad ocular [27], esto es que la señal se ve afectada en el momento en el que el usuario mueve sus ojos con los ojos abiertos, con los ojos cerrados, cuando parpadea o cuando hay un cambio de luz en el ambiente [28]. Existen métodos estadísticos para eliminar los artefactos, tales como ICA (Análisis de componentes independientes), diferentes tipos de regresiones y otros métodos estadísticos de alto orden [24] [25] [26].

A medida que los sistemas BCI han ido avanzando a lo largo de los años, se presentan propuestas de mejoras en la adquisición y tratamiento de la señal. Algunos ejemplos son las prótesis neurológicas, que por medio de un microelectromecanismo de sensores de electroencefalografía, adquieren señales de baja potencia, la amplifican y digitalizan y le realizan un procesamiento para que puedan ser enviadas inalámbricamente a un dispositivo móvil [29]. Otros, han intentado controlar sillas de ruedas por medio de dispositivos móviles utilizando diademas de EEG inalámbricas [30]. Se han realizado investigaciones para reemplazar los electrodos por sensores de no contacto por medio de acoplamiento capacitivo [31]. También, se han desarrollado investigaciones en donde se realiza un monitoreo constante al individuo a través de módulos de 19 sensores de la señal EEG y enviando la señal por protocolo Wi-Fi hacia un servidor por medio de socket TCP [32].

Es importante resaltar que los anteriores estudios muestran que los sistemas BCI y los equipos de adquisición de señales EEG en su gran mayoría dependen de los computadores para realizar el procesamiento, sin embargo, así como los algoritmos de procesamiento, la comunicación inalámbrica y el tamaño de los dispositivos ha ido mejorando; se deben enfocar esfuerzos para que sean sistemas más compactos, más cómodos y para desligar el uso de los computadores y migrar a sistemas embebidos que permitan mayor libertad y movilidad especialmente en personas con discapacidades. Adicionalmente, hablando en cuanto a la caracterización y clasificación de movimientos, los estudios indican que es importante garantizar una señal limpia de artefactos para que el sistema la traduzca en apropiados comandos de control.

En síntesis, las limitaciones pueden estar reflejadas desde la marginación en procesos de contratación en empleos, la reducción de la efectividad productiva del país, las pocas oportunidades de movilidad, seguridad y transporte público hasta la no utilización de prótesis debido a su costo o a

su diseño. Se pueden proponer soluciones que realicen mejoras significativas en la calidad de vida de las personas discapacitadas; pero se debe tener en cuenta que mientras se efectúan propósitos a largo plazo tales como hacer ajustes pertinentes al sistema de salud; construir y adecuar los edificios, oficinas, colegios, universidades, transporte y espacios públicos; investigar en el campo de la salud y la genética; entre otros; se pueden realizar cambios a las soluciones actuales, es decir realizar investigaciones en mejoras de prótesis y en la actualización de los sistemas BCI.

Con miras a implementar en un futuro un artefacto que pueda ayudar a personas con discapacidades en extremidades inferiores, a efectuar normalmente su proceso de marcha adquiriendo las señales de control de movimiento directamente desde el cerebro; se pretende con este trabajo responder a la pregunta ¿Cuáles son los resultados de la implementación de un dispositivo de adquisición de señales cerebrales de miembro inferior con procesamiento de eliminación de artefactos oculares en hardware? Esto con el fin de entregar una señal lista para realizar una caracterización y clasificación de movimientos de miembro inferior.

OBJETIVOS

Objetivo General

- Diseñar y probar un dispositivo para la adquisición de señales EEG de miembro inferior que incluya un procesamiento para reducir los artefactos causados por actividad ocular.

Objetivos Específicos

- Diseñar un dispositivo que adquiera una señal EEG de miembro inferior con todos sus componentes electrónicos integrados.
- Implementar el protocolo de comunicación entre el dispositivo de adquisición y el procesador de la señal.
- Realizar el tratamiento de la señal adquirida por medio de un microprocesador con el dispositivo diseñado para garantizar una señal limpia de artefactos causados por actividad ocular.
- Implementar una metodología de validación para realizar pruebas de funcionamiento del dispositivo diseñado en donde se evidencie el correcto procesamiento de la señal.

MARCO REFERENCIAL

Bio-potenciales

Para producir una acción, las células nerviosas sensoriales y motoras reciben y emiten información generando una secuencia de señales, una señal de entrada, de integración o activación, portadora y salida. Estas señales, conocidas como bio-potenciales, son generadas debido a la actividad electroquímica de células excitables componentes del tejido nervioso, muscular y glandular.

La actividad eléctrica de cada célula se produce por una alteración de su potencial eléctrico de membrana y se realiza un intercambio de iones en la membrana celular. En estado de reposo el potencial de la membrana es aproximadamente -70 mV; en el momento en el que la célula se estimula por el sistema nervioso central, los iones de sodio se difunden en el interior de la célula alcanzando un potencial de 40 mV. Este comportamiento se conoce como potencial de acción, la suma de varios potenciales de acción son el origen de señales que pueden ser medidas y analizadas. Dichas señales (bio-potenciales) registran actividad eléctrica de eventos biológicos ocurridos en diferentes parte del cuerpo humano, por ejemplo, en corazón llamado electrocardiografía (ECG); en músculos llamado electromiografía (EMG) y en cerebro llamado electroencefalografía (EEG) [33] [34] [35].

Para medir los bio-potenciales se utilizan electrodos, los electrodos son artefactos que captan la actividad iónica de los potenciales y la transforma en actividad eléctrica. Los tipos de electrodos utilizados para medir bio-potenciales se clasifican en tres grandes grupos: Microelectrodos, electrodos de superficie y electrodos de aguja [36].

Los microelectrodos son micro estructuras que se obtienen de las técnicas de fabricación de circuitos integrados, la estructura contiene puentes de silicio y un bisel para insertar el electrodo exclusivamente en un racimo nervioso [37].

Los electrodos de superficie son ubicados sobre la piel, generalmente son fabricados en metales como plata, oro, acero inoxidable, platino y otros. Los electrodos de superficie se dividen en electrodos secos, quienes no requieren de sustancias conductoras y electrodos húmedos, quienes deben ser acompañados por una sustancia electrolítica o un gel conductor para reducir el ruido

generado por la piel y el aire entre el electrodo y la superficie [38]. Se utilizan para medir ECG, EMG y EEG en la superficie de la piel [36].

Por último, los electrodos de aguja se utilizan para atravesar la piel y hacer un registro de potenciales en una región local específica [36]. Consisten en una aguja muy delgada de metal que es insertada directamente en el cerebro, en caso de EEG o en el músculo, en caso de EMG. La ventaja de los electrodos de aguja es que se logra una amplitud mayor dependiendo de la cercanía con el lugar de medición; así como también una mejoría en la captación de la señal debido a la poca interferencia con otros bio-potenciales. Sin embargo, al ser un método totalmente invasivo, su uso se reduce a experimentos clínicos en fases muy avanzadas [38].

Electrocardiografía

El ECG es el registro del movimiento del miocardio (músculo cardíaco) cuando se realizan movimientos de contracción y relajación, con el fin de bombear la sangre de los órganos y venas hacia el corazón y luego trasladarla a los pulmones para la liberación de dióxido de carbono y la captura de oxígeno [33]. “Es un gráfico en el que se estudian las variaciones de voltaje en relación al tiempo” [39]. Se puede registrar en formas diferentes, bien sea en papel milimetrado, cuya duración normal es de 30 segundos; o de manera continua a través de un monitor cardíaco; éste último registro se realiza durante cirugías o casos especiales [39].

La suma de los potenciales de acción generados durante un latido completo del corazón produce una señal compuesta por diferentes ondas. La onda P es usualmente la primera onda identificada en el electrocardiograma y es producida por la despolarización de las aurículas, el complejo QRS durante la despolarización de los ventrículos y la onda T en la repolarización de los ventrículos [33]. Esta señal se conoce normalmente como QRS debido a que en esta etapa es en la cual se presenta mayor actividad eléctrica. En la Figura 1 se observa un electrocardiograma normal.



Figura 1. ECG normal [40]

El ECG se utiliza fundamentalmente para detectar trastornos del ritmo cardiaco, también conocidos como arritmias, así como también en el diagnóstico de enfermedades que afectan al individuo debido a un bajo aporte de sangre al corazón [39]. Algunas de las arritmias encontradas con la ayuda del ECG son:

- Fibrilación Auricular.
- Angina de pecho e infarto.
- Aumento de cavidades del corazón.
- Enfermedades de válvulas cardiacas.
- Trastornos de los electrolitos sanguíneos.
- Pericarditis.

Electromiografía

La señal EMG registra los potenciales de acción generados en el proceso de contracción muscular. La señal corresponde a una señal aleatoria; normalmente es de amplitudes bajas, de 0 a 10 mV y de baja frecuencia, de 50 a 150 Hz; por lo tanto, puede verse afectada por ruidos eléctricos [33]. La señal de EMG se muestra en la Figura 2.

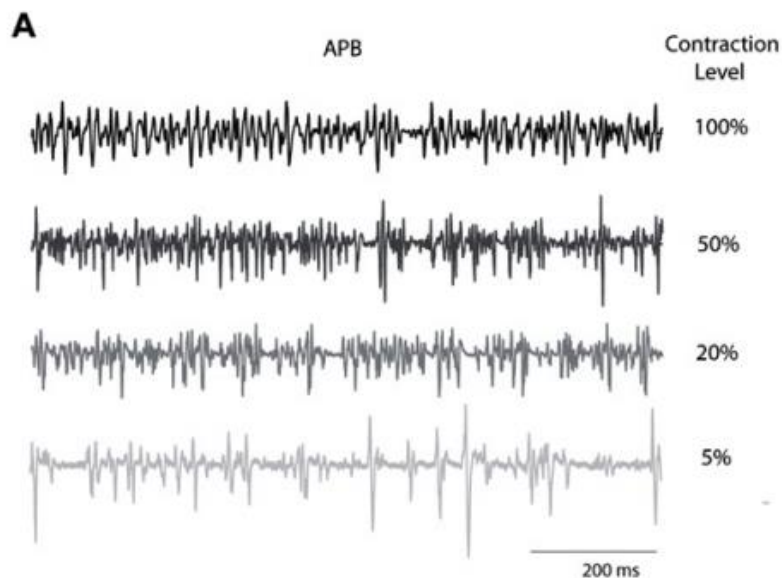


Figura 2. EMG normal [41]

El EMG tiene fines de tratamiento o diagnóstico de situaciones como: entumecimiento de músculos, hormigueo, dolor, debilidad constante, cojera, músculos atrofiados, defectos en tendones o fatiga entre otros [42].

Electroencefalografía

El cerebro es el órgano principal del sistema nervioso central, controla y coordina tanto los movimientos involuntarios (respiración, digestión, movimientos viscerales, etc.) como los voluntarios (extremidades). Se estima que existen 10^{11} células nerviosas en el cerebro, que están altamente conectadas y que realizan entre 1000 y 10000 sinapsis [43].

La señal cerebral proviene de la actividad de sincronización sináptica que realizan las neuronas corticales; dicha actividad excita las células postsinápticas y se crea un voltaje extracelular cercano a las dendritas neurales, el cual es más negativo que en cualquier otra región de la neurona. Se genera una situación en donde existe una región positiva y una región negativa separada por una distancia; a la región positiva se le conoce como fuente y a la región negativa como depósito, esto se conoce como dipolo [44].

La señal viaja desde el interior del cerebro hasta el electrodo. En el inicio, la señal se propaga mediante el volumen de conducción, el cual es un proceso en donde los iones repelen iones cercanos con igual carga, a su vez, estos iones repelen otros y así sucesivamente hasta que el proceso se transforma en una onda con carga que viaja a través del espacio extracelular. Cuando la señal alcanza el límite del cerebro, con el fin de ser captada, ésta debe atravesar la duramadre, las capas del cráneo, el cuero cabelludo y finalmente llegar al electrodo [44]. A causa de la forma del electrodo de superficie, se crea un aislante natural, el aire entre el cuero cabelludo y la superficie del electrodo, que impide la completa propagación de la señal; es por esto que se precisa la utilización de un gel conductor que cierre dicho espacio y facilite la conducción.

Para captar la señal cerebral sobre el cuero cabelludo se utilizan electrodos de superficie, mientras que si se desea registrar la actividad cerebral en la base del cráneo se utilizan electrodos basales, el registro de cualquiera de estos dos tipos de electrodos se conoce como EEG. Se define el EEG como “una técnica que provee medidas de la distribución espacial de los campos de voltaje del cerebro, la cual varía en función del tiempo” [45]. En el caso en el que se registre la señal cerebral en la superficie de la corteza craneal, se utilizan electrodos quirúrgicos, éste registro se conoce

como electrocorticograma (ECoG). Por último si se desea registrar la señal específica en un punto de la corteza cerebral se utilizan electrodos quirúrgicos de aplicación profunda y cuyo registro se conoce como estéreo electroencefalograma (E-EEG) [46].

Para ubicar las zonas sensoriales y motoras, se utiliza el *Homúnculo de Penfield*, el cual es un mapa de la corteza cerebral que muestra de manera detallada la parte de la corteza a la que corresponden ciertas partes del cuerpo. En la Figura 3a se observa el homúnculo sensitivo en donde se resalta la parte de la cara y de las manos y en la parte b de la figura se muestra el homúnculo motor; allí se observa que también gran parte de la corteza contiene información de la mano y de la cara [46].

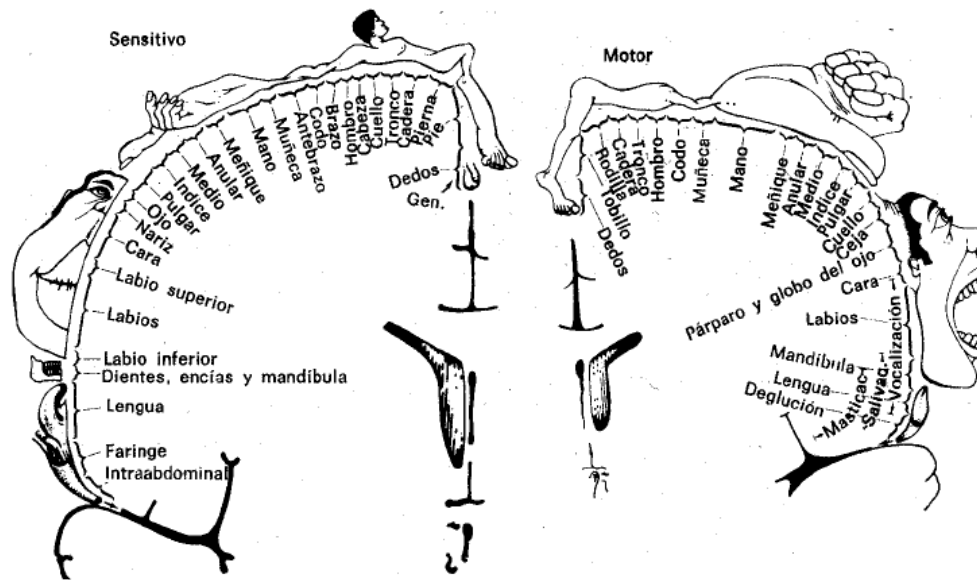


Figura 3. Homúnculo de Penfield [46]

Sistema Internacional 10-20

En la toma de la señal EEG se realiza la ubicación de los electrodos de superficie sobre el cuero cabelludo mediante el sistema internacional 10-20 el cual localiza los electrodos con base en la relación entre la localización del electrodo y el área de la corteza cerebral. Los números hacen referencia a que en la superficie craneal se localizan los electrodos bien sea al 10% o 20% de la distancia total entre la parte frontal y la parte posterior; así como entre la parte izquierda y derecha del cráneo [19].

Para lograr una identificación de los electrodos en el cerebro se nombran las secciones del mismo, haciendo uso de las letras F (frontal), T (temporal), C (central), P (parietal) y O (occipital). El número cero (en la imagen se denota con la letra *z* debido a su traducción del idioma inglés) se refiere al electrodo localizado en la línea media; los números pares hacen referencia al hemisferio derecho mientras que los impares hacen referencia al izquierdo [19].

En la Figura 4 se muestra el diagrama del sistema 10-20 y la localización de los electrodos; así como también el porcentaje específico entre cada uno de los puntos.

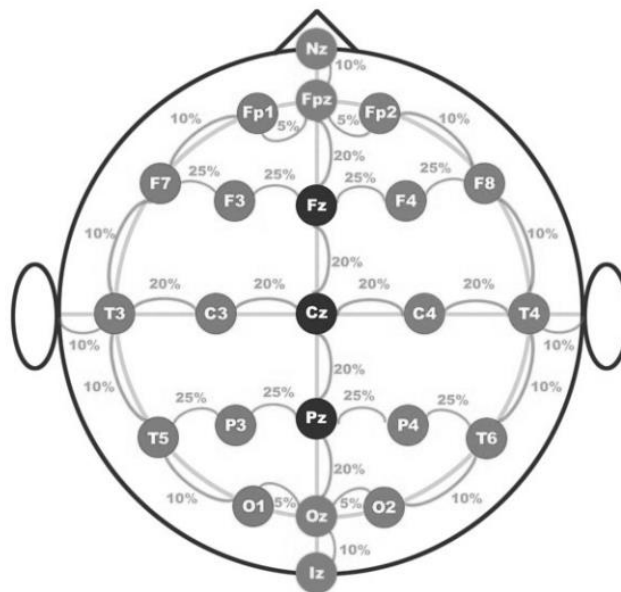


Figura 4. Localización de electrodos para EEG mediante el sistema 10-20 [19]

La adecuada colocación de los electrodos se realiza mediante un procedimiento específico; a continuación se explica la fase inicial del proceso, para el proceso completo se recomienda referirse a la referencia [19].

Inicialmente se mide la distancia entre el hueso Nasion (el puente de la nariz) y el hueso Inión (protuberancia en la parte posterior del cráneo); esta es la medida entre el electrodo Nz y el electrodo Iz. Luego se mide la mitad de esta distancia, esta posición corresponde al electrodo Cz. Posteriormente, se calcula el 10% de la distancia total entre Nz e Iz y a partir de cada uno de los puntos se ubican los electrodos Fpz y Oz respectivamente. Finalmente, se calcula el 20% de la distancia entre Nz e Iz y se colocan los electrodos Fz y Pz [19]. Estos electrodos son los electrodos

de referencia, para la ubicación de una zona particular se debe realizar la referencia con el homúnculo de Peinfield y el sistema 10–20 simultáneamente.

Bandas Cerebrales

Las señales obtenidas en el EEG son difíciles de interpretar porque representan la actividad de billones de neuronas transmitidas por membranas, fluidos y el cuero cabelludo. En la Figura 5 se puede ver una señal EEG normal. Las señales EEG son señales con frecuencias entre los 0.5 Hz a 100 Hz y con voltajes comprendidos entre $1\mu\text{V}$ y $200\mu\text{V}$ [46] [47].

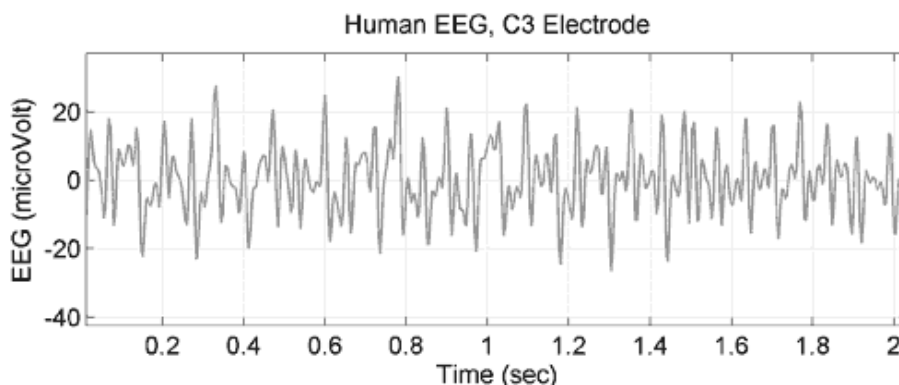


Figura 5. Señal EEG [48]

La señal EEG puede ser categorizada mediante bandas cerebrales, también conocidas como ondas o bandas cerebrales [49]. La Figura 6 muestra las bandas cerebrales [46].

Banda Beta (β): Corresponden a cuando la persona está pensando, respondiendo a algún estímulo o en estado de vigilia; su frecuencia está entre 13 Hz a 30 Hz y su voltaje es menor de $20\mu\text{V}$ [49]. Se encuentra en la región parietal y frontal [47]. Se atenúa debido a interferencias musculares en las zonas de lectura [50].

Banda Alpha (α): La persona está despierta y en estado de relajación y concentración mental; su frecuencia oscila entre 8 Hz a 13 Hz y su voltaje está típicamente entre 20 y $200\mu\text{V}$ [46] [49]. Puede ser encontrada con mayor amplitud en la región occipital [47]. Ésta banda reduce su actividad cuando los ojos están abiertos o con estímulos auditivos; para obtener la banda adecuadamente se requiere que los ojos se encuentren cerrados. Es la banda cerebral que mejor determina si el sujeto se encuentra despierto o en alerta [50].

Banda Theta (θ): La persona está dormida, su frecuencia oscila entre 4 Hz a 8 Hz y son señales de más de 20 μV [49]. Se encuentra en las regiones temporales [47]. Es normal en menores de 15 años y en adultos somnoliento [50].

Banda Delta (δ): La persona está en un estado profundo de sueño, su frecuencia se encuentra entre 0,5 Hz a 4 Hz [49]. Se encuentra en diferentes regiones cerebrales [47].

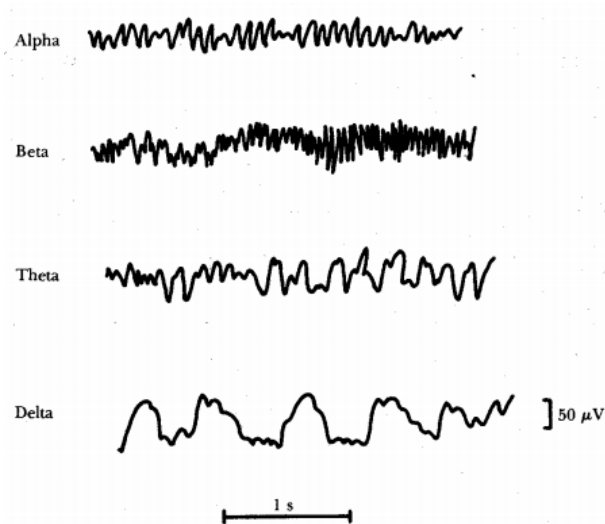


Figura 6. Bandas cerebrales [46]

Potenciales evocados

Los potenciales evocados son pequeñas variaciones en el electroencefalograma que se hallan sincronizadas con un estímulo cognitivo [51]. Por medio de los potenciales evocados se puede evaluar el estado neurológico de una persona porque permite diagnosticar problemas en la trayectoria sensorial del sistema nervioso [52].

Los estímulos pueden ser visuales para verificar el funcionamiento del nervio óptico y las vías visuales cerebrales; auditivos para estudiar la vía auditiva hasta el cerebro o somatosensoriales, que estudian las vías de las sensaciones cutáneas desde mano o pie hasta cerebro [53].

La onda P300 es uno de los potenciales evocados cognitivos que más se utiliza para estudiar las funciones cognitivas y de atención; y permiten medir la actividad neuronal inducida por la tarea antes de ser efectuada [51]. Se presenta como un pico positivo en el registro del EEG, 300 ms

después de que se realiza un estímulo aleatorio [54]. La adquisición de la onda es más fuerte cuando los electrodos se ubican en la región parietal, sin embargo también pueden obtenerse buenas señales en regiones frontales y temporales [55].

Artefactos en las señales EEG

Los artefactos en señales EEG son señales de interferencia generadas bien sea por movimientos voluntarios e involuntarios [26] o por interferencias electrónicas externas. La detección de artefactos es de vital importancia para minimizarlos y realizar un adecuado tratamiento de la señal dependiendo de su aplicación [56].

Los artefactos del EEG pueden ser fisiológicos o no fisiológicos; los primeros corresponden a movimientos de cabeza, cuerpo o pulsaciones en el cráneo; también otros bio-potenciales producidos por ojos, lengua y movimientos musculares; además actividades de las glándulas sudoríparas o actividad vasomotora. Los artefactos no fisiológicos corresponden a interferencia eléctrica externa o mal funcionamiento del sistema de adquisición [57]. Los artefactos más comunes en la señal EEG se presentan en la Figura 7.

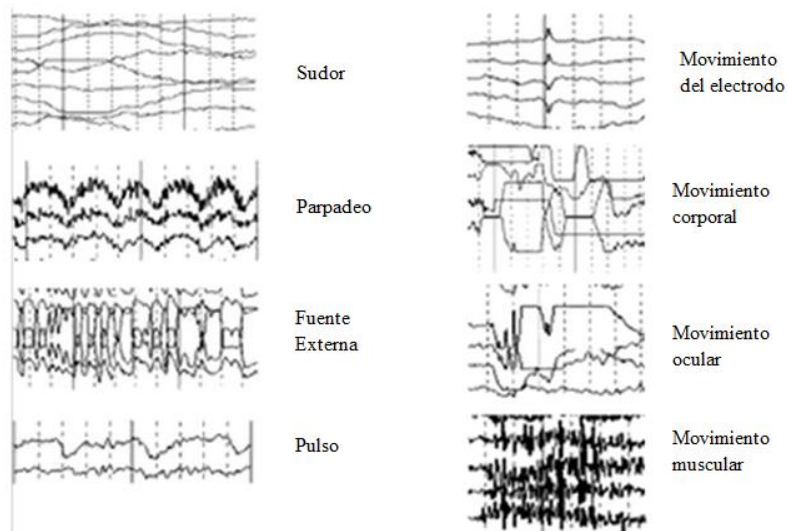


Figura 7. Artefactos de la señal EEG [58]

En algunos casos, los artefactos pueden ser identificados gracias a su forma y distribución, sin embargo, en otros casos se requiere de una persona experta que los detecte [57]. Es posible eliminar inicialmente artefactos no deseados realizando una verificación de los electrodos, en caso en el que

solamente uno de los electrodos muestre una amplitud alta se considera como artefacto. También se pueden eliminar segmentos del EEG que contengan formas repetitivas, irregulares u ondas rítmicas que no corresponden a la zona en la que se ubican los electrodos [57].

Artefactos Oculares

El movimiento de los ojos corresponde a la mayor fuente de contaminación en el registro EEG. En el EEG se pueden observar diferentes movimientos, cada uno de ellos modifica el potencial eléctrico alrededor de los ojos y por lo tanto del cráneo. Existe el movimiento dipolo entre cornea y retina, el movimiento dipolo de retina y movimiento de párpado [59]. “En el momento en que se producen movimientos del glóbulo ocular, el campo eléctrico alrededor del ojo cambia, produciéndose una señal eléctrica conocida como electro-oculograma (EOG). Esta señal se propaga por el cuero cabelludo apareciendo en la señal EEG como artefactos oculares no deseados” [60].

El EOG ha sido una herramienta efectiva para tratar los artefactos oculares, se realizan mediciones alrededor del ojo para captar tanto movimientos verticales como horizontales, en la figura se muestra el esquema utilizado para medir el EOG. En la Figura 8 se observa que el electrodo E7 corresponde a la referencia mientras que los otros corresponden a las mediciones tanto verticales como horizontales [59].

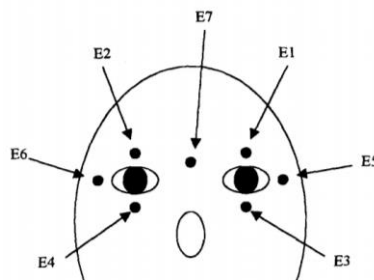


Figura 8. Localización de electrodos para EOG [59]

Artefactos Musculares

Los artefactos musculares aparecen debido a movimientos faciales, de mandíbula o parpadeo. Causan pequeños potenciales especialmente visibles en las regiones frontal y temporal aunque pueden ser captados por cualquier electrodo a lo largo del cráneo. En ocasiones reduciendo las

frecuentes altas por medio de filtros, éstos artefactos son eliminados, sin embargo, los picos de artefactos musculares reducen su forma y pueden ser confundidos con picos epilépticos y artefactos musculares repetitivos pueden ser confundidos con la onda alfa o beta.

Son fácilmente reconocibles por su forma y repetición, pueden ser eliminados desde la adquisición pidiendo al individuo que relaje la mandíbula, que mantenga la boca abierta o que cambie de posición. También puede ser eliminado al empujar el electrodo un poco para que se mantenga en una sola posición o masajeando la zona en donde se está colocando el electrodo [57].

Artefactos por Movimiento

En caso en el que el individuo mueva el cuerpo o la cabeza o incluso los cables de los electrodos, se producen artefactos de movimiento. Son fácilmente reconocibles por su duración y la asociación visual con movimientos, de manera tal que se pueden eliminar pidiendo al individuo que no realice ningún movimiento; en el caso de individuos con desórdenes de movimiento, tales como enfermedad de Parkinson, los artefactos de movimiento pueden ser removidos por alguna técnica de tratamiento de señales [57].

Artefactos por ECG

El artefacto por electrocardiograma puede aparecer en los electrodos de referencia, normalmente se observa un poco la onda R, de amplitud baja en la zona temporal derecha como una onda negativa. También se encuentran ondas P y T cuando el artefacto alcanza mayores amplitudes. En caso de que el individuo cuente con marca pasos, entonces el artefacto se verá reflejado en altas frecuencias.

Éste artefacto es difícil de eliminar en el proceso de adquisición, requiere de técnicas de procesamiento de señales para eliminarlo o atenuarlo [57].

Otros artefactos

En el registro de EEG se pueden encontrar adicionalmente artefactos de pulso con ondas sinusoidales o triangulares debido a ondas producidas por el contacto entre las arterias del cráneo y el electrodo. Artefactos causados por la resistencia de la piel, por ejemplo sudor o cambios

eléctricos en el potencial por el contacto de la piel con el electrodo. Movimientos de lengua o enfermedades inherentes al individuo pueden generar artefactos también.

Por otro lado, las interferencias externas deben ser tenidas en cuenta en un registro EEG. Es el caso de la conexión eléctrica del dispositivo que adquiere la señal. Interferencias de 50 Hz o 60 Hz pueden verse reflejadas en el registro. Adicionalmente, interferencias de otros dispositivos electrónicos tales como estaciones de televisión, de comunicación, marca pasos, entre otros [57].

Procesamiento de señales EEG

Idealmente, los sistemas BCI requieren señales limpias de cualquier artefacto mecánico, eléctrico, electrónico o biológico. Sin embargo, no es posible obtener una señal pura si se está realizando una adquisición de señal EEG mediante electrodos de superficie. No obstante, a través de la implementación de filtros y algoritmos basados en métodos matemáticos para identificación de frecuencias de interés y de eliminación de artefactos de diferentes fuentes, se puede lograr una señal lo suficientemente limpia para su utilización en sistemas BCI.

Filtrado de la señal EEG

Debido a que la señal EEG registra una banda de frecuencias reducida, se deben realizar algunos filtros con el fin de eliminar frecuencias que realicen interferencias y afecten directamente en el procesamiento.

Es el ejemplo de una componente de corriente directa que se puede observar con facilidad en los registros EEG, para su eliminación se aplica un filtro pasa-altas con frecuencia de corte entre 0,5 y 1 Hz; estas frecuencias suelen asociarse con ruidos de electrodos y algunos movimientos voluntarios o de electrodos. Otros filtros importantes corresponden a un pasa-bajo con una frecuencia de corte entre 35 y 70Hz para reducir artefactos musculares y adicionalmente un filtro tipo Notch en 50 o 60 Hz para eliminar ruidos producidos por la red eléctrica, así como también un filtro conocido como media móvil el cual suaviza la señal y a su vez elimina artefactos de altas frecuencias [61].

A causa de que las ondas cerebrales no pueden ser clasificadas únicamente mediante sus características físicas sino que requieren de clasificación por bandas cerebrales, se necesitan filtros

especiales para cada una de ellas. Según el objetivo final del sistema BCI, se recomienda implementar filtros pasa-bandas con frecuencias de corte correspondientes a las frecuencias de la banda cerebral en interés. Sin embargo, se invita a cuidar el uso de los filtros digitales lineales ya que pueden realizar cambios tanto de amplitud como de fase en la señal EEG [62].

Reducción de artefactos oculares

La reducción de artefactos oculares es de vital importancia debido a que la señal ocular es la que más contamina la señal EEG. El movimiento de los ojos realiza un cambio en el campo eléctrico que rodea a los músculos de los ojos y distorsiona el campo eléctrico en el cráneo. La utilización de señales EEG contaminadas por artefactos oculares puede arrojar resultados erróneos en el análisis, puesto que aportan información no solamente de amplitud sino de frecuencia que interfiere en la señal EEG principal, se resalta que el efecto de los artefactos oculares debe ser tenido en cuenta en cualquier análisis de señal EEG [59].

Existen diferentes tipos de artefactos oculares que se relacionan a diferentes movimientos: cambios en la retina (recepción de luz u oscuridad), movimientos horizontales, verticales y de párpado. Esta actividad se refleja de diferentes formas en los registros de EOG [59]. Existen expertos que pueden identificar los artefactos oculares a simple vista en el registro EEG, sin embargo, es necesario de diferentes técnicas matemáticas que determinen automáticamente los artefactos oculares por medio de umbrales o características especiales de la señal. Por lo anterior, se han desarrollado diversos métodos para la reducción o eliminación de los artefactos oculares que pueden ser divididos en tres grupos: métodos basados en regresión lineal, métodos de separación ciega de fuentes y métodos de filtros adaptativos [62].

Métodos basados en regresión lineal

Estos métodos consisten en lograr una aproximación a la señal original EEG atenuando la señal EOG, para la realización de dichos métodos se requieren electrodos de EEG y EOG registrando señales al mismo tiempo. Se determina un factor de atenuación y se aplica a la señal EOG, posteriormente se resta de la señal adquirida para encontrar la señal EEG limpia. Estudios han comprobado que si se realiza una regresión múltiple se puede llegar a mejores resultados. La

desventaja de los métodos basados en regresión lineal es que requieren de una señal EOG de referencia simultánea con la señal EEG [62].

Métodos de separación ciega de fuentes

Los métodos de separación ciega de fuentes (BSS por sus siglas en inglés: blind source separation) son métodos de aprendizaje no supervisado, no requieren de señales externas y estiman la señal original a partir de la señal adquirida (también llamada señal mezclada). Los algoritmos de BSS consideran que la señal adquirida $x(t)$ es una señal mezclada que está compuesta por una matriz de mezcla \mathbf{A} y las señales originales $s(t)$ (Ecuación 1). De manera tal que por medio de arreglos matriciales se puede determinar una matriz de desmezcla \mathbf{W} que al multiplicarla por la señal adquirida $x(t)$ arroje una estimación de la señal real $y(t)$ (Ecuación 2); la estimación corresponde a las componentes de la señal real [63].

Ecuación 1

$$x(t) = \mathbf{A} \cdot s(t)$$

Ecuación 2

$$y(t) = \mathbf{W}x(t)$$

Los métodos BSS son: Análisis de Componentes Independientes (ICA por sus siglas en inglés Independent Component Analysis), Análisis de Correlación Canónica (CCA por sus siglas en inglés Canonical Correlation Analysis) y Modo empírico de descomposición (EMD por sus siglas en inglés Empirical Mode Decomposition) [63].

ICA

Los algoritmos basados en ICA son los más utilizados para reducción de artefactos en señales EEG [62]. El análisis ICA consiste en encontrar la matriz de mezcla \mathbf{A} determinada de tamaño $N \times P$ y las señales fuente $s[m]$ cuyas componentes son lo estadísticamente más independientes posible. El algoritmo considera que la señal EEG y EOG son estadísticamente independientes. El objetivo del método consiste en encontrar una matriz de desmezcla \mathbf{W} de tamaño $P \times N$ que corresponda a $y[m] = \mathbf{W}x[m]$ la cual es una estimación de $s[m]$ [63]. El método ICA necesita una revisión visual de un experto para determinar qué componentes corresponden a artefactos haciendo la tarea dispendiosa, subjetiva y sobretodo no automática. Se han desarrollado umbrales para identificar de

manera automática los artefactos pero se corre el riesgo de eliminar información importante de la señal EEG. Adicionalmente el algoritmo es altamente complejo en desarrollo computacional y su éxito radica en la adecuada identificación de las componentes de artefactos [62].

CCA

El algoritmo CCA es un método es “un método de búsqueda de vectores de base para dos conjuntos de variables tales que las proyecciones de las dos variables sobre esta base están correlacionados al máximo” [63]. Para resolver el problema de BSS, el método BSS-CCA obliga a las señales fuente a tener una máxima correlación [63].

Para tal fin, el método considera dos vectores: la señal adquirida $x[m]$ y la señal adquirida retrasada $z[m] = x[m - 1]$ se construyen dos combinaciones lineales: $b[m] = W_{x_1}x_1[m] + \dots + W_{x_N}x_N[m] = W_x^T x[m]$ y $c[m] = q_{z_1}z_1[m] + \dots + q_{z_N}z_N[m] = q_z^T z[m]$. Finalmente se busca que los vectores W_x y q_z maximicen la correlación ρ entre $b[m]$ y $c[m]$ maximizando la Ecuación 3 en donde R_{xx} y R_{zz} corresponden a las matrices de autocovarianza de x y z respectivamente y R_{xz} la matriz cruzada de varianza entre x y z .

$$\max_{W_x, q_z} \rho(b, c) = \frac{W_x^T R_{xz} q_z}{\sqrt{(W_x^T R_{xx} W_x)(q_z^T R_{zz} q_z)}}$$

La matriz de desmezcla W es el estimado del vector de señales fuente que se encuentra resolviendo la Ecuación 4; en donde los valores propios ρ^2 corresponden a la correlación canónica al cuadrado y los vectores propios W_x son las columnas de la matriz de desmezcla W [63].

$$R_{xx}^{-1} R_{xz} R_{zz}^{-1} R_{zx} W_x = \rho^2 W_x$$

De manera tal que con la ayuda del vector de correlación se pueden determinar las componentes que no están altamente correlacionadas y por lo tanto correspondan a artefactos [63], [64], [65], [66] y [67].

EMD

Es considerado una técnica emergente que divide la señal adquirida en componentes que son estimadas a partir de procesos iterativos y son llamadas Funciones de Modo Intrínseco (IFM por sus siglas en inglés). Si la señal adquirida es $x[m] = \{a_0[m]\}$ el proceso divisorio procede a estimar la k -ésima IFM de la siguiente manera:

1. Se inicia tomando $d_{k,0}[m] = a_{k-1}[m]$ para cada m .
2. Se obtiene la media $\mathcal{M}\{d_{k,j}[m]\}$.
3. Se extraen los detalles $\{d_{k,j+1}[m]\} = \{d_{k,j}[m]\} - \mathcal{M}\{d_{k,j}[m]\}$.
4. Se incrementa j y se repite desde el paso 2 hasta que $\{d_{k,j+1}[m]\}$ es designada como una IMF.

Este método tiene la ventaja de ser de fácil implementación porque el algoritmo no es pesado y existen diferentes técnicas dependiendo de cómo la media es calculada [63].

Métodos de Filtrado Adaptativo

Como se mencionó anteriormente, la implementación de filtros lineales puede causar pérdida de datos importantes de la señal. Sin embargo, el uso de filtros adaptativos minimiza el error debido a que los coeficientes son modificados por un proceso de realimentación que está diseñado para eliminar ruido. La señal deseada no está correlacionada con el ruido ni con la señal de interferencia; de manera que la señal deseada será una estimación obtenida de la resta entre la señal de referencia la cual contiene la realimentación de la señal adquirida. En la Figura 9 se muestra el diagrama del método adaptativo [62].

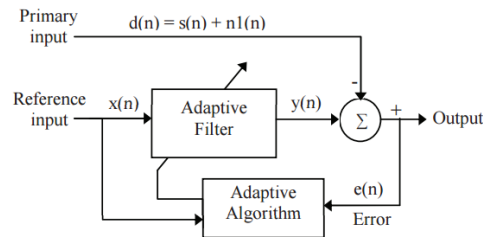


Figura 9. Diagrama del método de filtrado adaptativo [62]

Estado del arte actual

Cada vez existe un mayor desarrollo de los dispositivos BCI que adquieren señales EEG para alguna aplicación específica. Es el caso de Emotiv® quienes desarrollan dispositivos que capturan las señales EEG y las transmiten inalámbricamente a un computador [68]. En el ámbito académico, el Emotiv® ha sido utilizado para diferentes propósitos, por ejemplo el control de un brazo robótico realizando el procesamiento de la señal adquirida a través de análisis lineal discriminante [69], el

control de un brazo robótico utilizando detección y discriminación de eventos además del seguimiento a los movimientos de la cabeza [70] y el control de un videojuego utilizando las señales adquiridas del Emotiv® como comandos de movimiento [71]. Además de realizar aplicaciones, los dispositivos como el Emotiv® son elementos de adquisición que pueden ser parte de un dispositivo BCI de bajo costo [72].

Existen otros dispositivos llamados ThinkGear™ y MindWave™ que son desarrollados por la empresa NeuroSky®, éstos sensores, al igual que el Emotive® adquieren las señales y las transmite inalámbricamente a un computador. Algunas de las aplicaciones de la propia compañía radican en monitoreo de sueño y desarrollos de BCI. Otras aplicaciones han sido desarrolladas por diferentes investigadores, dentro de las que se encuentran, sistemas de control automático de elementos del hogar que se realiza a través del nivel de atención captado por los dispositivos de NeuroSky® [73]. Revisión del nivel de distracción utilizando un juego de computador y un control de juego (timón) analizando las ondas cerebrales obtenidas del dispositivo [74]. Control del robot Festo Robotino con un sistema BCI utilizando el dispositivo de NeuroSky® [75]. Una de las aplicaciones más cercanas al presente proyecto corresponde al control de una prótesis robótica de brazo a través del dispositivo MindWare™ que envía las señales al computador, allí procesa la señal e inalámbricamente envía los comandos de control a la prótesis robótica [76].

Comunidades científicas también han realizado mejoras en el campo de los sistemas BCI, una de las principales corresponde a Open BCI quienes tienen a disposición todos los desarrollos relacionados con los sistemas BCI, el análisis de ondas cerebrales, actividades musculares, cardíacas, entre otros. Y quienes permiten acceder a una diadema de adquisición de señales EEG que puede ser impresa y construida por el usuario para sus desarrollos en el campo [77].

Diferentes investigadores han desarrollado aplicaciones de BCI en sistemas embebidos que permiten un pre-procesamiento de la señal EEG adquirida, es el caso del sistema BCI embebido desarrollado en Romania que fue diseñado netamente con elementos analógicos para realizar el procesamiento con el propósito de ser incluido en sistemas BCI en un futuro [78]. Además se implementó un sistema de adquisición EEG multicanal y seguidor de movimientos de cabeza con la ayuda de amplificadores y filtros análogos para posteriormente realizar la conversión análoga digital y enviar los datos a través de Bluetooth a un computador o a un teléfono inteligente [79].

El propósito principal de los sistemas embebidos es realizar dispositivos costo efectivos, es el caso del sistema de adquisición que registra las señales en un computador a través de un procesador

digital de señales (DSP por sus siglas en inglés) [80]. Otro de los propósitos de estos sistemas es que sea portable, como el dispositivo desarrollado en una universidad en Malasia que utiliza elementos análogos para adquirir la señal de un canal con dos electrodos activos [81].

Finalmente, uno de los estudios más próximos al presente trabajo corresponde a una tesis de maestría de Commonwealth de Virginia en la cual se desarrolla un sistema de adquisición EEG compacto e inalámbrico de bajo costo, allí utilizan el integrado ADS1299 para adquirir la señal, un microcontrolador Atmel 32 para realizar la comunicación con el integrado que adquiere y un módulo inalámbrico para enviar las señales al computador [82].

Como se puede observar los sistemas BCI en su mayoría utilizan el computador para realizar la tarea de procesamiento de las señales, en la actualidad los sistemas de adquisición han sido el centro de las investigaciones, así como también la transformación de dichas señales en comandos de control; sin embargo, es importante que el procesamiento de la información comience a ser realizado a través de sistemas embebidos y que dependan cada vez menos de un computador, como es el caso de los sistemas embebidos mencionados anteriormente.

Adicionalmente, los sistemas se enfocan en ámbitos visuales, movimiento de cabeza, seguimiento de ojos y movimiento de miembros superiores, no obstante, el movimiento de miembros inferiores ha sido poco investigado debido a que la señal es difícil de captar por su localización en la corteza cerebral y su contaminación con señales aledañas, en el Homúnculo de Peindfield (Figura 3) se observa que la parte motora de miembros inferiores se encuentra en el surco central y no es superficial sino que la se encuentra adentro en el cráneo.

METODOLOGÍA

El presente estudio muestra una investigación experimental aplicada. En este apartado se plantea la metodología que se implementa en el proyecto con el fin de cumplir con los objetivos propuestos.

El trabajo se desarrolla en cuatro fases. La fase 1 corresponde a un procesamiento de las señales de la base de datos de Physionet con el fin de verificar los resultados del método elegido para eliminar artefactos oculares. En la fase 2 se realiza la adquisición y procesamiento de señales EEG con la tarjeta de evaluación para terminar de ajustar la técnica del procesamiento de las señales. Posteriormente en la fase 3 se realiza la implementación del dispositivo en una tarjeta de tamaño reducido y finalmente en la fase 4 se realizan las pruebas de funcionamiento.

De manera tal que al terminar las cuatro fases se cuenta con el diseño de un dispositivo electrónico portátil y compacto que realice una adquisición y procesamiento de las señales EEG de miembro inferior con el fin de que en trabajos futuros se realice la caracterización y clasificación de la señal.

Fase 1. Procesamiento de las señales de la base de datos de Physionet.

Para cumplir con los objetivos del trabajo se requiere de un análisis de señales de EEG que arroje una aproximación inicial de los resultados que se esperan en la adquisición de señales con el dispositivo que se va a diseñar. Por lo tanto, el análisis inicial se realiza utilizando la base de datos EEG Motor Movement/Imagery Dataset (eegmmidb) que fue creada por los desarrolladores del sistema de instrumentación BCI2000 [83].

Señales de la base de datos de Physionet.

La base de datos contiene más de 1500 registros obtenidos de 109 voluntarios. Se registraron 64 canales de EEG siguiendo el sistema internacional 10-20. Cada uno de los sujetos desarrolló 14 tareas que duraron un minuto en las tareas de reposo y dos minutos en las tareas de movimiento o imaginación de movimientos. Cada uno de los registros cuenta con el archivo de anotaciones en donde se detalla el número de muestra en el cual se realizan movimientos.

Las tareas fueron las siguientes:

1. Reposo con ojos abiertos
2. Reposo con ojos cerrados
3. Tarea 1: Abrir y cerrar el puño derecho o izquierdo
4. Tarea 2: Imaginar abrir y cerrar el puño derecho o izquierdo
5. Tarea 3: Abrir y cerrar los dos puños o los dos pies
6. Tarea 4: Imaginar abrir y cerrar los dos puños o los dos pies
7. Tarea 1.
8. Tarea 2.
9. Tarea 3.
10. Tarea 4.
11. Tarea 1.
12. Tarea 2.
13. Tarea 3.
14. Tarea 4.

Con la ayuda de las herramientas de Physionet es posible descargar los archivos de las señales en formato *.mat que pueden ser abiertos como matrices en Matlab® y los archivos de las anotaciones en formato *.txt para su visualización [83].

Procesamiento de las señales de la base de datos de Physionet

El procesamiento de las señales se realiza de la siguiente manera: Con la ayuda de la función “rand” del software Matlab®, se eligen aleatoriamente 5 individuos dentro de la base de datos y se realiza el procesamiento primero utilizando la información de los 64 electrodos y después utilizando la información de los electrodos que se medirán en la fase 2, es decir, los electrodos Fp1, Fp2, F3, F4, C3 y C4. Los electrodos son elegidos debido a que se desea captar las señales de movimiento de miembro inferior y movimientos oculares, gracias al homúnculo de Peindfield (Figura 3) se puede determinar que los electrodos C3 y C4 son los que contienen mayor información de miembros inferiores y los electrodos Fp1, Fp2, F3 y F4 se utilizan para referenciar movimientos oculares.

El procesamiento se inicia con la implementación del algoritmo de reducción de artefactos oculares BSS-CCA, posteriormente se identifican y eliminan los artefactos y se reconstruye la señal.

Finalmente se aplica un filtro pasa bandas con frecuencias de corte 0,5 Hz y 35 Hz, esto con el fin de eliminar cualquier componente DC y reducir la incidencia de los artefactos musculares de altas frecuencias así como también la eliminación de la frecuencia de la red eléctrica.

Lo que se observa en la revisión bibliográfica acerca de la reducción o remoción de artefactos oculares es que los investigadores utilizan métodos basados en ICA [84] [85] [86] [87] [88] [89] [90] [91], modelos adaptativos [92] [93], métodos de reducción por Wavelet [87] [88] [89] [94] [95], EMD [96] [97] [91], métodos auto-regresivos [98] [99] y CCA [66]. Sin embargo, estos algoritmos son implementados en estudios que cuentan con un computador para procesar los datos. La utilización de sistemas BSS con ICA o Wavelets implica un alto nivel de procesador y de memoria, es por tal motivo que al implementarlos mediante un software como Matlab® en un computador, los tiempos de procesamiento no son muy relevantes. No obstante, el objetivo de este trabajo es que el dispositivo sea compacto y portátil por lo cual se utilizará un microcontrolador para el procesamiento. Por lo anterior, el algoritmo debe ser de fácil implementación, bajo costo computacional y no debe requerir funciones demasiado complejas, con el fin de realizar un análisis con un bajo tiempo de procesamiento.

Para realizar la eliminación de artefactos oculares se elige el método BSS-CCA debido a que es uno de los métodos que tienen un costo computacional bajo y es de fácil implementación después de contar con librerías para manejo matricial. Adicionalmente, se intenta dar mayor importancia a un método que se basa en un principio estadístico que arroja la correlación entre las componentes de la señal y por lo tanto permite tomar decisiones acerca de la información útil o no de manera automática sin depender de expertos que indiquen anomalías.

Fase 2. Adquisición y procesamiento de señales EEG

Para evaluar la señal EEG, es necesario tomar muestras propias en donde se conozcan las condiciones de adquisición. Al tratarse de un estudio piloto se registran las señales EEG de 5 individuos con el protocolo que se describe más adelante y se procesan de la misma manera que en la fase 1 a través del software Matlab® para su posterior análisis.

Tarjeta de evaluación y demostración ADS1299EEG-FE

Las señales son captadas con la tarjeta de evaluación y demostración ADS1299EEG-FE que contiene el integrado ADS1299. Éste integrado es un conversor análogo-digital de bajo ruido, multicanal, de 24 bits con un amplificador de ganancia programable. Tiene un multiplexor por canal que puede ser conectado a señales internamente generadas para pruebas, temperatura y detección de lead off. Las muestras pueden ser tomadas a una frecuencia de muestreo desde 250 hasta 16000 muestras por segundo [100]. Dicha tarjeta de evaluación cuenta con dos tarjetas, una que corresponde a la configuración del integrado ADS1299 y la otra corresponde a una tarjeta madre nombrada como MMB0 que contiene un integrado que ejecuta toda la programación y comunicación entre el integrado y el software de adquisición en el computador.

La tarjeta de evaluación permite hacer una revisión completa acerca del funcionamiento del integrado. Para tal fin, se dispone de una interfaz gráfica en la que es posible visualizar las señales adquiridas, realizar análisis en frecuencia y programar el integrado en línea modificando los registros. La interfaz gráfica se muestra en la Figura 10 [101].

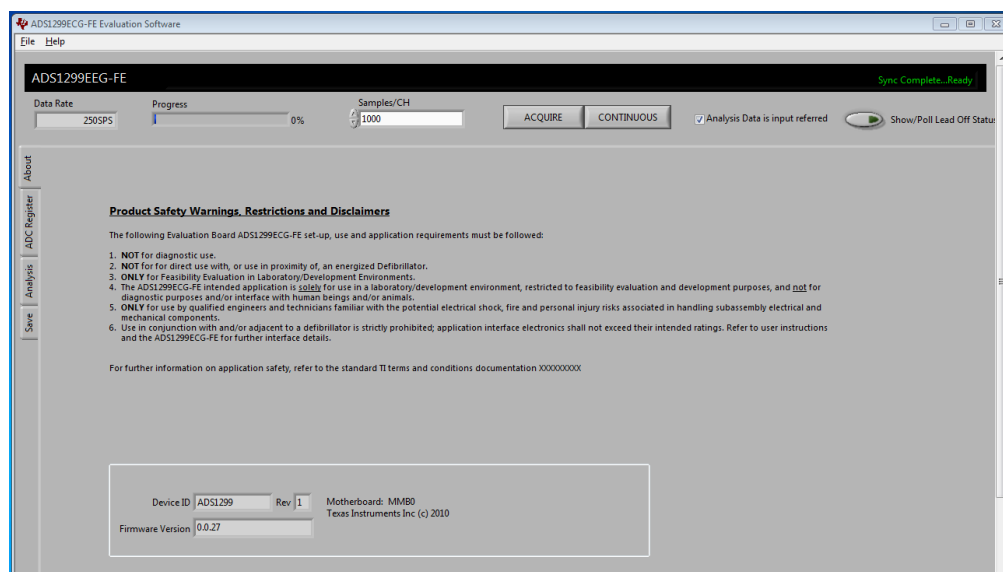


Figura 10. Interfaz gráfica de la Tarjeta de Evaluación

La configuración de los registros en la tarjeta de evaluación se realiza por medio de la localización de los jumpers y por la configuración en el software. En la Figura 11 se muestra la tarjeta de evaluación. La posición de los jumpers por defecto se presenta en el Anexo 1.



Figura 11. Tarjeta de Evaluación del integrado ADS1299

En la Figura 12 se muestra el diagrama de bloques de la tarjeta de evaluación con los jumpers más importantes.

Como se observa en la Figura 12, el J6 es el encargado de captar las señales de los electrodos, el J25 contiene el electrodo BIAS y también el electrodo de referencia, posteriormente estas señales pasan por un filtro pasa bajos y entran al integrado. Los jumpers J6, J7 y J8 son los encargados de configurar el buffer para el electrodo BIAS, configurar el buffer para el electrodo de referencia y permitir la entrada del electrodo de referencia al integrado respectivamente. El J4 se encarga de la conexión a la alimentación, el J3 permite incluir una referencia de voltaje externa y el J18 permite configurar la tarjeta para trabajar con un oscilador externo.

Para la toma de muestras se realiza una configuración de señales diferenciales en vez de la configuración de señales únicas. Lo anterior, debido a que se desea contar con las señales de los electrodos con respecto a un punto de referencia pero no comparar las señales de los electrodos entre sí. Para tal fin, en la tarjeta de desarrollo se realiza un puente entre todas las entradas negativas de los electrodos y el electrodo de referencia se ubica en la referencia que internamente conecta todas las entradas negativas. Esto genera una señal diferencial entre la señal de entrada positiva de cada electrodo y la señal de referencia.

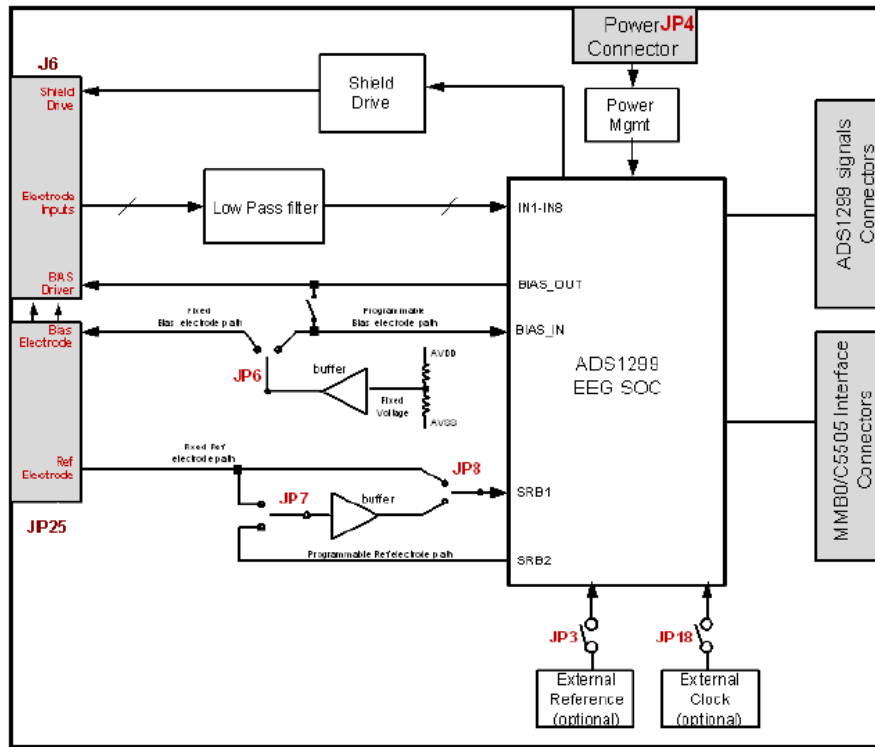


Figura 12. Diagrama de bloques de la Tarjeta de Evaluación

Adicionalmente, la conexión de los electrodos se realiza implementando el montaje que se muestra en la Figura 13.

Protocolo de adquisición de las señales EEG

Para adquirir las señales con la tarjeta de evaluación se ajusta la configuración para la toma de señales únicas y se registra la información de la señal EEG proveniente de los electrodos Fp1, Fp2, F3, F4 C3 y C4 según el sistema internacional 10-20.

La frecuencia de muestreo es 250 muestras por segundo y se cumple con el siguiente protocolo de adquisición:

1. Se le pide al participante que adopte la postura semisedente y se solicita un máximo de relajación física y mental.
2. Se limpia el cuero cabelludo con alcohol en los lugares en donde se colocan los electrodos.
3. Los electrodos se ubican según el sistema internacional 10-20 en Fp1, Fp2, F3, F4, C3 y C4 para EEG.

4. Los electrodos se fijan al con pasta conductora y se coloca sobre ellos cinta de esparadrapo.
5. El electrodo de referencia se ubica en el lóbulo de la oreja izquierda del participante.
6. El registro de la señal EEG se realiza a partir de las siguientes tareas:
 1. Treinta segundos de relajación con ojos abiertos.
 2. Treinta segundos de extensión y flexión de rodillas sin limitar movimientos oculares o parpadeo.

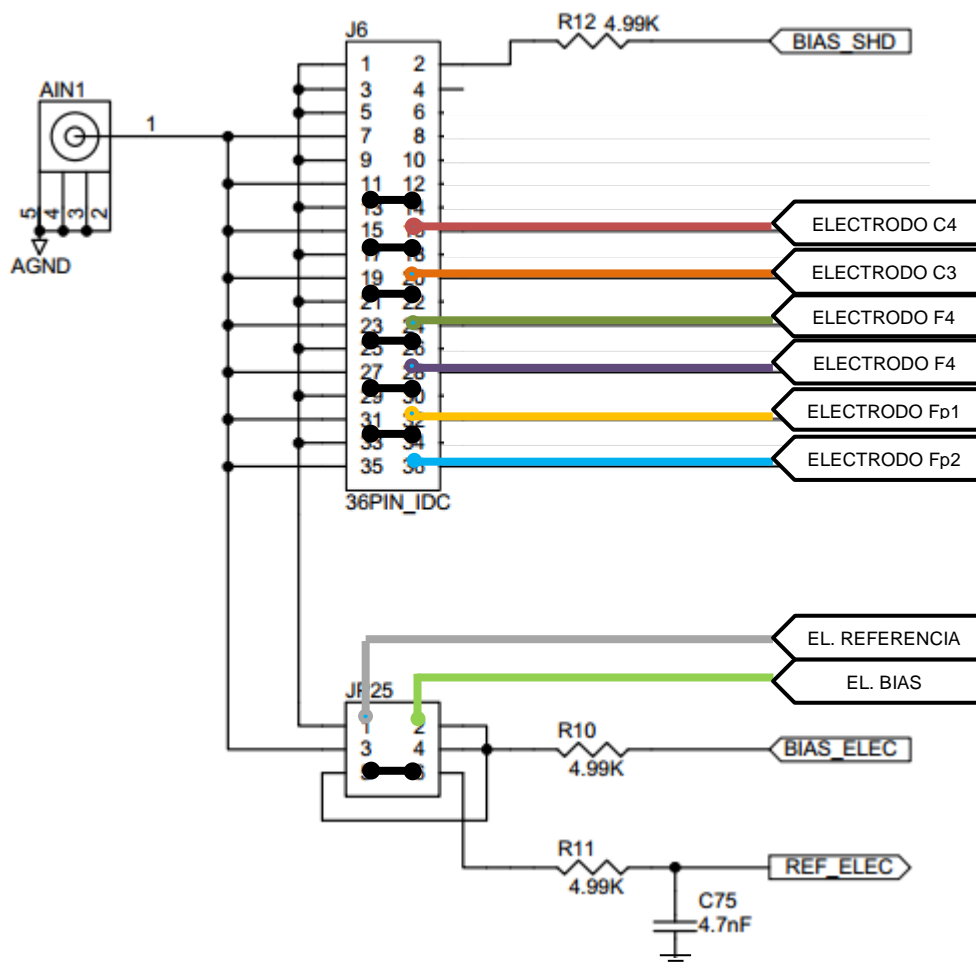


Figura 13. Montaje de los electrodos para captar las señales.

Procesamiento de las señales adquiridas

Después de crear la base de datos de las 5 señales de EEG se implementa el procesamiento desarrollado en la fase 1 utilizando el algoritmo BSS - CCA por medio del software Matlab®. De manera tal que se puedan analizar los resultados y tomar decisiones acerca del procesamiento que se implementará en el dispositivo electrónico.

Fase 3. Implementación del dispositivo

El diseño y la implementación del dispositivo se divide inicialmente en cuatro partes: adquisición, procesamiento, visualización y placa PCB.

Adquisición: Como se mencionó anteriormente la adquisición se realiza por medio del integrado ADS1299 a través de electrodos de superficie de copa con recubrimiento de oro.

Procesamiento: Consecuentemente con los resultados de la implementación de los algoritmos de reducción de artefactos oculares se busca un microprocesador con alta tasa de procesamiento y capacidad de memoria.

Visualización: Debido a que el fin último del dispositivo es dejar la señal lista para la caracterización y clasificación, la visualización de la señal adquirida y procesada se realiza en un computador a través de un protocolo UART.

Placa PCB: Después de definir todos los dispositivos electrónicos y comprobar su funcionamiento. Se diseña la placa PCB con dispositivos electrónicos de superficie, adicionalmente se diseña el circuito para que el sistema pueda ser alimentado a través de baterías.

Fase 4: Pruebas de funcionamiento

Las pruebas de funcionamiento se realizan tomando una señal EEG con el dispositivo diseñado, guardando los datos originales y realizando el procesamiento con el dispositivo y con el software Matlab®. Tanto la señal procesada en el dispositivo diseñado como procesada con el software se observan en el computador con el fin de comparar la salida de los dos procesamientos.

RESULTADOS Y CONTRIBUCIÓN

Fase 1. Procesamiento de las señales de la base de datos de Physionet.

Adquisición de las señales de la base de datos de Physionet

Se eligieron 5 individuos de manera aleatoria de la base de datos EEG Motor Movement/Imagery Dataset (eegmmidb) y se descargaron las tareas correspondientes al movimiento de ambas muñecas o tobillos, a saber, los registros 5, 9 y 13. Con la ayuda de las herramientas de Physionet, cada uno de los archivos fue descargado en formato *.mat generando matrices en el entorno Matlab®. La escogencia aleatoria con la ayuda de la función “rand” en el software Matlab® arrojó la selección de los individuos: 5, 15, 57, 65 y 85.

En el experimento realizado por los investigadores de la base de datos, se realizó un registro de 64 electrodos por un periodo de tiempo aproximado de 200000 segundos, es por esta razón que cada una de las matrices tiene aproximadamente 64x20000 elementos. Cabe resaltar que al abrir los archivos descargados se encontró una columna adicional que corresponde a las anotaciones realizadas en cada registro, sin embargo, el archivo de las anotaciones fue descargado en formato *.txt para facilitar su lectura e interpretación.

En la Figura 14 se muestra uno de los registros completos de 64 electrodos obtenidos de la base de datos, en este caso corresponde al individuo número 15 realizando la tarea 5 (elegido sin ninguna particularidad especial, únicamente para efectos de ejemplo), se muestran trescientos registros para efectos de visualización excepto cuando se muestran las componentes de las señales, allí se observa la totalidad de la señal.

Procesamiento de las señales con 64 canales de EEG

En el Anexo 2 se muestra el código completo del procesamiento en el software Matlab®. Se implementó el algoritmo BSS-CCA para eliminar los artefactos oculares utilizando el código del método BSS-CCA (Anexo 3). De la implementación de dicho algoritmo se obtienen como resultado: las componentes de la señal $y(t)$, una matriz de desmezcla W y un vector de correlación ρ en el que se observa que las componentes son organizadas de mayor a menor correlación.

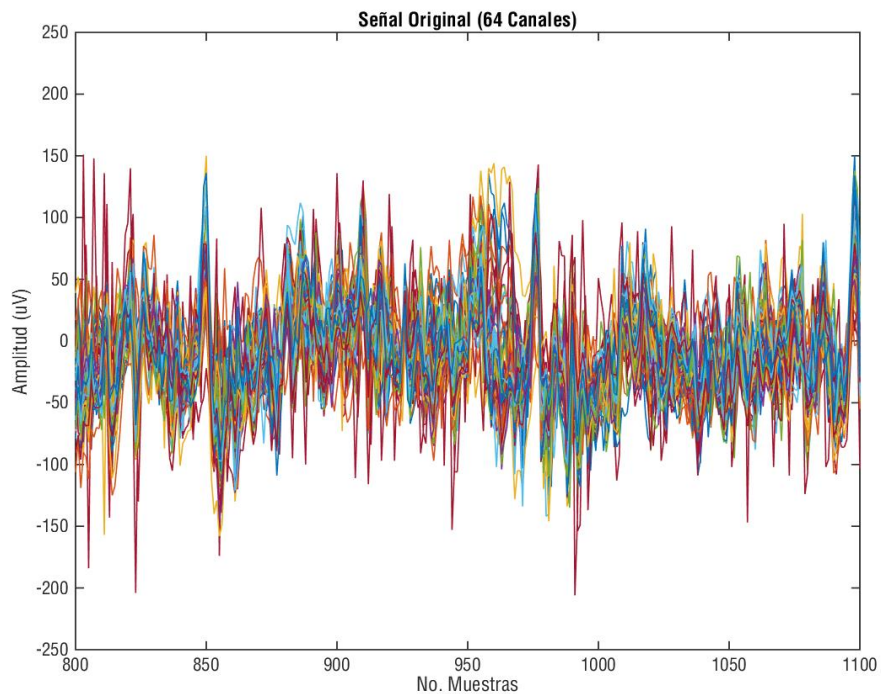


Figura 14. Señal Original 64 Canales

La cantidad de componentes que arroja el método son exactamente la misma cantidad de canales disponibles en el análisis. Se procedió a eliminar las componentes que contenían artefactos oculares, apoyándose en la literatura se estableció el umbral de correlación y se decidió que las componentes que obtuvieran una correlación inferior a 0.75 serían consideradas como artefactos oculares.

Para el ejemplo, el vector de correlación obtenido después de aplicar el método BSS-CCA se muestra en la Figura 15 y según el umbral se eliminan las componentes a partir de la número 50. Después de lo anterior se reconstruyó la señal como se muestra en la Figura 16 y finalmente, se implementó un filtro pasa bandas con frecuencias de corte 0,5 Hz y 35 Hz como se observa en la Figura 17.

No. Elemento	ρ	No. Elemento	ρ	No. Elemento	ρ	No. Elemento	ρ
1	0.9925	21	0.9313	41	0.8548	61	0.3927
2	0.9862	22	0.9296	42	0.8470	62	0.3659
3	0.9800	23	0.9280	43	0.8352	63	0.3092
4	0.9732	24	0.9263	44	0.8335	64	0.1730
5	0.9682	25	0.9249	45	0.8276		
6	0.9641	26	0.9221	46	0.8136		
7	0.9580	27	0.9197	47	0.7996		
8	0.9556	28	0.9170	48	0.7887		
9	0.9539	29	0.9135	49	0.7633		
10	0.9525	30	0.9124	50	0.7439		
11	0.9509	31	0.9089	51	0.6791		
12	0.9487	32	0.9054	52	0.6721		
13	0.9471	33	0.9022	53	0.6290		
14	0.9417	34	0.8975	54	0.6198		
15	0.9405	35	0.8949	55	0.6127		
16	0.9403	36	0.8888	56	0.6023		
17	0.9385	37	0.8855	57	0.5552		
18	0.9373	38	0.8808	58	0.5201		
19	0.9344	39	0.8749	59	0.4698		
20	0.9337	40	0.8655	60	0.4586		

Figura 15. Elementos del vector de correlaciones 64 canales

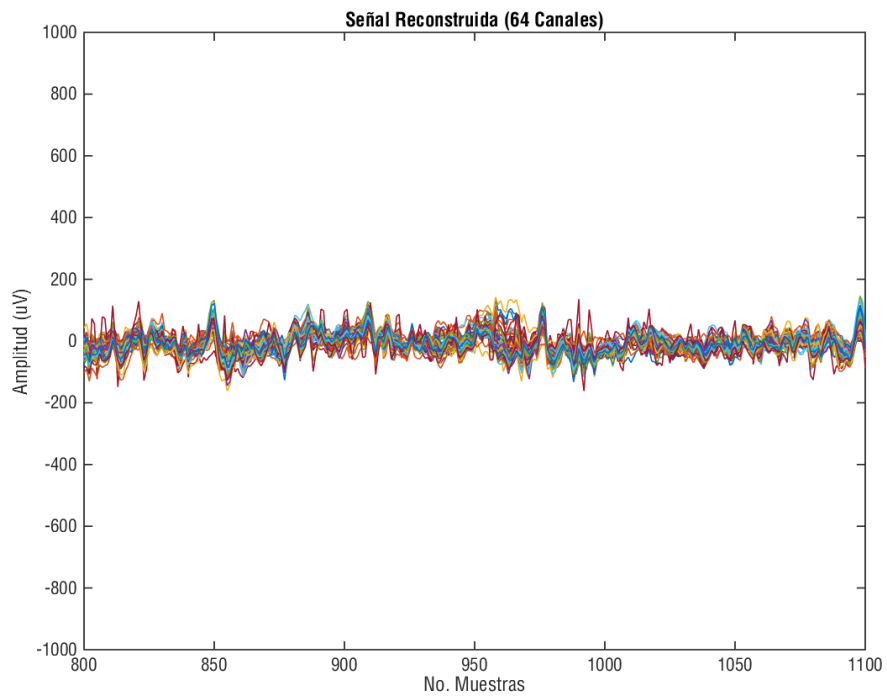


Figura 16. Señal Reconstruida 64 Canales

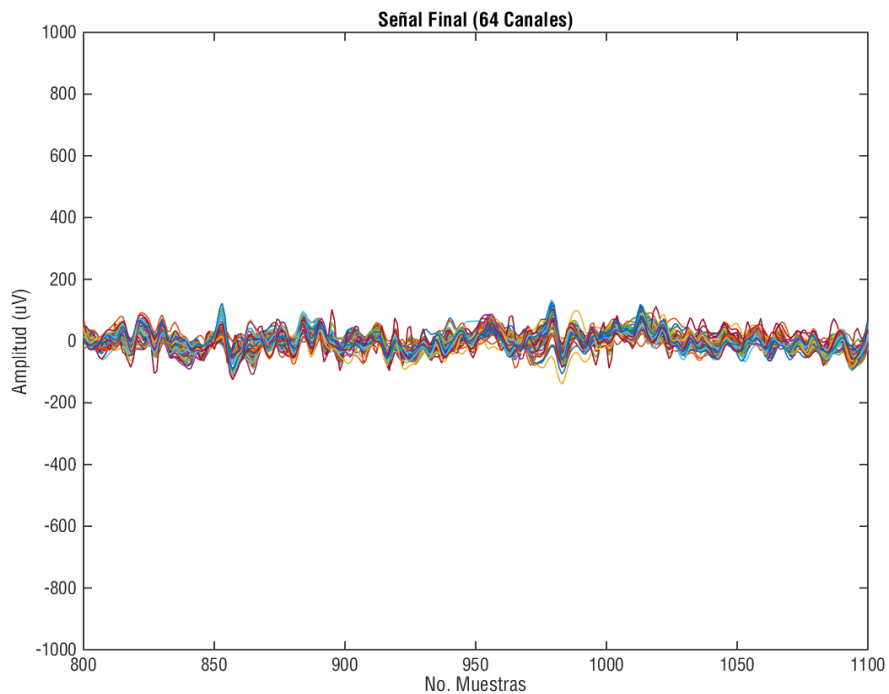


Figura 17. Señal Final 64 Canales

Procesamiento de las señales con 6 canales de EEG

La segunda parte del procesamiento se realizó segmentando los electrodos que se toman en cuenta para la adquisición con la tarjeta de evaluación, es decir con los electrodos Fp1, Fp2, F3, F4, C3 y C4. Para obtener únicamente las señales de los electrodos deseados, se remitió a la descripción de la base de datos en donde se encuentra especificado el número de columna de la matriz que corresponde a determinado electrodo. Para este caso se extrajeron las columnas según el electrodo así:

- Electrodo Fp1, columna 22.
- Electrodo Fp2, columna 24.
- Electrodo F3, columna 32.
- Electrodo F4, columna 36.
- Electrodo C3, columna 9.
- Electrodo C4, columna 13.

En la Figura 18 se observa la señal original seleccionando únicamente la información de los 6 electrodos.

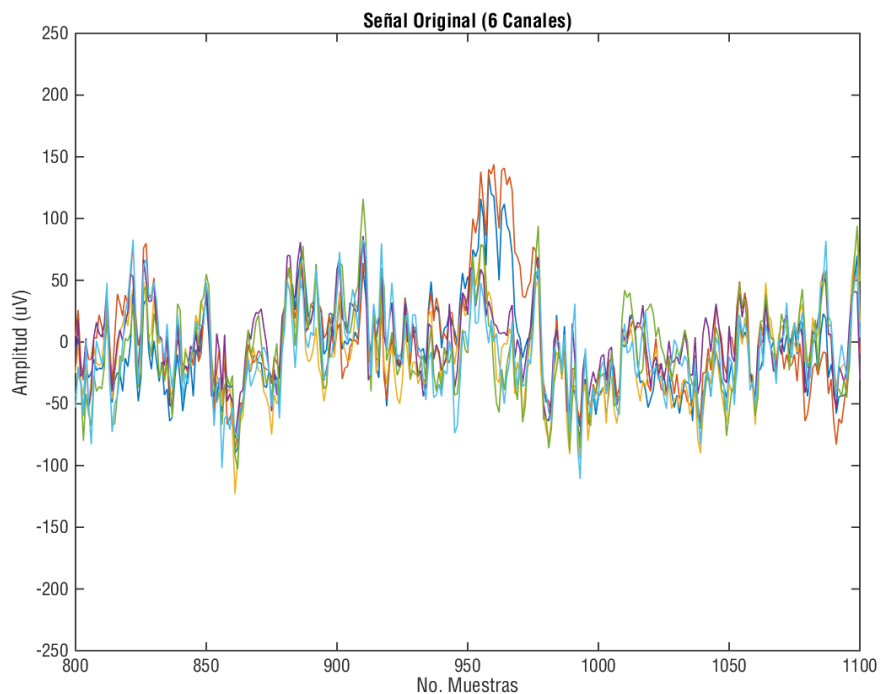


Figura 18. Señal Original 6 Canales

Se realizó el mismo procesamiento explicado anteriormente: implementación del algoritmo BSS-CCA, eliminación de artefactos oculares por umbral, reconstrucción de la señal y filtrado de 0,5 a 35 Hz. En la Figura 19 se muestran las componentes después de la implementación del código, en la Figura 20 se muestra el vector de correlaciones en donde se observa que únicamente la componente número 6 es eliminada, posteriormente se muestra en la Figura 21 la señal reconstruida y finalmente la señal filtrada se muestra en la Figura 22.

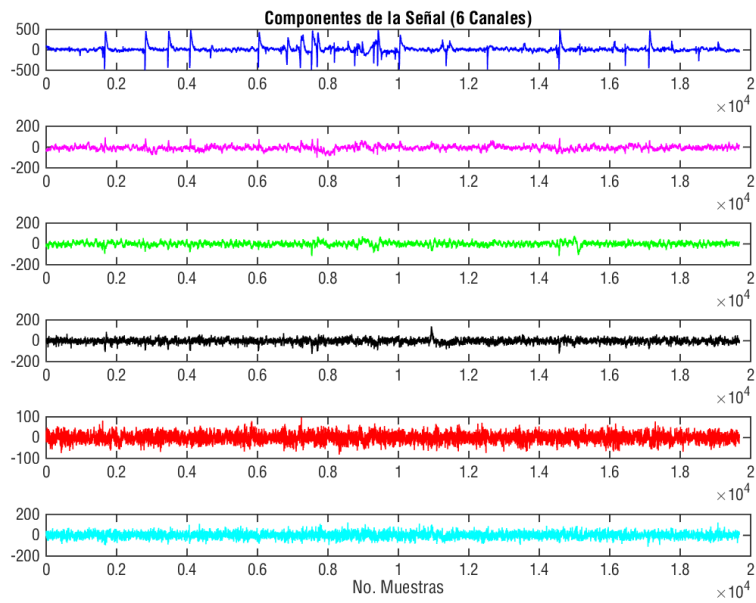


Figura 19. Componentes de la señal 6 canales.

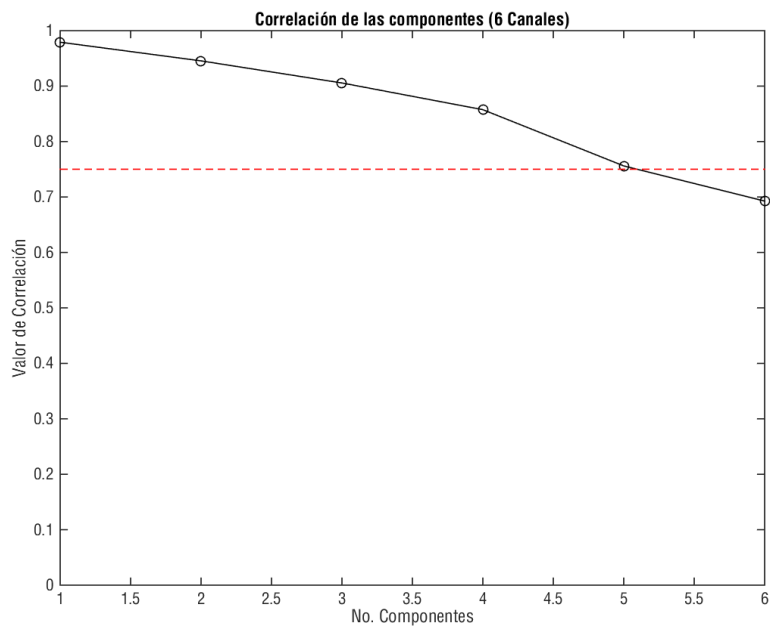


Figura 20. Correlación de las componentes 6 canales

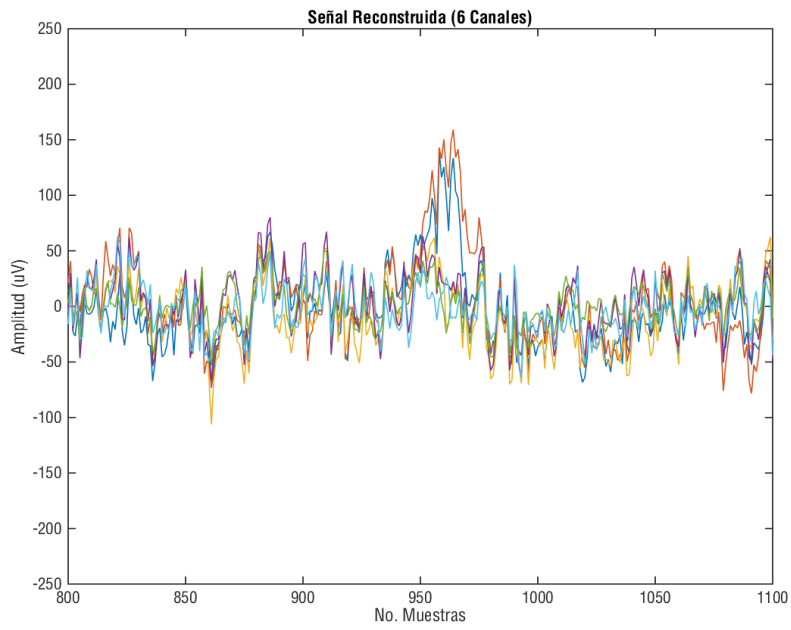


Figura 21. Señal Reconstruida 6 Canales

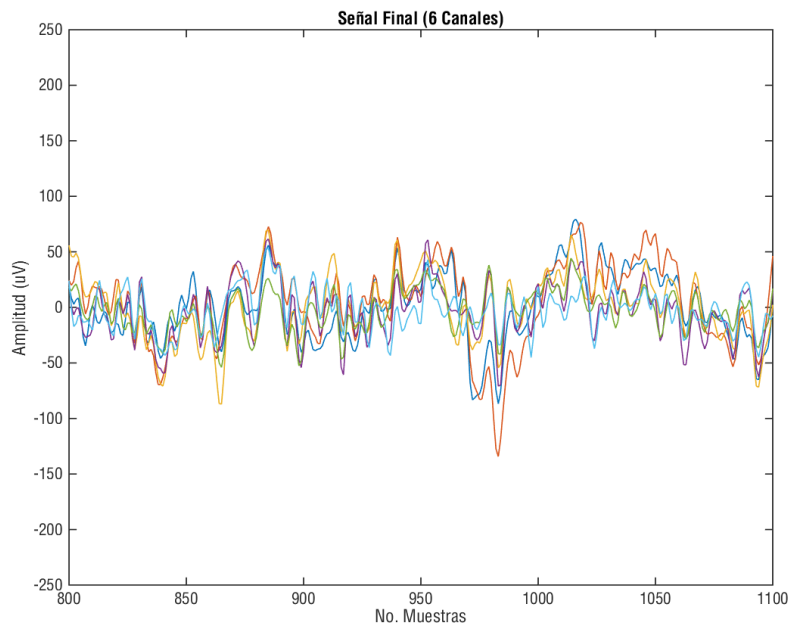


Figura 22. Señal Final 6 Canales

En la Figura 23 se muestra la comparación de la señal original con respecto a la señal reconstruida después de la eliminación de artefactos. La figura muestra únicamente un canal (canal 5) para efectos de visualización.

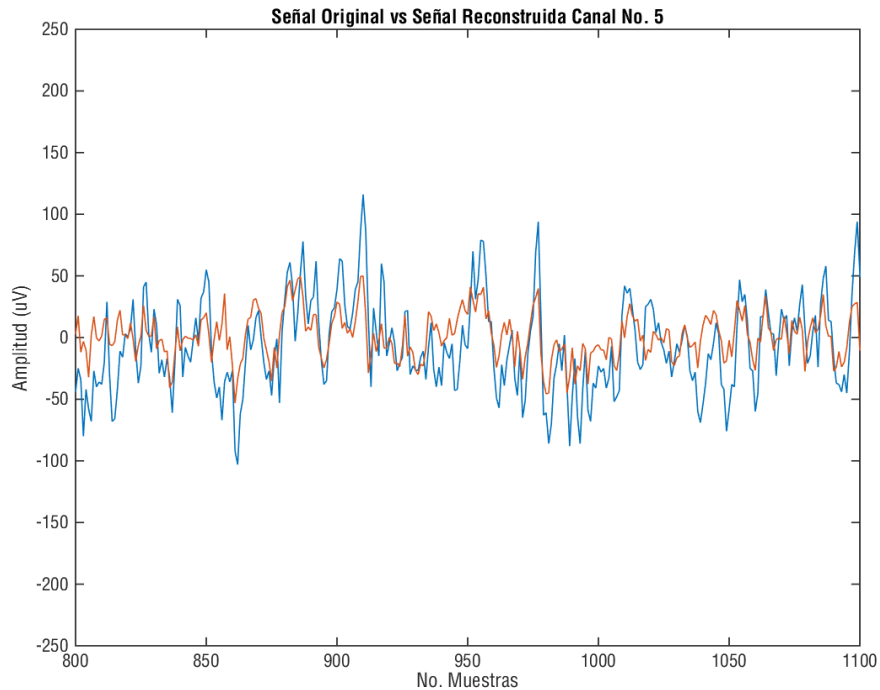


Figura 23. Señal original vs señal reconstruida canal 5.

El análisis anterior se realizó para cada uno de los registros descargados de la base de datos. En cada una de las señales se implementó el algoritmo BSS-CCA y se utilizó el umbral para eliminar los artefactos oculares. Las señales originales, los resultados de los vectores de correlación, las señales reconstruidas y las señales filtradas de cada uno de los registros elegidos aleatoriamente se encuentran en el Anexo 4.

Después de obtener el procesamiento de los dos escenarios se puede concluir que la implementación del algoritmo BSS-CCA arrojó resultados favorables al presentar la señal en 6 componentes organizadas de mayor a menor con respecto a la correlación entre ellas. Así mismo, la utilización del umbral propuesto permite eliminar artefactos oculares sin eliminar información cerebral importante. En el análisis de 64 canales se observó una mayor actividad por parte del umbral, es

decir que se eliminan mayor cantidad de componentes en lo que respecta al análisis de 6 canales, allí a se eliminaron normalmente 1 y 2 componentes en las pruebas realizadas.

Lo anterior, permite determinar que entre mayor cantidad de electrodos exista, es mejor el filtrado no solo de artefactos oculares sino de otras fuentes de artefactos pues se garantiza una correlación entre las componentes puramente cerebrales.

Otro punto importante es la comparación de las señales originales con relación a las reconstruidas, como se puede observar en la Figura 23 y en las figuras del Anexo 4, la señal reconstruida se ve más suave y los picos de voltaje junto con la amplitud general de la señal son más bajos. La señal se ve atenuada y sin cambios abruptos entre muestras. Adicionalmente, se observa cómo la esencia de la señal no se ve afectada pues realiza un seguimiento fiel de la señal original con eliminación de artefactos oculares.

Fase 2. Adquisición de señales EEG.

Para obtener una base de datos utilizando el integrado ADS1299 se procedió a configurar la tarjeta de evaluación y a seguir el protocolo de medición con el fin de contar con datos propios para realizar el procesamiento descrito en la Fase 1.

Montaje y desarrollo del experimento

La tarjeta de desarrollo se configuró a una tasa de muestreo de 250 muestras por segundo. Los electrodos fueron conectados de acuerdo al montaje mostrado en la Figura 13. En la realización del experimento se determinó que un electrodo de BIAS era necesario para la adecuada medición, por lo tanto en el montaje se adicionó en la entrada correspondiente. En la Figura 24 se muestra una fotografía del montaje del experimento.

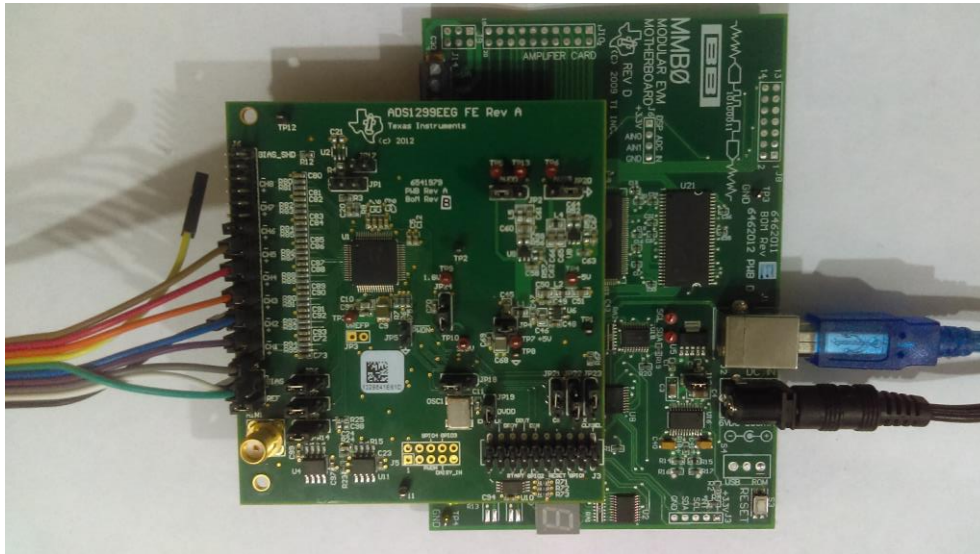


Figura 24. Montaje del experimento en la tarjeta de evaluación.

Después de realizar el montaje para la toma de mediciones se le solicitó al primer individuo que tomara asiento, a partir de allí se implementó el protocolo de medición de la señal EEG como se mencionó en el apartado anterior. Al terminar cada una de las mediciones se guardó el archivo en formato *.xls para su análisis.

Después de terminar con las mediciones, se realizó una revisión visual de las señales adquiridas para cada uno de los individuos con el fin de verificar su utilidad; finalmente los electrodos se limpiaron y se dispuso el montaje de nuevo para la siguiente medición.

Al terminar el experimento se obtuvieron 5 señales de EEG de 6 canales cada una. En la Figura 25 se muestra una de las señales obtenidas de EEG correspondiente al individuo 1 realizando movimiento de extremidad inferior.

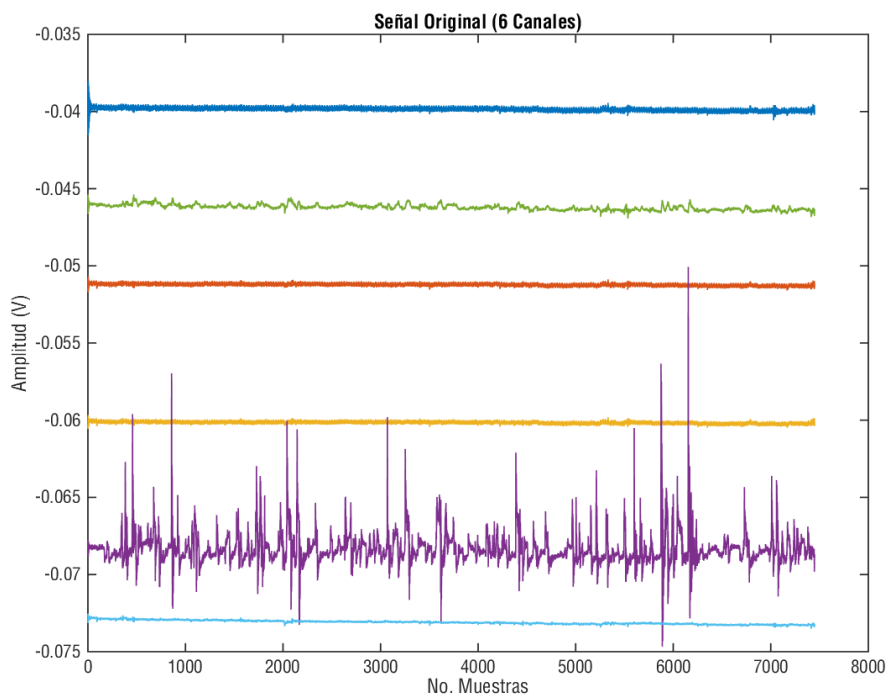


Figura 25. Señal adquirida con la tarjeta de evaluación.

Procesamiento de las señales adquiridas

Las señales adquiridas son guardadas como matrices de 6 columnas correspondientes a cada uno de los canales por 7500 muestras que corresponden a 30 segundos de señal y se encuentran consignadas en un archivo con extensión *.xls, por lo que inicialmente se realizó la lectura del archivo y se formó una matriz correspondiente a la matriz inicial de 6 por 7500 llamada X . Se procedió a realizar el mismo procesamiento descrito en la Fase 1: implementación del método BSS-CCA, eliminación de artefactos, reconstrucción y filtrado de la señal para cada una de las 10 señales adquiridas (2 por individuo), el código de procesamiento en Matlab® se encuentra en el Anexo 5.

A continuación se presenta el resultado del procesamiento de una de las señales adquiridas (individuo 5 elegido únicamente para cuestiones de ejemplo). Se muestra la señal original (Figura 26), las componentes (Figura 27), los elementos del vector de correlaciones (Figura 28), la señal reconstruida (Figura 29) y la señal filtrada (Figura 31). Adicionalmente, se presentan las figuras comparando las señales originales con las reconstruidas de un único canal, canal 5 para efectos de ejemplo en la Figura 30. En el Anexo 6 se muestran los resultados de las 5 señales procesadas.

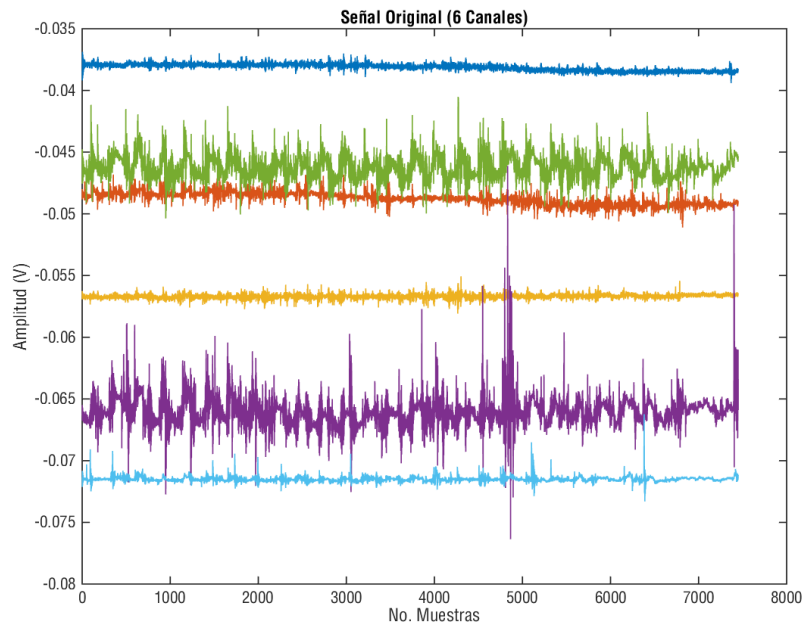


Figura 26. Señal Original Adquisición con Tarjeta de Evaluación

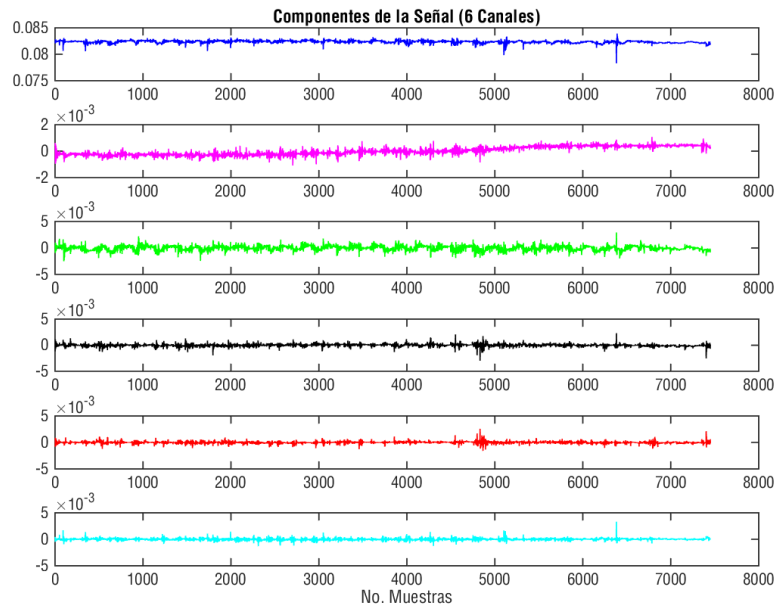


Figura 27. Componentes de la señal adquirida con la Tarjeta de Evaluación.

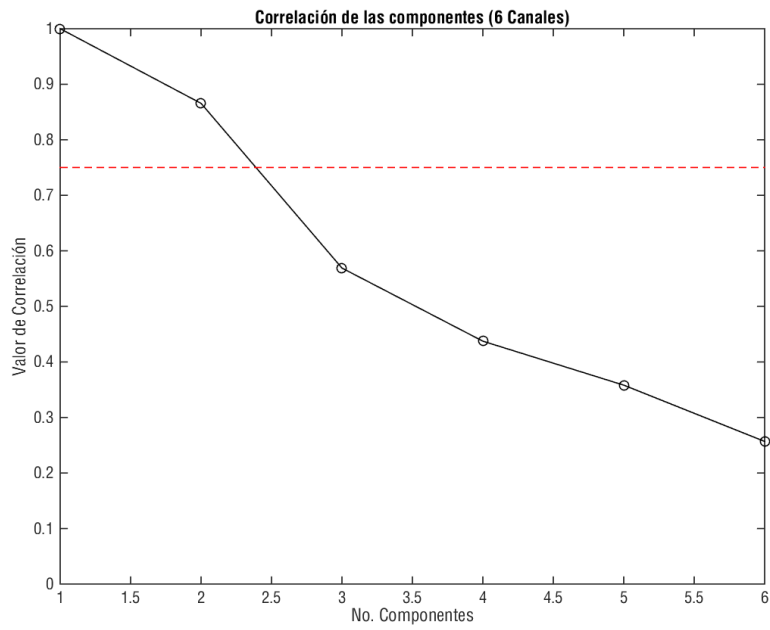


Figura 28. Elementos del vector de correlación

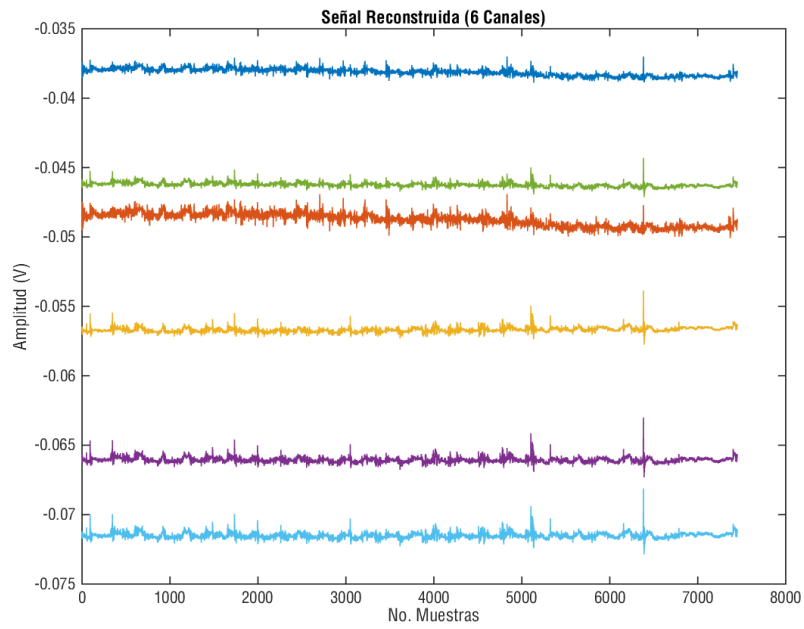


Figura 29. Señal reconstruida de la adquirida con la Tarjeta de Evaluación

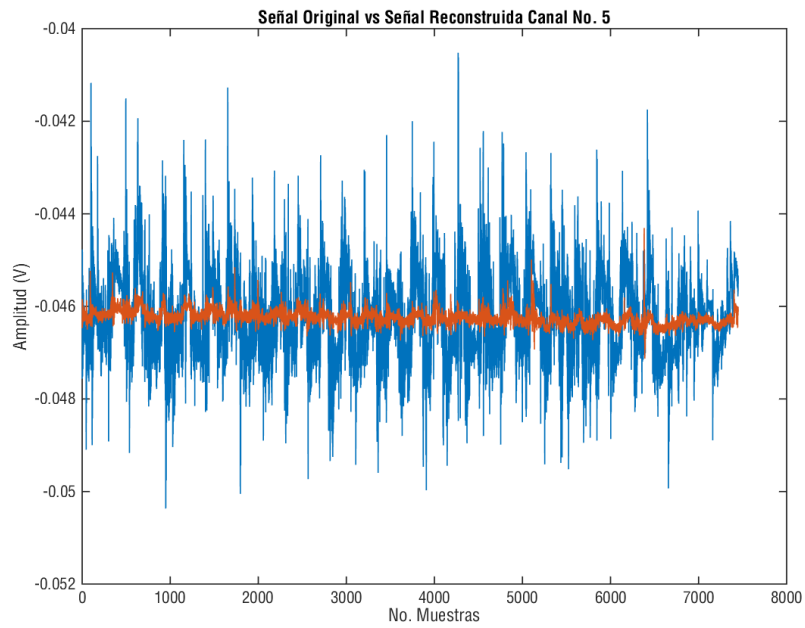


Figura 30. Señal Reconstruida del canal 5.

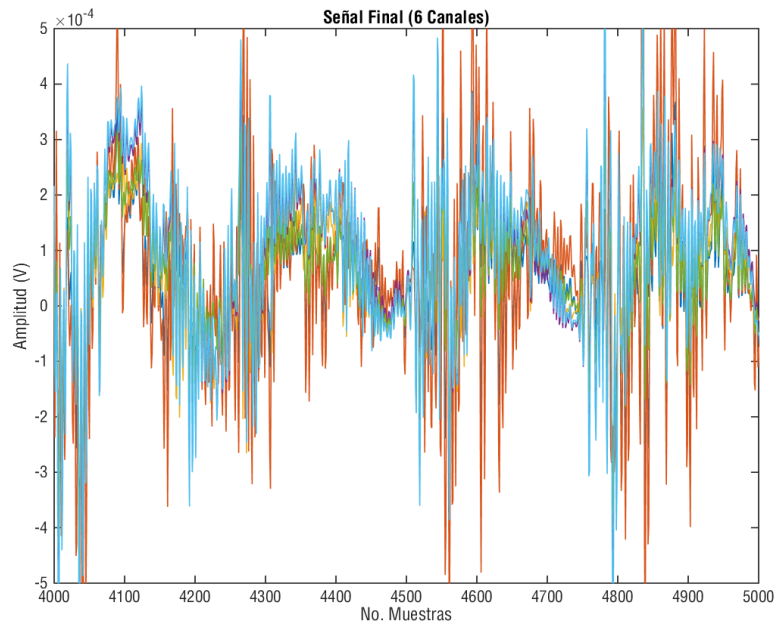


Figura 31. Señal filtrada de la adquirida por la Tarjeta de Evaluación

Al terminar esta fase se pudo observar que el procesamiento arrojó mejores resultados para las señales de la base de datos. Lo anterior debido a que probablemente el instrumento utilizado para tomar las señales contiene filtros anti ruidos que ayudan a suavizar la señal. Mientras que el integrado ADS1299 permitió obtener la señal pura únicamente amplificada sin ningún pre procesamiento.

No obstante, la implementación del procesamiento en las señales arrojó resultados positivos en cuanto a la utilidad del método que elimina artefactos oculares. Como se observa en la señal original en comparación con la señal reconstruida se ve claramente una reducción en artefactos y en ruidos. En este caso se eliminaron 4 componentes, sin embargo, como se observa en el Anexo 6 hubo señales a las que el algoritmo no realizó ninguna eliminación debido a que no se encontraron bajas correlaciones entre las señales.

Se concluye que el procesamiento es adecuado para tratar las señales obtenidas por el integrado y por lo tanto fue el procesamiento que se implementó en el dispositivo.

Fase 3. Implementación del dispositivo

De acuerdo a los resultados obtenidos en las fases anteriores se procedió a realizar la implementación del procesamiento en una tarjeta de tamaño reducido.

Adquisición de la señal

La adquisición de la señal se realizó utilizando la tarjeta de evaluación del integrado ADS1299, como se mencionó anteriormente la tarjeta de evaluación cuenta con dos tarjetas conectadas entre sí, por lo tanto desconectaron las tarjetas y toda la adquisición se realizó utilizando la tarjeta que contiene el integrado ADS1299. Lo anterior debido a que la tarjeta madre cuenta con configuraciones predeterminadas y realiza un procesamiento por defecto de las señales; dicha tarjeta se utilizó únicamente para alimentación.

Las dos tarjetas separadas se muestran en la Figura 32. Y en la Figura 33 se muestra la conexión entre ellas, la cual corresponde únicamente a pines de alimentación. La configuración de los electrodos y de los jumpers corresponde a la mostrada en la Figura 13.

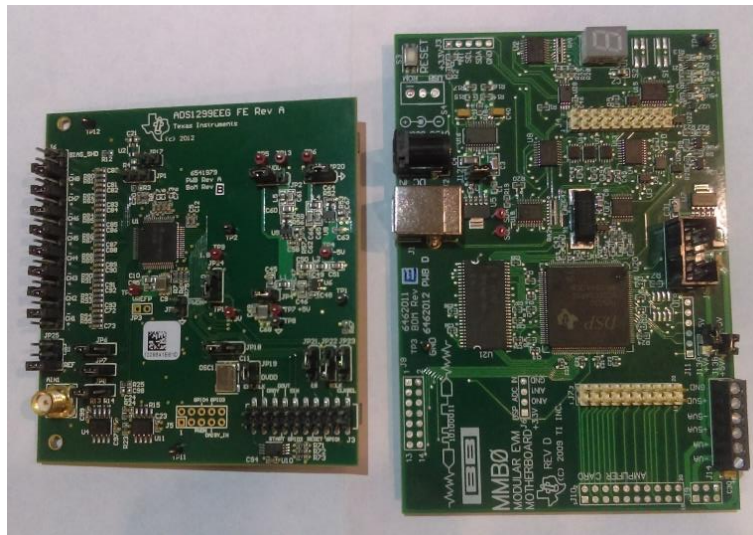


Figura 32. Tarjeta que contiene el integrado ADS1299 (izquierda) y Tarjeta Madre (derecha).

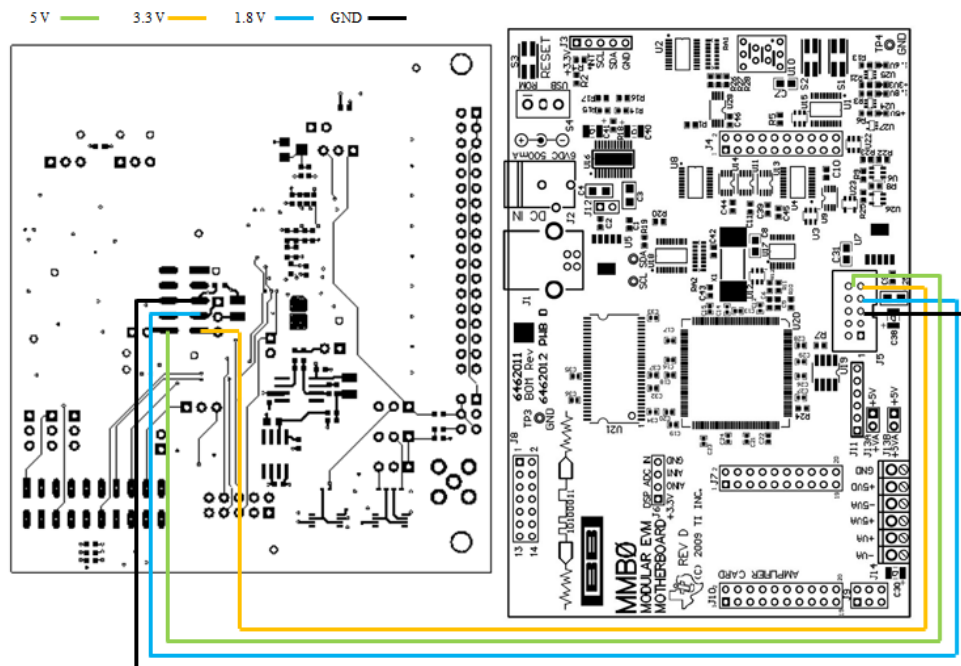


Figura 33. Conexión entre la tarjeta que contiene el integrado y la tarjeta madre.

La tasa de muestreo fue de 250 muestras por segundo y se realizó la adquisición de 6 canales. El integrado ADS1299 utiliza el protocolo *Serial Peripheral Interface* (SPI) para su comunicación con otros dispositivos. Dentro de la comunicación intervienen el reloj del integrado (CLK), el

selector de chip (CS: chip select), el reloj de la comunicación (SCLK), la entrada de datos (DIN) y la salida de datos (DOUT). El integrado actúa como esclavo mientras que el microcontrolador cumple la función de maestro.

La comunicación SPI permite realizar diferentes configuraciones que se relacionan en cuatro modos determinados por la polaridad y la fase. En este caso el integrado funcionó en el modo en el que el reloj permanece estado lógico alto y la información se envía en la transición de alto a bajo.

El periodo del reloj del maestro (CLK) es de mínimo 444 ns y el periodo del reloj del SPI (SCLK) es de 50 ns. La comunicación en el integrado se realiza como se muestra en la Figura 34.

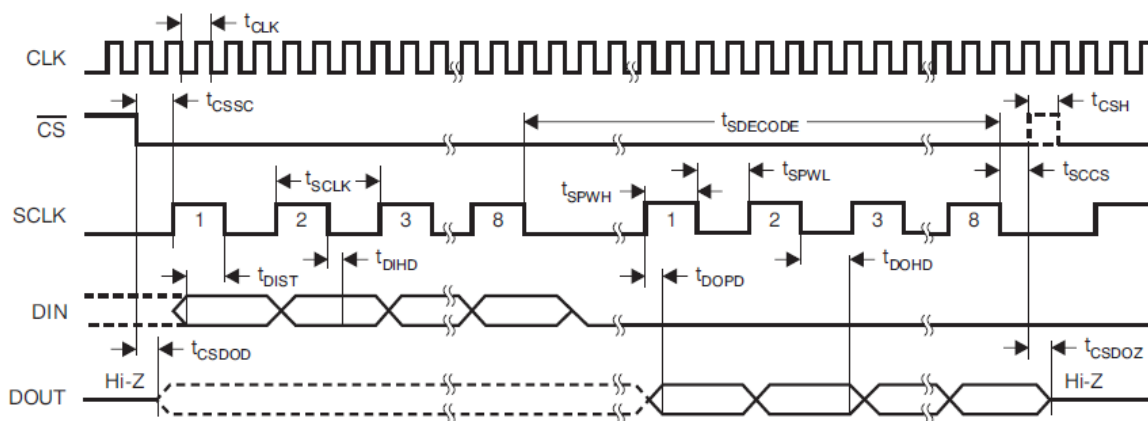


Figura 34. Comunicación SPI del integrado ADS1299.

Procesamiento de la señal

Para el procesamiento se tuvieron en cuenta dos dispositivos, el primero correspondió a un dsPIC33FJ128MC802 y el segundo a un ARM® Cortex® M4 STM32F446. El desempeño de cada uno se puede observar en la Tabla 1 en donde se aprecia que el ARM® cuenta con mejores condiciones para la aplicación requerida.

Tabla 1. Comparación entre los dos microcontroladores

Característica	dsPIC	ARM Cortex M4
Comunicación SPI	SI (2 puertos)	SI (4 puertos)
Oscilador Interno	7,37 MHz	180 MHz
Memoria de programación	128 KB	256 – 512 KB
Tamaño de memoria RAM	16 KB	128 KB

La empresa ARM® provee una plataforma de desarrollo llamada ARM-MBED con el fin de facilitar la implementación del Internet de las Cosas en sistemas embebidos. La plataforma ARM-MBED cuenta con un entorno de desarrollo totalmente en línea que permite realizar la compilación y la programación en la tarjeta de desarrollo. Adicionalmente, en línea se encuentra toda la documentación necesaria para el adecuado desarrollo de cada una de las aplicaciones [102].

El lenguaje de procesamiento para el microcontrolador fue C++. El software MATLAB® cuenta con un complemento que realiza la traducción del código, sin embargo, ésta no fue utilizada para el presente proyecto debido a que la licencia utilizada no cuenta con el mismo. Por lo tanto se procedió a la escritura del código y adicionalmente a la búsqueda de librerías de algebra lineal que ayudaran a la implementación del algoritmo. El código del programa que se utilizó para el procesamiento se encuentra en el Anexo 7. La Figura 35 muestra las conexiones físicas entre el integrado ADS1299 y el microcontrolador en esquemático y la Figura 36 muestra una fotografía de todo el montaje incluyendo la tarjeta de alimentación. A continuación se explican las partes principales del código.

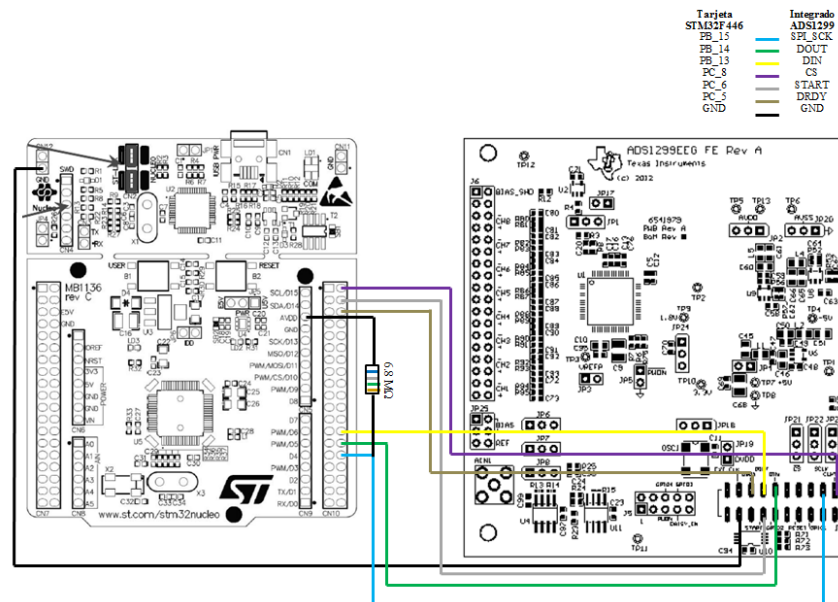


Figura 35. Conexiones entre el microcontrolador y el integrado ADS1299.

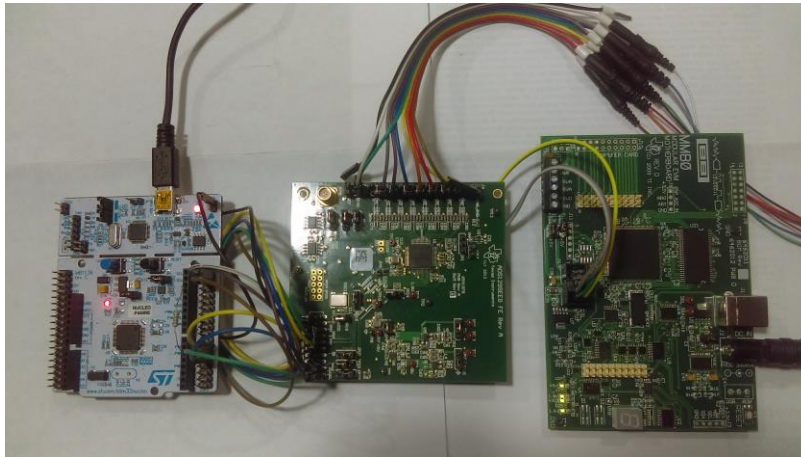


Figura 36. Montaje completo de adquisición con el microcontrolador

Inicialmente, se definieron las librerías a usar, dentro de las que se encuentran la librería “mbed.h” que contiene toda la información para la utilización del compilador y de las funciones brindadas por ARM® MBED, la librería “ads1299.h” que contiene la configuración del integrado y se presenta en el Anexo 8 y la librería “Eigen/Dense.h” que es un conjunto de archivos fuentes de libre acceso cuya base son las librerías Blas y Lapack conocidas por su desarrollo de funciones relacionadas con álgebra lineal [103].

Posteriormente se determinaron las entradas y salidas, con la ayuda de la librería “mbed.h” se configuró el puerto UART únicamente para fines de visualización (etapa que se explicará más adelante), se configuraron además las salidas y entradas correspondientes a la comunicación SPI; SPI_CLK: el reloj de la comunicación, SPI_MISO: entrada de datos de maestro y salida de datos del esclavo, SPI_MOSI: salida de datos del maestro y entrada de datos del esclavo y SPI_CS: selector de chip. Estos se mapearon a los pines de salida y entrada PB_15, PB_14, PB_13 y PC_8 respectivamente.

La salida correspondiente al pin de Inicio (START) fue mapeado a PC_6 y la entrada de Datos Listos (DRDY) fue mapeado a PC_6 y corresponde a una interrupción. La conexión de estos pines con el integrado ADS1299 se muestra en la Figura 35 en donde se adicionó una resistencia de $6.8\text{M}\Omega$ en la salida SPI_CLK. Adicionalmente, se generó una salida para un LED de verificación de funcionamiento.

Existe dentro del código un apartado para pruebas de funcionamiento, lo anterior se realizó a través de la inicialización de una matriz predeterminada de la cual se conocían todos los resultados.

Luego, se generaron cada uno de los 7 prototipos utilizados en el procesamiento:

- Enviar comando (ads_send_command): En este prototipo se utilizó la comunicación SPI para enviar comandos al integrado ADS1299.
- Escribir en los registros (ads_wreg): Además de utilizar la comunicación SPI se requirió de seguir un protocolo para escribir los datos en los diferentes registros. Para realizar la escritura en los registros se puede remitir al datasheet del integrado.
- Leer los registros (ads_rreg): Se utilizó la comunicación SPI y el protocolo descrito en el datasheet para leer los registros.
- Datos listos (data_ready): En este protocolo se adquirió la señal en el momento en el que la interrupción de DRDY. Se generó la matriz de datos de 6 canales, se acondicionó la señal y se realizó un submuestreo. La configuración de muestreo fue de 250 muestras por segundo, sin embargo, en el momento de realizar el procesamiento en el microcontrolador, éste envió un mensaje en el que indicaba que se encontraba fuera de memoria. Por lo tanto se procedió a implementar un submuestreo en el que se procesaran únicamente 125 muestras debido al desempeño del microcontrolador. El código se puede consultar en el Anexo 7.
- Configuración del integrado (config_ads): Se realizó una verificación del integrado conectado y se configuraron los registros según lo descrito en la fase 2.
- Método BSS-CCA (bss_cca): Se realizó la escritura del código del algoritmo BSS-CCA en lenguaje C++. Dentro del cual se encontró la eliminación de artefactos oculares según umbral. En este protocolo ingresó la matriz de datos y el resultado fue la señal reconstruida. Este protocolo se explicará más adelante.
- Filtro 0.5 – 35 Hz (filtro): En este protocolo se implementó el filtro pasa-bandas de 0,5 a 35 Hz.

- Posteriormente se declararon las variables globales dentro de las que se encuentran las matrices de datos de entrada, señal reconstruida y señal filtrada.

Finalmente se implementó el programa principal (main) en el cual se configuró la comunicación SPI, el integrado ADS1299 y se esperó a la interrupción de DRDY para luego implementar el método BSS-CCA y filtrar la señal.

Uno de los puntos más importantes en el presente trabajo es la implementación del código de procesamiento en la tarjeta reducida. Por tal motivo se explica detalladamente la implementación del método BSS-CCA que es el que elimina artefactos oculares en el microcontrolador.

Inicialmente, se realizó el cambio de formato de las matrices, la matriz en la cual se almacenó la información recibida del integrado ADS1299 se convirtió en una matriz aceptada en formato “Eigen” para su utilización con la librería del mismo nombre. Posteriormente, se procedió a la creación de las dos matrices (Y y X_1) con corrimiento para el desarrollo del método BSS-CCA.

```

/*****
/* Rutina Para Efectuar el Método BSS - CCA
/*****

void bss_cca(void){
    volatile int i,j,d,v;
    d=1;

    // Matriz Data
    MatrixXd data(M,N);
    for (i = 0; i < M; i = i + 1)
    {
        for (j = 0; j < N; j = j + 1)
        {
            data(i,j)=dato[j][i];
            pc.printf("data[%d][%d]: %e\r\n",i,j,data(i,j));
        }
    }

    // Creación de La Y
    MatrixXd Y(M,N-1);
    for (i = 0; i < M; i = i + 1)
    {
        for (j = 0; j < N-1; j = j + 1)
        {
            Y(i,j)=data(i,j+d);
        }
    }
}

```

```

    }
}

// Creación de La X1
MatrixXd X1(M,N-1);
for (i = 0; i < M; i = i + 1)
{
    for (j = 0; j < N-1; j = j + 1)
    {
        X1(i,j)=data(i,(j+1)-d);
    }
}

// Creación de Cyy = T*Y*Y'
MatrixXd Cyy(M,M);
Cyy=((1/(N*1.0))*Y*(Y.transpose()));

// Creación de Cxx = T*X1*X1'
MatrixXd Cxx(M,M);
Cxx=(1/(N*1.0))*(X1*(X1.transpose()));

// Creación de Cxy = T*X1*Y'
MatrixXd Cxy(M,M);
Cxy=(1/(N*1.0))*(X1*(Y.transpose()));

// Creación de Cyx = Cxy'
MatrixXd Cyx(M,M);
Cyx=(Cxy.transpose());

```

Luego, se realizó la creación de las matrices de correlaciones entre las matrices anteriormente creadas, para tal fin se realizaron operaciones matemáticas con la ayuda de la librería “Eigen”. De esta operación se obtienen las matrices cuadradas de correlación C_{yy} , C_{xx} , C_{xy} y C_{yx} . Después, se crean las matrices inversas de las matrices de correlaciones C_{yy} y C_{xx} . Se creó también una matriz previa a la matriz de desmezcla que realiza la multiplicación entre la matriz inversa de C_{xx} , la matriz C_{xy} y la matriz inversa de C_{yy} .

```

// Creación de ICyy = inv(Cyy)
MatrixXd ICyy(M,M);
ICyy=Cyy.inverse();

// Creación de ICxx = inv(Cxx)
MatrixXd ICxx(M,M);
ICxx=Cxx.inverse();

// Matriz antes de la W y la r
MatrixXd PreW(M,M);
PreW=ICxx*Cxy*ICyy

```


Posteriormente, se procedió a encontrar tanto los valores como los vectores propios para la matriz previa a la matriz de desmezcla. Y luego se realizó la organización de coeficientes de correlación de mayor a menor. Finalmente, la matriz de desmezcla W se formó organizando los elementos de la matriz previa a la matriz de desmezcla de acuerdo con la organización de coeficientes de correlación.

```
// Obtener Eigen valores y vectores = W, r
EigenSolver<MatrixXd> es(PreW);
MatrixXcd r = es.eigenvalues();
MatrixXcd W = (es.eigenvectors()).transpose();

ArrayXd rr=r.real();
ArrayXd ra=rr.abs();
ArrayXd rs=ra.sqrt();

// [r,I] = sort('descend');
MatrixXd rf(M,M);
for (i = 0; i < M; i++)
{
    v=0;
```

```
    for (j = 0; j < M; j++)
    {
        if (rs(v)>rs(j) ) {
            // No Opere
        }
        else // Si el numero no es el mayor
        {
            v=j;
        }
    }
    rf(0,i)=rs(v);
    rf(1,i)=v;
    rs(v)=0;
}

for (i=0; i<M;i++){
    pc.printf("rf[%d][%d]: %e\r\n",i,rf(0,i));
}

// W = W(:,I)';
MatrixXd Wo(M,M);
for (i=0; i<M; i++)
{
    for (j=0;j<M;j++)
    {
        Wo(i,j)=W((rf(1,i)),j);
    }
}
}
```

```

// La señal estimada Y
MatrixXcd Yf(M,N);
Yf=Wo*data;

// Eliminar componentes

for (i=0; i < M; i++)
{
    if (rf(0,i) <= 0.75) {
        for (j=0 ; j < N; j++){
            Yf(i,j)=0;
        }
    }
}

// Reconstrucción de La señal
Xf= Wo.inverse()*Yf;
for (i = 0; i < M; i = i + 1)
{
    for (j = 0; j < N; j = j + 1)
    {
        pc.printf("Xf[%d][%d]: %e\r\n",i,j,Xf(i,j));
    }
}
}

```

Finalmente se realizó el cálculo de las componentes almacenadas en la matriz Y y se procedió a eliminar los artefactos oculares con la ayuda del umbral de correlación establecido en 0.75. Como procesamiento final se realizó la reconstrucción de la señal a través de la matriz de desmezcla y se la llamó X_f .

Se realizó la medición del tiempo de adquisición de las 250 muestras, así como el tiempo de procesamiento. El tiempo de adquisición fue de $512.415 \mu s$ y el de procesamiento fue de $96.703 \mu s$.

Visualización de la señal

El microcontrolador que cuenta con comunicación UART permitió verificar la adquisición de los datos. A través de la creación de un puerto COM en el computador se realizó la comunicación entre el mismo y el microcontrolador. Para la visualización y manipulación de los datos se utilizó el software Hercules®.

Por medio del software se capturaron los datos y se guardaron en un archivo de texto plano para posteriormente llevarlos al entorno Matlab® y hacer la gráfica de los mismos. Los resultados

arrojados por el procesamiento fueron: la señal original adquirida, el vector de correlaciones, la señal reconstruida y la señal filtrada.

Placa PCB

Después de realizar el diseño y la implementación del procesamiento en el microcontrolador y realizar las pruebas de funcionamiento se procedió a diseñar el dispositivo final en una placa PCB.

Las partes del diseño fueron las siguientes: Circuito de batería, circuitos de monitoreo de voltajes, circuito del integrado ADS1299, circuito del microcontrolador, circuitos de comunicación con el computador y circuitos de conexión de voltajes. El esquemático del dispositivo se puede observar en el Anexo 11.

Circuito de Batería

Uno de los requisitos de mayor importancia en el planteamiento del presente proyecto correspondió a que el sistema fuera alimentado a través de baterías. Lo anterior debido a que de esta manera se cumplía con la portabilidad del dispositivo y adicionalmente se eliminaba el riesgo de tener componentes de frecuencia de la red eléctrica dentro de las señales adquiridas. Después de realizar el análisis de voltaje de todos los integrados se encontró que se necesitaba alimentación de 5V y 3.3V y que por lo tanto, se requería una batería de al menos 12V para suplir las necesidades. Adicionalmente, se propuso un circuito de carga de baterías. En la Figura 37 se observa la conexión de las baterías y el circuito de carga de baterías.

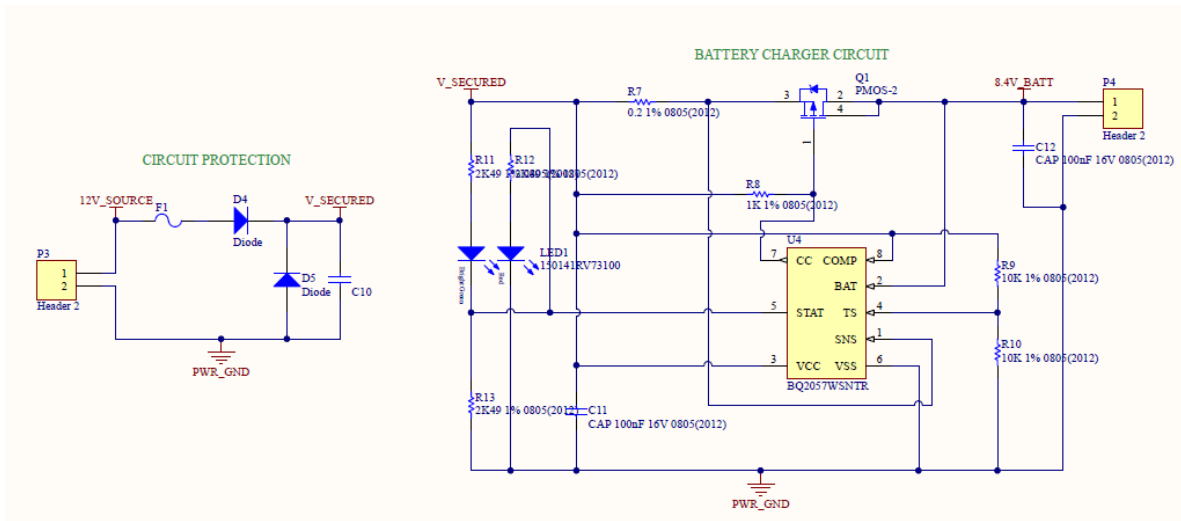


Figura 37. Circuito de batería y carga de baterías.

Circuito de monitoreo de voltajes

Para garantizar que no se presente una sobrecarga en el sistema, se diseñaron circuitos de protección y de pruebas para monitorear el voltaje en cada uno de los puntos necesarios. Se muestra en la Figura 38 la fuente de 5V y la regulación de 3,3V. Adicionalmente en la Figura 39 se muestran las protecciones de voltajes y las uniones de las diferentes tierras dentro del circuito.

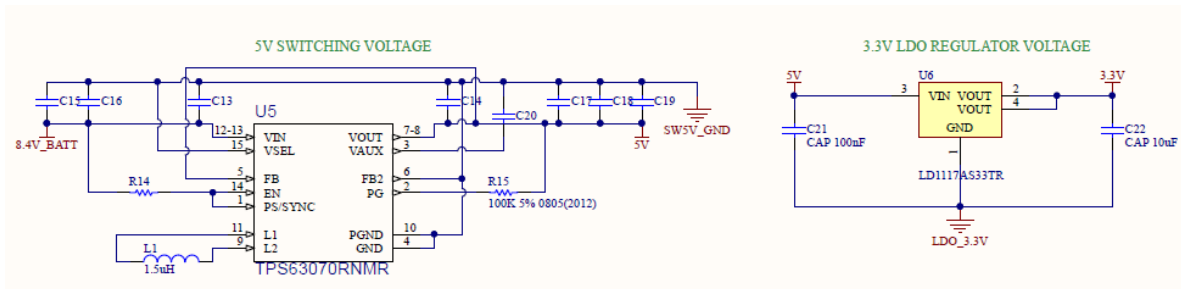


Figura 38. Fuente de 5V y de 3.3V regulados.

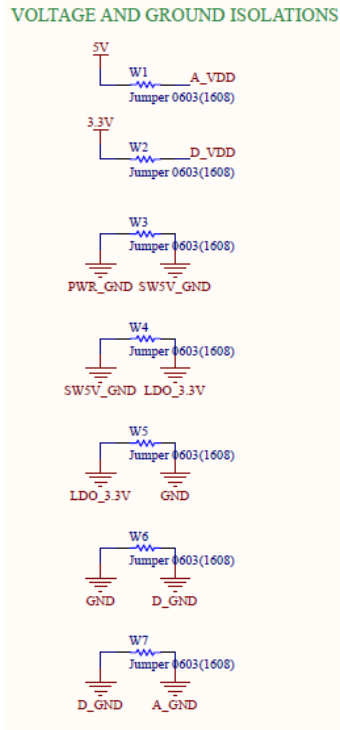


Figura 39. Protección de voltajes y unión de tierras del dispositivo

Circuito del integrado ADS1299

Se realizó la configuración completa del integrado para su adecuado funcionamiento, adicionalmente se dispusieron las entradas de los 8 canales de señales y del canal de referencia. Cabe resaltar que una de esas entradas puede ser utilizada como electrodo de BIAS. Adicionalmente a las configuraciones requeridas se realizó la conexión entre el integrado y el microcontrolador. En la Figura 40 se muestran dichas conexiones.

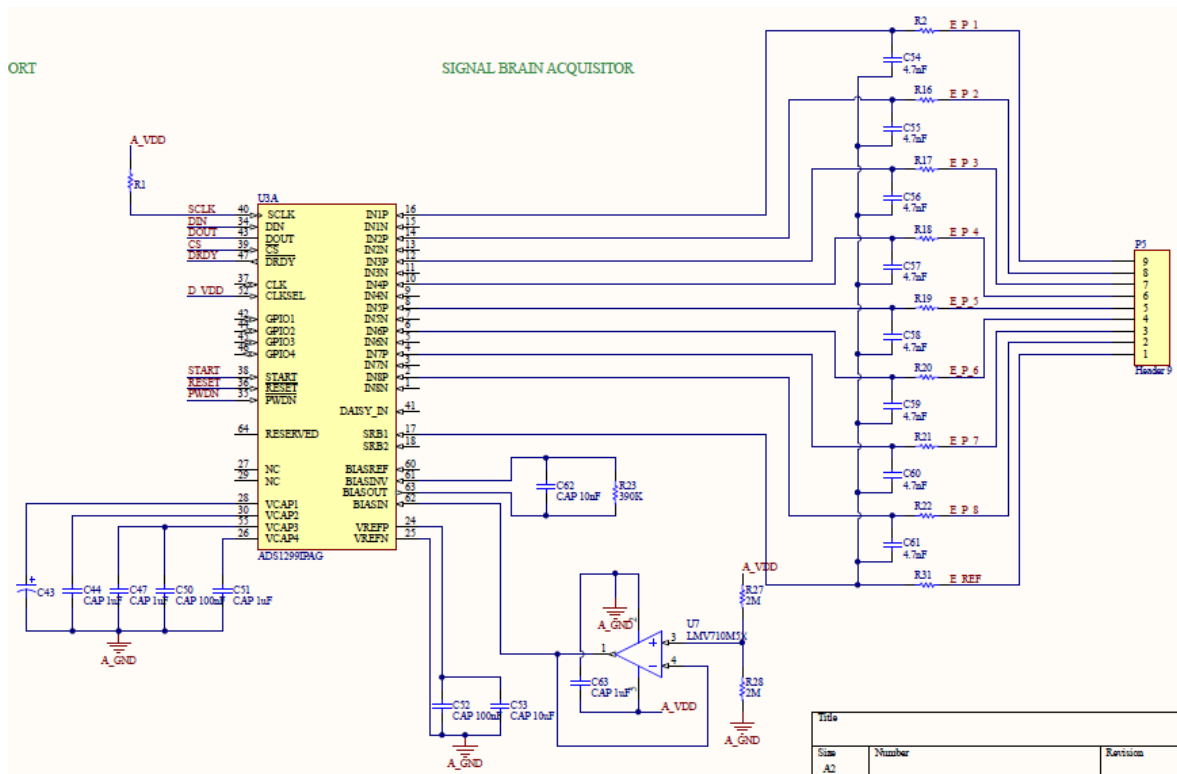


Figura 40. Circuito del integrado ADS1299.

Circuito del microcontrolador

El microcontrolador utilizado en la implementación del dispositivo tuvo inconvenientes de memoria al realizar el procesamiento de más de 125 muestras por segundo, por lo que se decidió realizar el diseño final con un microcontrolador con mejor desempeño. Por lo anterior, se eligió el microcontrolador STM32F747 que cuenta con una frecuencia de operación de 256 MHz, una memoria Flash de 1MB y una RAM de 512 KB. Las conexiones del microcontrolador con el integrado ADS1299 y las salidas del mismo se muestran en la Figura 41. Adicionalmente, se propuso un circuito de programación del microcontrolador para realizar de manera sencilla la operación.

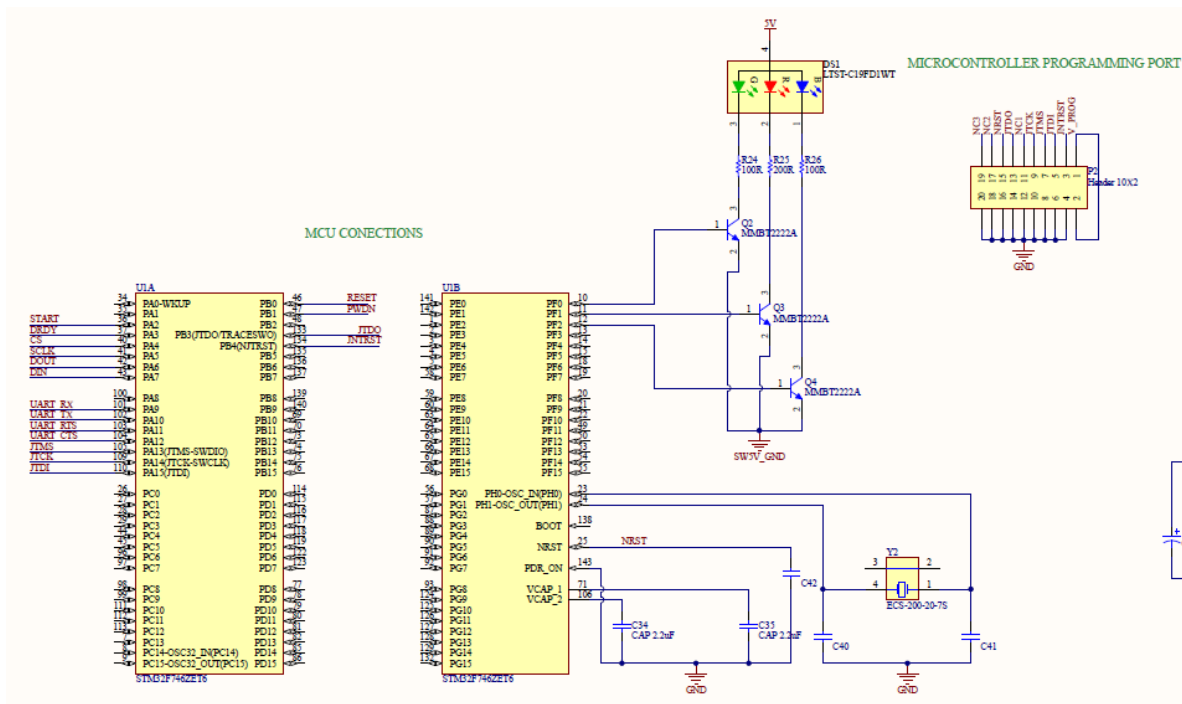


Figura 41. Circuito del microcontrolador y su programador.

Los circuitos de conexión del microcontrolador con el computador a través de puerto UART y los circuitos de conexión de voltajes tanto del integrado ADS1299 como del microcontrolador se muestran en el Anexo 11 con detalle.

La lista de componentes utilizados en el diseño se puede ver en el Anexo 12, así mismo se presenta un estimado del presupuesto requerido para el diseño con cambio a la fecha y sin tener en cuenta el costo del envío. El costo total es de aproximadamente \$ 657.000 pesos.

Finalmente, en la Figura 42 se muestra la placa PCB con todos sus componentes integrados y en la Figura 43 se muestra la parte superior de la tarjeta en modelo 3D. Las vistas del modelo 3D se pueden observar en el Anexo 13.

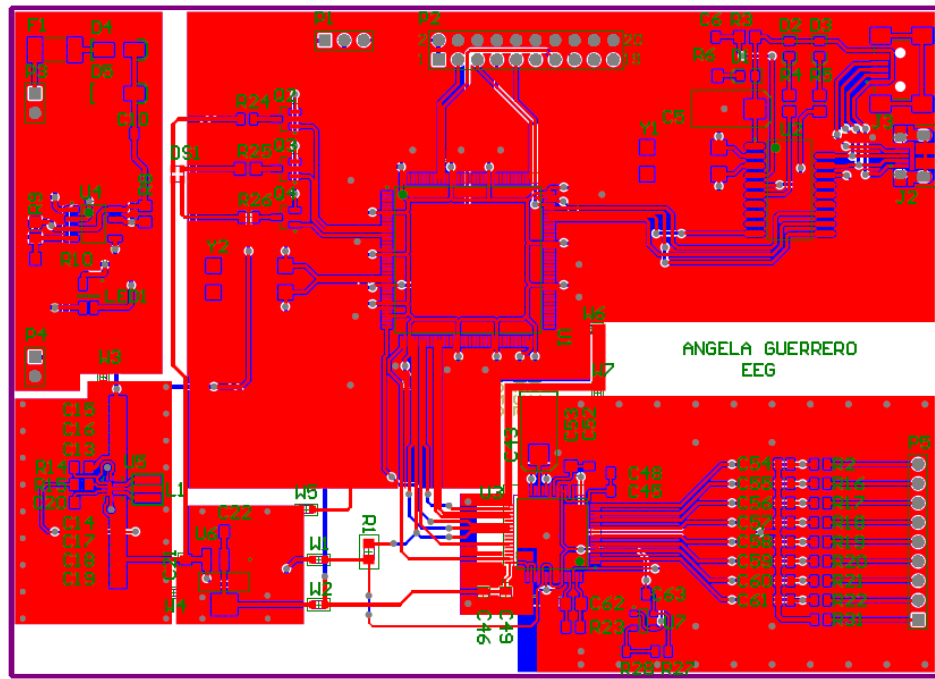


Figura 42. Placa PCB con sus componentes integrados.

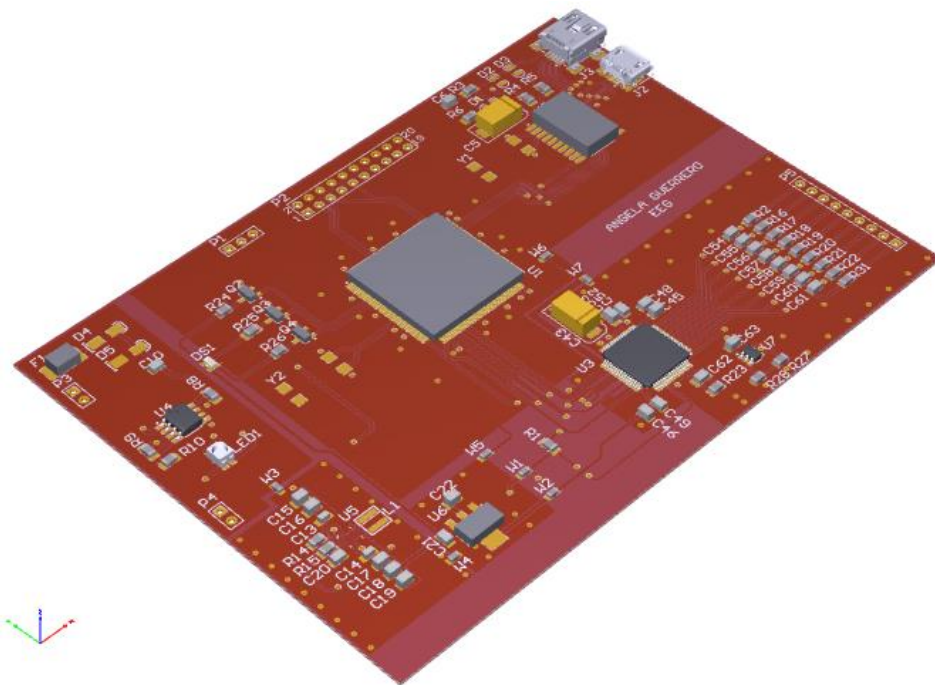


Figura 43. Tarjeta diseñada en modelo 3D.

Fase 4: Pruebas de funcionamiento

Para realizar las pruebas de funcionamiento del dispositivo se realizó un montaje en el cual se pudo adquirir una señal de EEG con el dispositivo diseñado y fue procesada mediante el microprocesador, al mismo tiempo la matriz de los datos originales se guardan y se procesa posteriormente en MATLAB®, finalmente se compararon los resultados.

Se realizaron 5 tomas en diferentes individuos de manera similar a la fase 3, se tomaron señales en reposo y en movimiento de miembro inferior. Cada uno de los resultados se guardaron en archivos de texto plano en donde se registraron la matriz *dato* la cual corresponde a los datos adquiridos originales, *rf* que corresponden a los valores de correlación, *Xf* que es la matriz reconstruida y *sfinal* la cual corresponde a la señal reconstruida filtrada. Posteriormente, se implementó el código de MATLAB® que se encuentra en el Anexo 9 para realizar las gráficas correspondientes.

A continuación se muestra el resultado de uno de los procesamientos en donde se muestran las imágenes de la señal original en la Figura 44, la señal reconstruida por el procesamiento en MATLAB® en la Figura 45 y la señal reconstruida por el microcontrolador en la Figura 46. Los resultados de las demás pruebas pueden observarse en el Anexo 10 incluyendo la tabla en donde se observan los valores de la correlación de cada una de las pruebas.

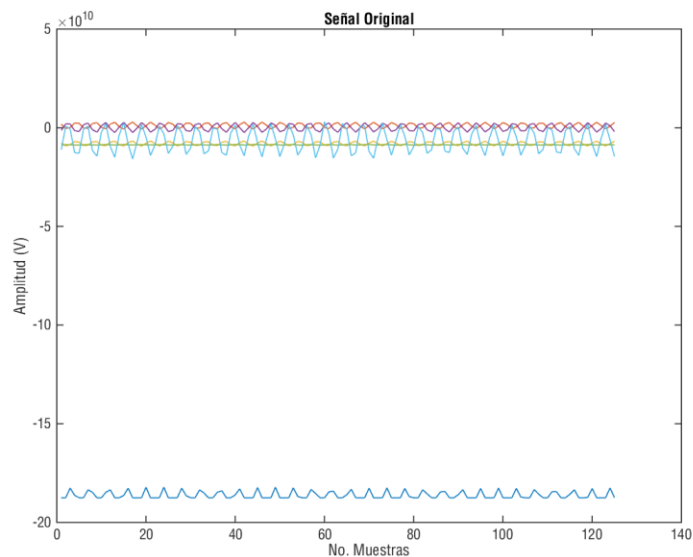


Figura 44. Señal original tomada por el dispositivo diseñado

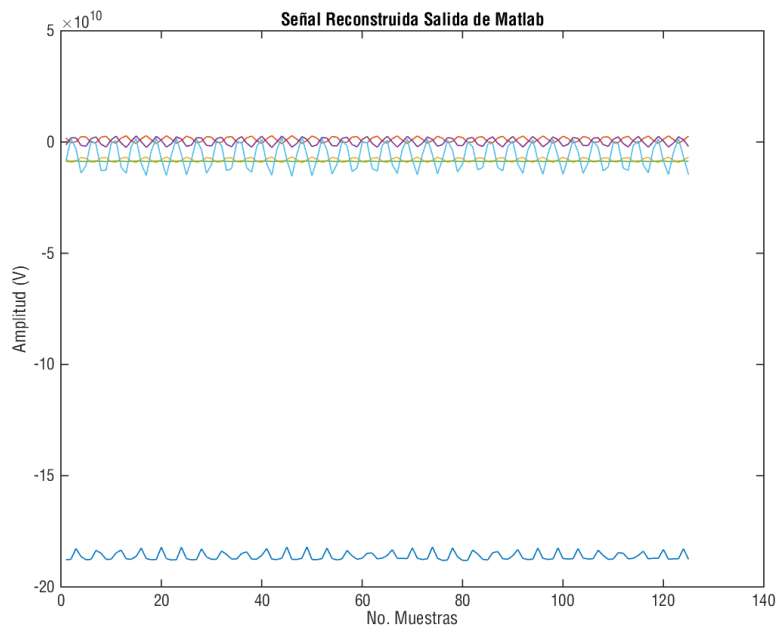


Figura 45. Señal reconstruida después del procesamiento en Matlab®

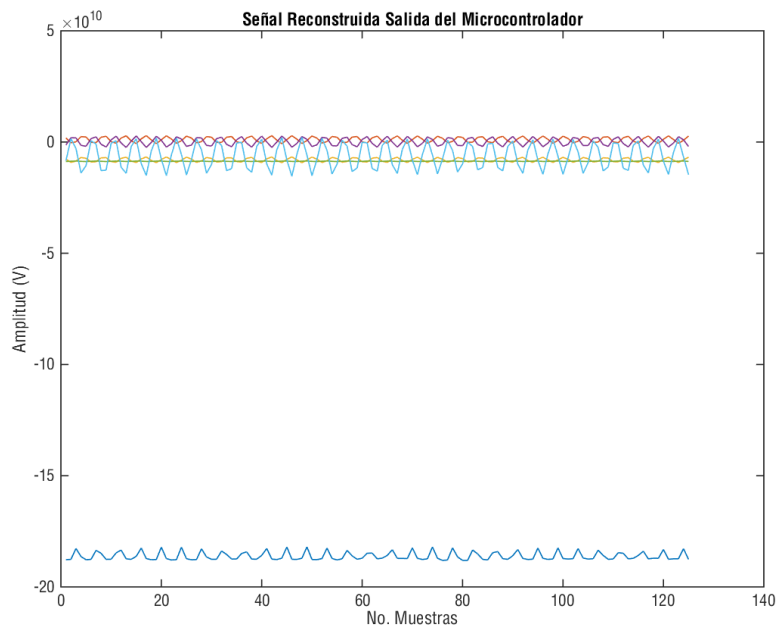


Figura 46. Señal reconstruida por el procesamiento en el microcontrolador

En las figuras de los resultados obtenidos en la fase de pruebas se observó que el procesamiento realizado por el microcontrolador fue exactamente el mismo realizado en MATLAB®. Sin embargo, para verificar el adecuado funcionamiento del procesamiento en el microcontrolador se procedió a tomar una de las señales adquiridas por el integrado ADS1299 en la fase 2, dividirla en segmentos de 250, efectuar el procesamiento en el microcontrolador y guardar los resultados para unirlos y verificar lo arrojado por el microcontrolador.

Para el ejemplo, se tomó la señal del individuo 5 en movimiento y se realizó una segmentación inicial a partir de la muestra número 5000 hasta la muestra 6250. Posteriormente, se segmentó la señal en partes de 250 muestras, se copió la matriz y se le programó al microcontrolador. Los resultados se fueron guardando en un archivo de Excel para leerlos posteriormente en Matlab® y graficar una comparación entre la señal original y la procesada. Para una mejor visualización se graficaron los seis canales por separado, éstas gráficas se pueden observar a partir de la Figura 47 hasta la Figura 52.

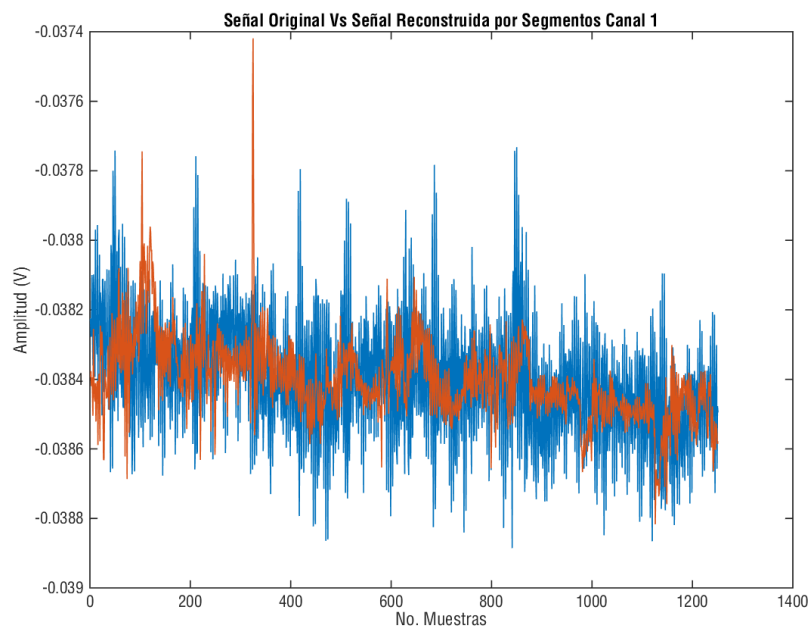


Figura 47. Señal Original Vs Señal Reconstruida por Segmentos Canal 1

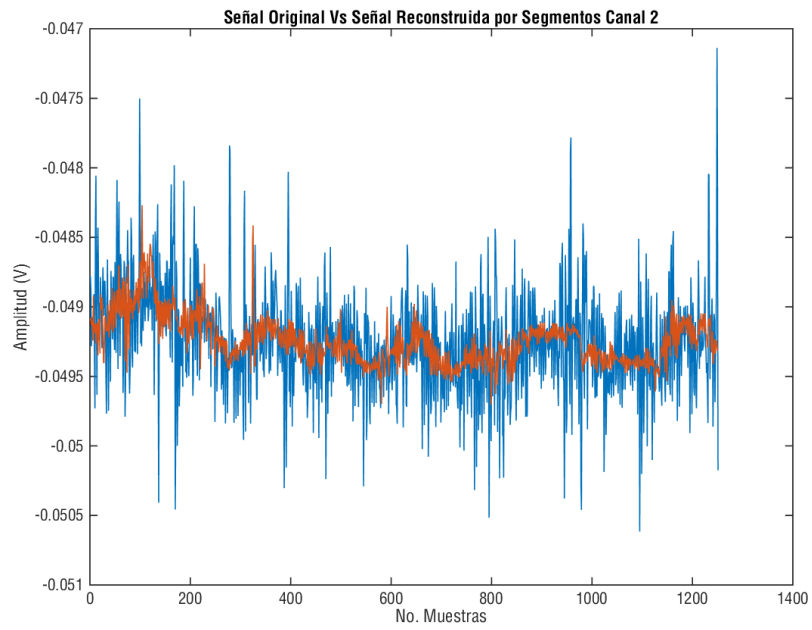


Figura 48. Señal Original Vs Señal Reconstruida por Segmentos Canal 2

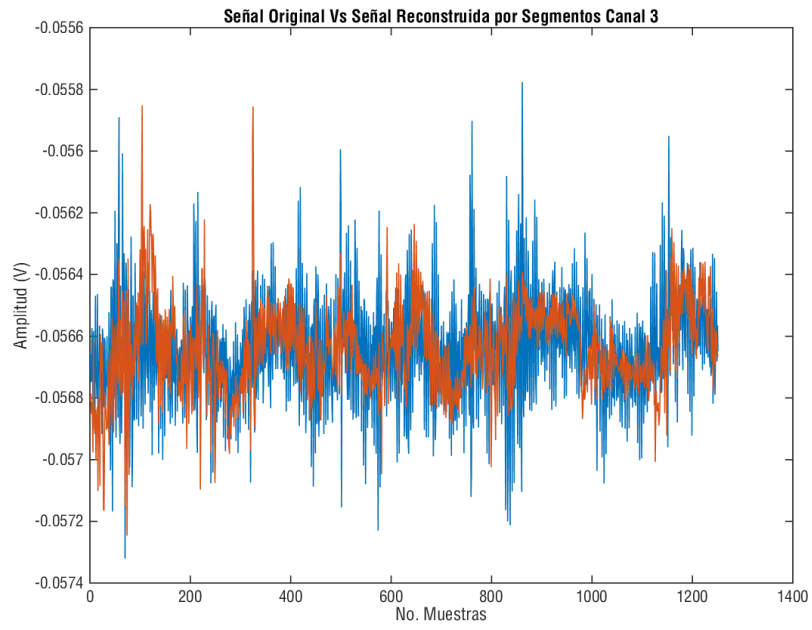


Figura 49. Señal Original Vs Señal Reconstruida por Segmentos Canal 3

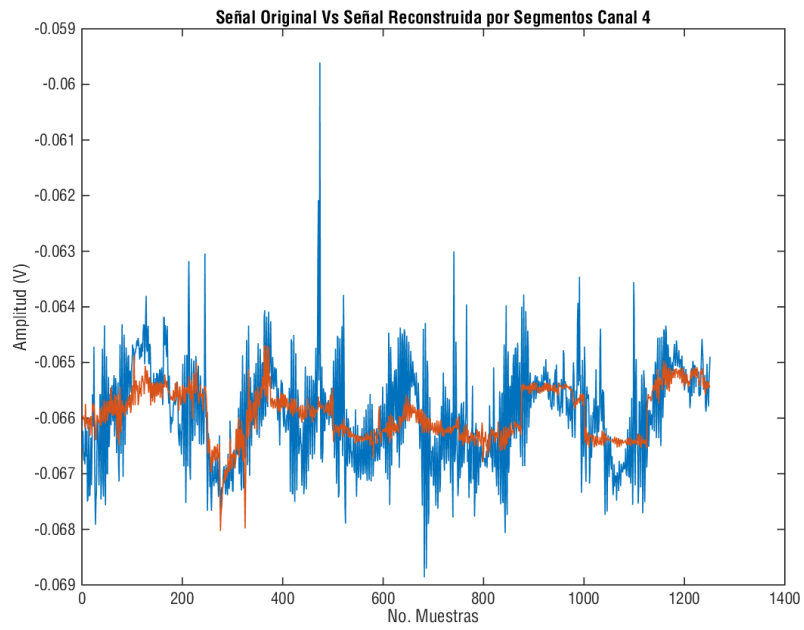


Figura 50. Señal Original Vs Señal Reconstruida por Segmentos Canal 4

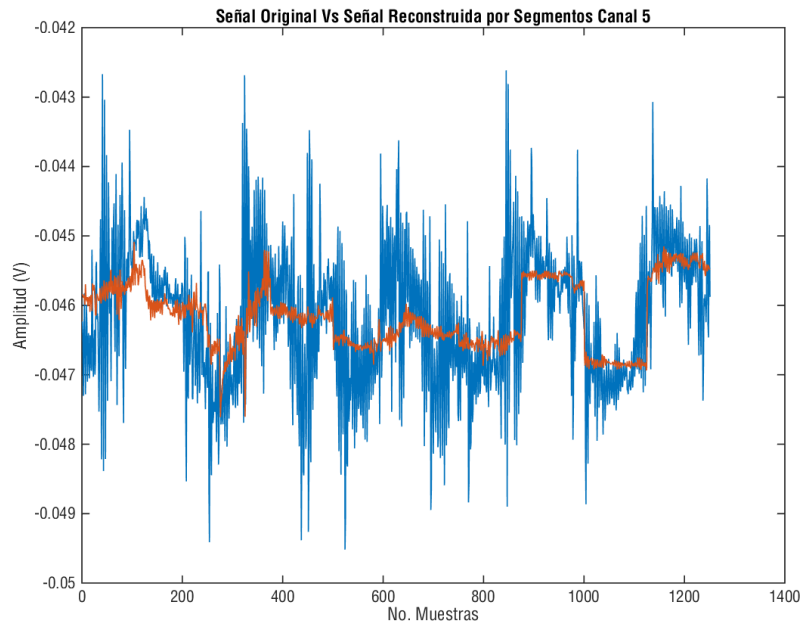


Figura 51. Señal Original Vs Señal Reconstruida por Segmentos Canal 5

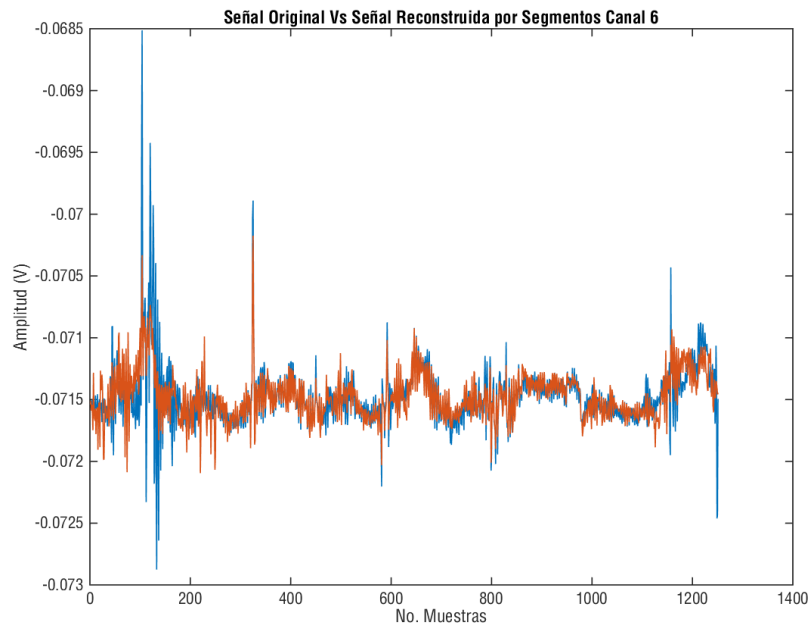


Figura 52. Señal Original Vs Señal Reconstruida por Segmentos Canal 6

Como se puede observar en las figuras, el método BSS-CCA arrojó excelentes resultados inclusive con un número de muestras reducidas; adicionalmente el procesamiento fue realizado correctamente por segmentos lo cual permite concluir que la implementación del método en el microcontrolador fue realizada adecuadamente.

Contribución

La implementación de un dispositivo que realice la adquisición y el procesamiento de señales cerebrales de miembro inferior tiene un gran impacto en diferentes áreas. Como primera medida se encuentra el desarrollo de equipos portátiles que puedan ser parte de prótesis robóticas inteligentes debido a que éstas requieren de componentes electrónicos funcionales y pequeños para la óptima ejecución de sus funciones. La importancia de realizar el procesamiento de las señales en pequeños procesadores es la de migrar de sistemas con una tasa muy alta de procesamiento, como lo son los procesadores de computadores, a microcontroladores.

De la misma manera la implementación de algoritmos que sean altamente efectivos en el procesamiento de señales cerebrales es mucho más fácil en computadores con lenguajes de alto

nivel y software elaborado computacionalmente hablando como MATLAB®. Sin embargo, se hace necesario desarrollar, implementar y probar algoritmos lo suficientemente eficientes para que puedan ser implementados en procesadores pequeños. Con la adecuada implementación de estos procesamientos se puede pensar en un futuro en donde los sistemas BCI no estén conectados a un computador sino que sean portátiles e igualmente efectivos.

Adicionalmente, teniendo en cuenta que las ondas cerebrales de miembros superiores son más fáciles de captar, se ve la necesidad de investigar en el campo de los miembros inferiores ya que su impacto puede ser considerable si se tiene en cuenta que la recuperación de la movilidad de miembros inferiores conlleva a un avance en la locomoción de las personas discapacitadas.

CONCLUSIONES Y RECOMENDACIONES

El diseño de un dispositivo que adquiere señales cerebrales de miembro inferior y las procesa para eliminar artefactos oculares en una placa de tamaño reducida fue desarrollado en el presente trabajo.

Se implementó el procesamiento de eliminación de artefactos a través del método BSS – CCA el cual descompone la señal y arroja un vector de correlaciones que permite tomar decisiones acerca de si las componentes corresponden a artefactos o a información importante de la señal cerebral. Adicionalmente, se implementó un filtro pasa-banda para filtrar las frecuencias útiles en el análisis de señales de EEG.

A pesar de que existen diferentes métodos para descomponer la señal y eliminar los artefactos oculares, tales como ICA o métodos adaptativos; la utilización del método CCA ayudó en la fácil implementación en el microcontrolador. Se concluye que el método CCA es un método útil para eliminar artefactos y ruidos de las señales a partir de análisis estadístico que permite determinar un umbral de correlación; sin embargo, debido a su componente netamente estadístico, el método no discrimina la fuente de los artefactos. En el presente trabajo se asumió que los artefactos eliminados correspondían a artefactos oculares debido a la posición de los electrodos, sin embargo, se requeriría de un análisis comparativo con métodos auto-correlacionados o adaptativos con el uso de electrodos de EOG para precisar el umbral de discriminación o las modificaciones necesarias en la configuración del método CCA para corroborar la eliminación total de artefactos oculares. Adicionalmente, es indispensable mencionar que existen diferentes fuentes de artefactos que provienen de otras fuentes y que pueden estar afectando la toma de datos. Además, para realizar una implementación en un futuro se requeriría el entrenamiento por parte del usuario con el fin de realizar un adecuado procesamiento de la señal dependiendo del usuario.

La adquisición de la señal se realizó a través del integrado ADS1299 el cual fue probado gracias a la utilización de la tarjeta de evaluación. Con esta herramienta se pudo realizar el procesamiento y determinar si era adecuado para las señales adquiridas con dicho integrado. Ya teniendo el procesamiento completo con óptimo funcionamiento y resultados favorables de las señales adquiridas con el integrado se procedió a implementarlo en un microcontrolador.

La tarjeta de evaluación incluye un software para verificar el funcionamiento del integrado a través de las diferentes configuraciones posibles con el que este cuenta. No obstante, tanto la

comunicación de la tarjeta como la configuración de la misma tuvieron dificultades. La tarjeta de evaluación únicamente se comunica con el sistema operativo Windows 7 y un puerto USB de baja velocidad. Por otro lado, en la configuración de la tarjeta expuesta en el datasheet no se explican ciertos criterios que se toman por conocidos y que interfieren en la toma de las señales. Al realizar las pruebas de funcionamiento que vienen integradas en la tarjeta de evaluación, todas arrojaron los resultados esperados. Pero al efectuar tomas de señales reales los resultados tuvieron mayor cantidad de ruido de lo esperado.

El microcontrolador STM32F446 fue el elegido para realizar tal tarea por su alta tasa de procesamiento y su cantidad de memoria. Además de contar con el protocolo de comunicación SPI que corresponde al protocolo utilizado por el integrado ADS1299. El código correspondiente al procesamiento de las señales, la comunicación SPI y la comunicación con un computador para fines de visualización fue desarrollado en su totalidad en lenguaje C++ y programado en el microcontrolador a través del compilador en línea MBED.

La implementación del código desarrollado en MATLAB® en lenguaje C++ tuvo algunas dificultades en cuanto a la inclusión de funciones de manejo matricial. Sin embargo, con la librería Eigen y el desarrollo del código del integrado se logró una implementación ordenada. Hubo no obstante dos inconvenientes importantes en la implementación del procesamiento en el microcontrolador. El primero correspondió a la comunicación SPI, puesto que la configuración detallada en el datasheet del integrado fue confusa y tomó un tiempo considerable ejecutar la comunicación. El segundo inconveniente surgió en el momento de la prueba del procesamiento en cuanto a capacidad de memoria del microcontrolador. A medida que se iba generando el código y se iba probando, los resultados eran favorables. Pero en el momento de incluir la librería Eigen y de formar las matrices como lo requería la librería la capacidad de memoria del microcontrolador fue disminuyendo. Dicha capacidad disminuyó tanto que en el momento de realizar la prueba con 250 muestras (correspondientes a un segundo) el microcontrolador se quedó sin memoria. Razón por la cual la implementación de un sub-muestreo fue absolutamente necesaria. Lo anterior, repercutió en el cambio de modelo del microcontrolador por uno de la misma familia pero con mejor desempeño para el diseño final del dispositivo.

La fase de pruebas arrojó resultados favorables en la implementación del procesamiento en lo concerniente a que los resultados del microcontrolador corresponden exactamente a los resultados del software MATLAB®. Los resultados de la prueba realizada por segmentos mostró un

procesamiento correcto sin importar la cantidad de muestras, sin embargo, es indispensable que se realice un procesamiento de mayor cantidad de muestras debido a que se podría llegar a mejorar el tiempo de procesamiento. Por lo anterior en el diseño propuesto, el microcontrolador corresponde a uno con mayor capacidad de memoria.

Finalmente se logró diseñar el dispositivo con componentes electrónicos de superficie alimentado por baterías el cual funciona únicamente para una configuración del integrado ADS1299. Sería bueno realizar modificaciones al diseño para que pueda ser configurado con entradas diferenciales y a su vez entradas independientes, así como también lograr la manera de que sea modular para incluir más canales de adquisición.

El diseño presentado en este trabajo contribuye a la implementación de algoritmos de procesamiento de señales que eliminen artefactos en un microcontrolador. No obstante, es una pequeña parte del desarrollo que se debe llegar a hacer para tener una prótesis robótica de miembro inferior, la salida del procesamiento del dispositivo diseñado corresponde al punto de partida para realizar la extracción de características y clasificación de movimientos. Adicionalmente, investigaciones en protocolos de comunicaciones inalámbricas y más rápidas deben ser desarrolladas, así como también estudios en la localización de electrodos de manera más sencilla y cómoda para el individuo.

REFERENCIAS

- [1] O. M. d. I. Salud, «Organización Mundial de la Salud,» 2011. [En línea]. Available: http://www.who.int/disabilities/world_report/2011/summary_es.pdf?ua=1. [Último acceso: Abril 2015].
- [2] Organización Mundial de la Salud, 2015. [En línea]. Available: <http://www.who.int/topics/disabilities/es/>.
- [3] Naciones Unidas, Mayo 2015. [En línea]. Available: <http://www.minsalud.gov.co/Documentos%20y%20Publicaciones/Convenci%C3%B3n%20sobre%20los%20Derechos%20de%20las%20Personas%20con%20Discapacidad%20Protocolo%20Facultativo.pdf>.
- [4] Organización Mundial de la Salud, 2015. [En línea]. Available: <http://www1.paho.org/Spanish/DD/PUB/Discapacidad-SPA.pdf?ua=1>.
- [5] Departamento Administrativo Nacional de Estadística, Mayo 2015. [En línea]. Available: http://www.dane.gov.co/files/investigaciones/discapacidad/Total_nacional.xls.
- [6] Dirección Contra Minas, Mayo 2015. [En línea]. Available: <http://www.accioncontraminas.gov.co/estadisticas/Paginas/victimas-minas-antipersonal.aspx>.
- [7] Ministerio de Salud y Protección Social, Mayo 2015. [En línea]. Available: <http://www.minsalud.gov.co/proteccionsocial/Paginas/DisCAPACIDAD.aspx>.
- [8] L. Puglisi y H. Moreno, «Prótesis Robóticas».
- [9] P. Serra Añó, «Módulo Ortoprotésica».
- [10] J. M. Dorador González, P. Ríos Murillo, I. Flores Luna y A. Juárez Mendoza, «Robótica y Prótesis Inteligentes,» *Revista Digital Universitaria*, vol. 6, nº 1, Enero 2004.
- [11] B. Lee, C. Liu y M. Apuzzo, «A primer on Brain-Machines Interfaces, Concepts, and

Technology: A key element in the future of functional neurorestoration.,» 2012.

- [12] J. Wolpaw, N. Birbaumer, D. McFarland, G. Pfurtscheller y T. Vaughan, «Brain Computer Interfaces for communication and control.,» *Clinical Neurophysiology*, vol. 113, pp. 767 - 791, 2002.
- [13] C. Guger, S. Daban, E. Sellers, C. Holzne, G. Krausz, R. Carabalona, F. Gramatica y G. Edlinger, «How many people are able to control a P300-based brain-computer interface (BCI)?,» *Neuroscience Letters*, vol. 462, pp. 94 - 98, 2009.
- [14] F. Akram, H. Han y T. Kim, «A P300-based brain computer interface system for words typing.,» *Computers in Biology and Medicine*, vol. 45, pp. 118 - 125, 2014.
- [15] A. Lopez, G. Pires y U. Nunes, «Assisted navigation for a brain-actuated intelligent wheelchair,» *Robotics and Autonomous Systems*, vol. 61, pp. 245-258, 2013.
- [16] G. Gentiletti, J. Gebhart, R. Acevedo, O. Yañez-Suarez y V. Medina-Bañuelos, «Command of a simulated wheelchair on a virtual environment using a brain-computer interface,» *IRBM*, vol. 30, pp. 218 - 115, 2009.
- [17] G. Pfurtscheller y F. H. Lopes da Silva, «Event-related EEG/EMG synchronization and desynchronization: basic principles.,» *Clinical Neurophysiology*, vol. 110, pp. 1842 - 1857, 1999.
- [18] J. Poblet, *Introducción a la bioingeniería.*, Primera ed., M. S. A., Ed., Marcombo S. A., 1988.
- [19] Trans Cranial Technologies, «10/20 System Positioning,» Trans Cranial Technologies Ltd., Wanchai, 2012.
- [20] R. Riki Rager, «Measurement and Application and Overview».
- [21] Universidad Autónoma de Nuevo León, [En línea]. Available: <http://gama.fime.uanl.mx/~mplatas/temas/tema6.pdf>.
- [22] Meditron, 2015. [En línea]. Available: <http://www.meditron.com.ve/s-76->

electroencefalografos.aspx.

- [23] Medscape, 2015. [En línea]. Available: <http://emedicine.medscape.com/article/1139483-overview>.
- [24] T.-P. Jung, C. Humphries, T.-W. Lee, S. Makeig, M. J. McKeown, V. Iragui y T. J. Sejnowski, «Extended ICA Removes Artifacts from Electroencephalographic Recordings,» *Advances in Neural Information Processing Systems*, vol. 10, pp. 894 - 900, 1998.
- [25] J. Iriarte, E. Urrestarazu, M. Valencia, M. Alegre, A. Malanda, C. Viteri y J. Artieda, «Independent Component Analysis as a Tool to Eliminate Artifacts in EEG: A Quantitative Study,» *Journal of Clinical Neurophysiology*, vol. 20, n° 4, pp. 249 - 257, Agosto 2003.
- [26] A. Delorme, T. Sejnowski y S. Makeig, «Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis,» *NeuroImage*, vol. 34, n° 4, pp. 1443 - 1449, Febrero 2007.
- [27] R. Nuno Vigário, «Extraction of ocular artefacts from EEG using independent component analysis,» *Electroencephalography and Clinical Neurophysiology*, vol. 103, n° 3, pp. 395 - 404, Septiembre 1997.
- [28] C. A. Joyce, I. F. Gorodnitsky y M. Kutas, «Automatic removal of eye movement and blink artifacts from EEG data using blind component separation,» *Psychophysiology*, vol. 41, n° 2, pp. 313 - 325, Marzo 2004.
- [29] L. Chin-Teng, K. Li-Wei, C. Jin-Chern, D. Jeng-Ren, H. Ruey-Song, L. Sheng-Fu, C. Tzai-Wen y J. Tzyy-Ping, «Noninvasive Neural Prostheses Using Mobile and Wireless EEG,» *Proceedings of the IEEE*, vol. 96, n° 7, pp. 1167 - 1183, Julio 2008.
- [30] G. N. Jayabhavani, N. R. Raajan y R. Rubini, «Brain mobile interfacing (BMI) system embedded with wheelchair,» de *Information & Communication Technologies (ICT), 2013 IEEE Conference on*, Islandia, 2013.
- [31] M. K. Hazrati, H. M. Husin y U. G. Hofmann, «Wireless brain signal recordings based on capacitive electrodes,» de *Intelligent Signal Processing (WISP), 2013 IEEE 8th International*

Symposium on, Funchal, 2013.

- [32] C. F. Reyes Manzano, B. A. Rico Jimenez y L. I. Garay Jimenez, «Sistemas mHealth para la adquisición de señales EEG,» *Research in Computing Science*, vol. 75, pp. 91 - 101, Julio 2014.
- [33] OPENSURG, ROBÓTICA MÉDICA, notas prácticas para el aprendizaje de la robótica en bioingeniería, Elche: CYTED, 2013.
- [34] J. C. Montoya Cardona, «SISTEMA DE ADQUISICIÓN DE BIOPOTENCIALES PARA ENTORNOS ACADÉMICOS,» 2013.
- [35] E. R. Kander, J. H. Schwartz y T. M. Jessell, *Neurociencia y Conducta*, Segunda ed., J. Stumpf, Ed., Prentice Hall, 1999.
- [36] A. L. Garza Lerma, A. Vazquez Ibarra y R. Guzman Dionisio, «Biopotenciales,» Universidad Autónoma de Ciudad Juárez, Ciudad Juárez.
- [37] J. E. Aragón Puerto y W. Calleja Arriaga, «Fabricación y caracterización eléctrica de microelectrodos de silicio para registro de señales nerviosas,» *Revista Mexicana de Ingeniería Biomédica*, vol. XXIV, n° 2, pp. 126 - 134, 2003.
- [38] I. A. Cifuentes Gonzáles, *Diseño y construcción de un sistema para la detección de señales electromiográficas*, Mérida, Yucatán: Universidad Autónoma de Yucatán, 2010.
- [39] L. Azcona, *Libro de la salud cardiovascular: 4. El electrocardiograma*, Madrid: Hospital Clínico San Carlos.
- [40] Red de Salud de Cuba, «El Electrocardiograma: Componentes, Valores normales y Semiología de sus perturbaciones,» Red de Salud de Cuba, La Habana.
- [41] K. Nazarpour, A. H. Al-Timemy, G. Bugmann y A. Jackson, «A note on the probability distribution function of the surface electromyogram signal,» *Brain Research Bulletin*, vol. 90, pp. 88 - 91, 2013.

- [42] C. A. Alva Coras, *Procesamiento de señales de electromiografía superficial para la detección de movimiento de dos dedos de la mano*, Lima: Universidad Ricardo Palma, 2012.
- [43] K. Najarian y R. Splinter, *Biomedical Signal and Image Processing*, CRC Press , 2005.
- [44] A. Jackson y D. Bolger, «The neurophysiological bases of EEG and EEG measurement: A review for the rest of us,» *Psychophysiology*, nº 51, pp. 1061 - 1071, 2014.
- [45] M. A. Guevara, A. Sanz-Martín, M. Corsi-Cabrera, C. Amezcua-Gutiérrez y M. Hernández-González, «CHECAsEN: programa para revisar señales EEG,» *Revista Mexicana de Ingeniería Biomédica*, vol. 31, nº 2, pp. 135 - 141, Diciembre 2010.
- [46] Universidad de Alcalá Departamento de Electrónica, «Tema 5. Electroencefalografía».
- [47] G. Borja, T. Ortega y A. Romero, «Diseño e implementación de un equipo para la adquisición y visualización en pc de señales electroencefalográficas,» *Prospect*, vol. 8, nº 1, pp. 21 - 28, Enero - Julio 2010.
- [48] Privación del sueño, Mayo 2015. [En línea]. Available: <https://dameunsilbidito.wordpress.com/2011/10/02/privacion-de-sueno/>.
- [49] L. Alvarez Osorio, «Acondicionamiento de Señales Bioeléctricas,» 2007.
- [50] L. E. Morillo, «Análisis Visual del EEG».
- [51] M. A. Idiazábal, A. B. Palencia-Taboada, J. Sangorrin y J. M. Espadeler-Gamissans, «Potenciales evocados cognitivos en el trastorno por déficit de atención con hiperactividad.,» *Revista Neurológica*, vol. 34, nº 4, p. 301, 2002.
- [52] M. A. Guevara, J. Jiménez, D. Velázquez, F. Ostrosky, M. Pérez, Y. Rodriguez y C. García de la Cadena, «Un sistema para la adquisición y el análisis automatizado de potenciales evocados en humanos,» *Revista Mexicana de Psicología*, vol. 10, nº 2, pp. 189 - 194, 1993.
- [53] Agència Valenciana de Salut, «Potenciales Evocados,» Agència Valenciana de Salut, Valencia.

- [54] J. Arcos, E. García y R. Torres, «Evaluación experimental y estadística de un prototipo de interfaz cerebro-computador,» *Ingeniería Biomédica*, vol. 4, nº 8, pp. 22 - 33, 2010.
- [55] J. De la Torre, *Procesado de señales EEG para un interfaz cerebro-máquina (BCI)*, Madrid: Universidad Carlos III de Madrid, 2012.
- [56] J. Gutierrez y A. Schlögl, «Comparación de Métodos Autorregresivos para la Detección de Artefactos en Señales EEG».
- [57] B. Fisch, «EEG Artifacts,» LSU Medical Center, 2005.
- [58] Neurofisiología Granada, Mayo 2015. [En línea]. Available: <http://neurofisiologiagranada.com/eeg/eeg-actividadcerebral.htm>.
- [59] R. J. Croft y R. J. Barry, «Removal of ocular artifact from the EEG: a review,» *Neurophysiol Clin*, vol. 30, pp. 5-19, 2000.
- [60] A. Estupiñan Donoso, «Reducción de artefactos oculares en señales EEG: Filtrado adaptativo como alternativa a la regresión lineal,» 2009.
- [61] J. J. González Murillo, «Trabajo Práctico 2: Filtrado de Señales,» Universidad Politécnica de Catalunya, Barcelona, 2014.
- [62] F. Abd Rahman, M. Fauzi Othman y N. Aimi Shaharuddin, «A review on the current state of artifact removal methods for electroencephalogram signals,» de *Control Conference (ASCC), 2015 10th Asian*, Asia, 2015.
- [63] A. Kachenoura, D. Safieddine, L. Albera, G. Birot, F. Wendling, L. Senhadji y I. Merlet, «Blind source separation methods applied to muscle artifacts removing from epileptic EEG recording: a comparative study,» de *Colloque National Recherche en Imagerie et. Technologies pour la Santé*, Rennes, Francia, 2011.
- [64] S. Tavildar y A. Ashrafi, «Application of Multivariate Empirical Mode Decomposition and Canonical Correlation Analysis for EEG motion Artifact Removal,» de *2016 Conference on Advances in Signal Processing*, Pune, 2016.

- [65] C. Chia-Ching, C. Tsan-Yu y F. Wai-Chi, «FPGA Implementation of EEG System-on-Chip with Automatic Artifacts Removal based on BSS-CCA Method,» de *Biomedical Circuits and Systems Conference*, Shanghai, 2016.
- [66] K. T. Sweeney, S. F. McLoone y T. E. Ward, «The use of Ensemble Empirical Mode Decomposition with Canonical Correlation Analysis as a Novel Artifact Removal Technique,» *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 60, pp. 97 - 105, 2013.
- [67] W. De Clercq, A. Vergult, B. Vanrumste, W. Van Paesschen y S. Van Huffer, «Canonical Correlation Analysis Applied to Remove Muscle Artifacts From the Electroencephalogram,» *TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 53, n° 12, 2006.
- [68] Emotiv, «Emotiv Brainware,» Emotive, 2017. [En línea]. Available: <https://www.emotiv.com/>. [Último acceso: 16 Febrero 2017].
- [69] D. Prince, M. Edmonds, A. Sutter, M. Cusumano, W. Lu y V. Asari, «Brain Machine Interface Using Eotiv EPOC to Control Robai Cyton Robotic Arm,» *IEEE*, 2015.
- [70] S. Aguiar, W. Yáñez y D. Benítez, «Low Complexity Approach for Controlling a Robotic Arm Using the Emotiv EPOC Headset,» *IEEE International Autum Meeting on Power, Electronics and Computing*, 2016 .
- [71] R. Raju, C. Yang, C. Li y A. Cangelosi, «A Video Game Design based on Emotiv Neuroheadset,» de *International Conference on Advanced Robotics and Mechatronics*, 2016.
- [72] D. Benítez, S. Toscano y A. Silva, «On the Use of the Emotiv EPOC Neuroheadset as a Low Cost Alternative for EEG Signal Acquisition,» *IEEE COLCOM*, 2016.
- [73] A. Ghodake y S. D. Shelke, «Brain Controlled Home Automation System,» *IEEE*, 2016.
- [74] J. Cheng, G. Mabsa y C. Oppus, «Prolonged Distraction Testing Game Implemented with ImpactJS HTML5, Gamepad and Neurosky,» de *7th IEEE International Conference Humanoid, Nanotechnology, Information Technology Communication and Control*,

Environment and Management, Filipinas, 2014.

- [75] J. Katona, T. Ujbanyi, G. Sziladi y A. Kovari, «Speed control of Festo Robotino mobile robot using NeuroSky MindWave EEG headset based Brain-Computer Interface,» de *7th IEEE International Conference on Cognitive Infocommunications*, Polonia, 2016.
- [76] D. Bright, A. Nair, D. Salvekar y S. Bhisikar, «EEG-Based Brain Controlled Prosthetic Arm,» de *Conference on Advances in Signal Processing*, 2016.
- [77] OpenBCI, «Open BCI,» 2017. [En línea]. Available: www.openbci.com. [Último acceso: 16 Febrero 2017].
- [78] L. Losonczi, L. Márton, T. Brassai y L. Farkas, «Embedded EEG signal acquisition systems,» de *7th International Conference Interdisciplinarity in Engineering*, Romania, 2013.
- [79] D.-G. Kim, K.-S. Hong y K.-W. Chung, «Implementation of Portable Multi-Channel EEG and Head Motion Signal Acquisition System,» *Journal of Next Generation Information Technology*, vol. 3, n° 4, 2012.
- [80] S. I. Arman, A. Ahmed y A. Syed, «Cost-Effective EEG Signal Acquisition and Recording System,» *International Journal of Bioscience, Biochemistry and Bioinformatics*, vol. 2, n° 5, 2012.
- [81] N. A. Noorazman y N. H. Aziz, «Portable EEG Signal Acquisition System,» *College Science in India*, 2009.
- [82] G. Kelly, «Development of a compact, low-cost wireless device for biopotential acquisition,» Virginia Commonwealth University, Virginia, 2014.
- [83] A. L. Goldberg, L. Amaral, L. Glass, J. M. Hausdorf, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng y H. Stanley, «PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals.,» *Physionet*, 13 Junio 2000. [En línea]. Available: <https://physionet.org/pn4/eegmmidb/>. [Último acceso: 28

Enero 2017].

- [84] T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. J. McKeon, V. Iragui y T. J. Sejnowski, «Removing electroencephalographic artifacts by blind source separation,» *Psychophysiology*, pp. 163-178, 2000.
- [85] W. Zhou y J. Gotman, «Automatic removal of eye movement artifacts from the EEG using ICA and the dipole model,» *Progress in Natural Science*, vol. 19, pp. 1165 - 1170, 2009.
- [86] M. Chaumon, D. Bishop y N. Busch, «A practical guide to the selection of independent components of the electroencephalogram for artifact correction,» *Journal of Neuroscience Methods*, pp. 47 - 63, 2015.
- [87] J. Kelly, D. Siewiorek, A. Smailagic, J. Collinger, D. Weber y W. Wang, «Fully Automated Reduction of Ocular Artifacts in High-Dimensional Neural Data,» *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 58, n° 3, 2011.
- [88] C. Burger y D. J. Van den Heever, «Removal of EOG artefacts by combining wavelet neural network and independent component analysis,» *Biomedical Signal Processing and Control*, pp. 67 - 79, 2015.
- [89] I. Daly, R. Scherer, M. Billinger y G. Müller-Putz, «FORCe: Fully Online and Automated Artifact Removal for Brain-Computer Interfacing,» *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, vol. 23, n° 5, 2015.
- [90] W. Kong, Z. Zhou, S. Hu, J. Zhang, F. Babiloni y G. Dai, «Automatic and Direct Identification of Blink Components from Scalp EEG,» *Sensors*, pp. 10783-10801, 2013.
- [91] B. Mijovic, M. De Vos, I. Gligorrijevic, J. Taelman y S. Van Huffel, «Source Separation From Single-Channel Recordings by Combining Empirical-Mode Decomposition and Independent Component Analysis,» *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 57, n° 9, 2010.
- [92] B. Nouredin, P. Lawrence y G. Birch, «Online Removal of Eye Movement and Blink EEG Artifacts Using a High-Speed Eye Tracker,» *IEEE TRANSACTIONS ON BIOMEDICAL*

ENGINEERING, vol. 59, nº 8, 2012.

- [93] A. Garcés Correa, E. Laciari, H. D. Patiño y M. E. Valentinuzzi, «Artifact removal from EEG signals using adaptive filters in cascade,» de *16th Argentine Bioengineering Congress and the 5th Conference of Clinical Engineering*, Argentina, 2007.
- [94] G. Geetha y S. N. Geethalakshmi, «Artifact Removal from EEG using Spatially Constrained Independent Component Analysis and Wavelet Denoising with Otsu's Thresholding Technique,» de *International Conference on Communication Technology and System Design*, 2011.
- [95] N. Siew-Cheok y P. Raveendran, «Enhanced μ Rhythm Extraction Using Blind Source Separation and Wavelet Transform,» *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 56, nº 8, 2009.
- [96] H. Zeng, A. Song, R. Yan y H. Qin, «EOG Artifact Correction from EEG Recording Using Stationary Subspace Analysis and Empirical Mode Decomposition,» *Sensors*, pp. 14839 - 14859, 2013.
- [97] K. L. Molla, T. Tanaka, T. M. Rutkowski y A. Cichocki, «SEPARATION OF EOG ARTIFACTS FROM EEG SIGNALS USING BIVARIATE EMD,» de *Acoustics, Speech, and Signal Processing*, 2010.
- [98] V. Lawhern, D. Hairston, K. McDowell, M. Westerfield y K. Robbins, «Detection and classification of subject-generated artifacts in EEG signals using autoregressive models,» *Journal of Neuroscience Methods*, pp. 181 - 189, 2012.
- [99] A. Schlogl, C. Keinrath, D. Zimmermann, R. Scherer, R. Leeb y G. Pfurtscheller, «A fully automated correction method of EOG artifacts in EEG recordings,» *Clinical Neurophysiology*, pp. 98 - 104, 2007.
- [100] Texas Instruments, *ADS1299: Low-Noise, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements*, 2012.

- [101] Texas Instruments, *EEG Front-End Performance Demonstration Kit*, 2014.
- [102] ARM Mbed, «ARM Mbed,» 2017. [En línea]. Available: <https://www.mbed.com/en/>.
- [103] Eigen, «Eigen C++,» 16 Junio 2017. [En línea]. Available: http://eigen.tuxfamily.org/index.php?title=Main_Page. [Último acceso: Julio 2017].

ANEXO 1.

CONFIGURACIÓN POR DEFECTO DE LA TARJETA DE EVALUACIÓN ADS1299 EVM

Nombre del Jumper	Posición	Comentario
JP1	No instalado	Usado para programar la configuración del BIAS
JP2	2-3	Alimentación unipolar análoga (AVDD = 5V)
JP3	No instalado	Relacionada con el circuito externo de referencia
JP4	1-2	Alimentación a la tarjeta de 5V
JP5	No instalado	Opción para proveer la señal PWDN desde la programación
JP6	1-2	El BIAS_ELEC se conecta al voltaje de media alimentación de la tarjeta.
JP7	1-2	Dirige el REF_ELEC a la entrada del buffer
JP8	1-2	Dirige el REF_ELEC directamente a SRB1
JP17	No instalado	Relacionado con el circuito de escudo
JP18	2-3	Reloj desde el oscilador de la tarjeta de evaluación
JP19	1-2	Alimentación del oscilador de la tarjeta de evaluación
JP20	1-2	Alimentación unipolar (AVSS = 0)
JP21	1-2	
JP22	2-3	
JP23	1-2	CLKSEL = 0
JP24	3-2	Alimentación digital (DVDD = 3.3V)
JP25	1-2	BIAS_ELEC conectado a todos los INPs
	3-4	BIAS_ELEC conectado a todos los INNs
	5-6	BIAS_ELEC en corto circuito con REF_ELEC que se conecta con SRB1
J6	5-6/ 7-8 / 9 -10 ... /35-36	Conectar jumpers en todos los canales

ANEXO 2

PROCESAMIENTO DE SEÑALES DE LA BASE DE DATOS DE PHYSIONET EN MATALB®

```
%% PROCESAMIENTO DE SEÑALES DE BASE DE DATOS PHYSIONET
close all
%% OBTENCIÓN DE LOS DATOS Y CREACIÓN DE LA SEÑAL ORIGINAL

% La matriz "val" es la matriz que proviene de cada uno de los registros de
% la base de datos.

% La matriz X es la que se se va a procesar, inicialmente contiene los 64
% electrodos.

x=val(1:64,:); % Matriz de la base de datos
X=x';
figure;
plot((X))
axis([800 1100 -250 250])
title('Señal Original (64 Canales)')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F1')

%% DESCOMPOSICIÓN DE LA SEÑAL MÉTODO BSS -CCA:

% Descomposición de la señal por el método BSS-CCA a través del algoritmo
% que se encuentra en el Anexo 2.

[Wc,rc] = bsscca(X',1) % El retraso del método se
% sugiere en 1. La matriz Wc es
% la matriz de desmezcla y el
% vector rc es el vector de
% correlaciones.

Yc=Wc*X';

% Se realiza una búsqueda dentro del vector de correlaciones y de acuerdo
% con lo relacionado al método BSS-CCA se eliminan las componentes que
% tengan una correlación menor a 0.75.

for i=1:length(rc)
    if rc(i)<= 0.75
        Yc(i,:)=0;
    end
end

% Proceso de reconstrucción de la señal después de eliminar componentes

Ac=inv(Wc);
Xc=Ac*Yc; % Al realizar esta operación
% se obtiene una señal Xc que
% es una estimación de la señal
% original con los artefactos
% eliminados.

Xc=Xc';
figure, plot((Xc))
axis([800 1100 -1000 1000])
title('Señal Reconstruida (64 Canales)')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F2')

%% FILTRADO DE LA SEÑAL RECONSTRUIDA

Fs=160; % Frecuencia de muestreo de la
% base de datos de Physionet.
```

```

% El filtro Butterworth Pasa Banda con frecuencias de corte de 0.5 - 45 Hz
% para hacer una ventana de las frecuencias cerebrales útiles.
Wn=[0.5 35]/(Fs/2);
[b,a]=butter(8,Wn,'bandpass');
sig_pb=filter(b,a,Xc); % Señal procesada sin
                        % artefactos utilizando el
                        % método BSS-CCA.

figure, plot(sig_pb)
axis([800 1100 -1000 1000])
title('Señal Final (64 Canales)')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F3')

%% PROCESAMIENTO DE LAS SEÑALES DE LOS ELECTRODOS SELECCIONADOS

% Se extraen los 6 canales de los electrodos seleccionados.

x=x';
Fp1=x(:,22); % Electrodo Fp1
Fp2=x(:,24); % Electrodo Fp2
F3=x(:,32); % Electrodo F3
F4=x(:,36); % Electrodo F4
C3=x(:,9); % Electrodo C3
C4=x(:,13); % Electrodo C4
X=[Fp1 Fp2 F3 F4 C3 C4]; % Matriz para procesar
figure, plot(X)
axis([800 1100 -250 250])
title('Señal Original (6 Canales)')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F4')

%% DESCOMPOSICIÓN DE LA SEÑAL MÉTODO BSS - CCA:

% Descomposición de la señal por el método BSS-CCA a través del algoritmo
% que se encuentra en el Anexo 2.

[Wc,rc] = bsscca(X',1) % El retraso del método se
                        % sugiere en 1. La matriz Wc es
                        % la matriz de desmezcla y el
                        % vector rc es el vector de
                        % correlaciones.

Yc=Wc*X';

% Gráfica de las componentes junto con el vector de correlaciones.
figure
subplot 611, plot(Yc(1,:), 'b')
title('Componentes de la Señal (6 Canales)')
subplot 612, plot(Yc(2,:), 'm')
subplot 613, plot(Yc(3,:), 'g')
subplot 614, plot(Yc(4,:), 'k')
subplot 615, plot(Yc(5,:), 'r')
subplot 616, plot(Yc(6,:), 'c')
xlabel('No. Muestras');
print(gcf, '-dpng', 'F5')

figure, plot(rc, '-ok')
line=[1 2 3 4 5 6];
liney=[0.75 0.75 0.75 0.75 0.75 0.75];
hold on
plot(line, liney, '--r')
axis([1 6 0 1])
title('Correlación de las componentes (6 Canales)')
xlabel('No. Componentes');
ylabel('Valor de Correlación');
print(gcf, '-dpng', 'F6')

```



```

% Se realiza una búsqueda dentro del vector de correlaciones y de acuerdo
% con lo relacionado al método BSS-CCA se eliminan las componentes que
% tengan una correlación menor a 0.75.

for i=1:length(rc)
    if rc(i)<= 0.75
        Yc(i,:)=0;
    end
end

% Proceso de reconstrucción de la señal después de eliminar componentes

Ac=inv(Wc);
Xc=Ac*Yc;

% Al realizar esta operación
% se obtiene una señal Xc que
% es una estimación de la señal
% original con los artefactos
% eliminados.

Xc=Xc';
figure, plot(Xc)
axis([800 1100 -250 250])
title('Señal Reconstruida (6 Canales)')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F7')

%% COMPARACIÓN ENTRE LA SEÑAL ORIGINAL Y LA RECONSTRUIDA
figure, plot(X(:,1))
axis([800 1100 -250 250])
hold on
plot(Xc(:,1))
title('Señal Original vs Señal Reconstruida Canal No. 1')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F8')

figure, plot(X(:,2))
axis([800 1100 -250 250])
hold on
plot(Xc(:,2))
title('Señal Original vs Señal Reconstruida Canal No. 2')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F9')

figure, plot(X(:,3))
axis([800 1100 -250 250])
hold on
plot(Xc(:,3))
title('Señal Original vs Señal Reconstruida Canal No. 3')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F10')

figure, plot(X(:,4))
axis([800 1100 -250 250])
hold on
plot(Xc(:,4))
title('Señal Original vs Señal Reconstruida Canal No. 4')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F11')

figure, plot(X(:,5))
axis([800 1100 -250 250])
hold on
plot(Xc(:,5))
title('Señal Original vs Señal Reconstruida Canal No. 5')
xlabel('No. Muestras');

```

```

ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F12')

figure, plot (X(:,6))
axis([800 1100 -250 250])
hold on
plot (Xc(:,6))
title('Señal Original vs Señal Reconstruida Canal No. 6')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F13')

%% FILTRADO DE LA SEÑAL RECONSTRUIDA

Fs=160; % Frecuencia de muestreo de la
        % base de datos de Physionet.

% El filtro Butterworth Pasa Banda con frecuencias de corte de 0.5 - 35 Hz
% para hacer una ventana de las frecuencias cerebrales útiles.
Wn=[0.5 35]/(Fs/2);
[b,a]=butter(8,Wn,'bandpass');
sig_pb=filter(b,a,Xc); % Señal procesada sin
                       % artefactos utilizando el
                       % método BSS-CCA.

figure, plot(sig_pb)
axis([800 1100 -250 250])
title('Señal Final (6 Canales)')
xlabel('No. Muestras');
ylabel('Amplitud (uV)');
print(gcf, '-dpng', 'F14')

%% FIN DEL CÓDIGO

```

ANEXO 3.

ALGORITMO DE DESCOMPOSICIÓN DE LA SEÑAL POR MÉTODO BSS-CCA.

```
function [W,r] = bsscca(X,delay)
% bsscca() - Blind Source Separation through Canonical Correlation Analysis
%
% Usage:
%   >> [W,r] = bsscca(X,delay)
%
% Inputs:
%   X      - data matrix (dxN, data channels are rowwise)
%   delay  - delay at which the autocorrelation of the sources will be
%           maximized (def: 1)
%
% Output:
%   W      - separation matrix
%   r      - autocorrelation of the estimated sources at the given delay
%
% See also:
%   BSSCCA_IFC, AUTOBSS

if nargin < 2, delay = 1; end
if nargin < 1,
    help cca;
    return;
end

[d,T] = size(X);

% correlation matrices
Y = X(:,delay+1:end);
X = X(:,1:end-delay);
Cyy = (1/T)*Y*Y';
Cxx = (1/T)*X*X';
Cxy = (1/T)*X*Y';
Cyx = Cxy';
invCyy = pinv(Cyy);

% calculate W
[W,r] = eig(pinv(Cxx)*Cxy*invCyy*Cyx);
r = sqrt(abs(real(r)));
[r,I] = sort(diag(r), 'descend');
W = W(:,I)';

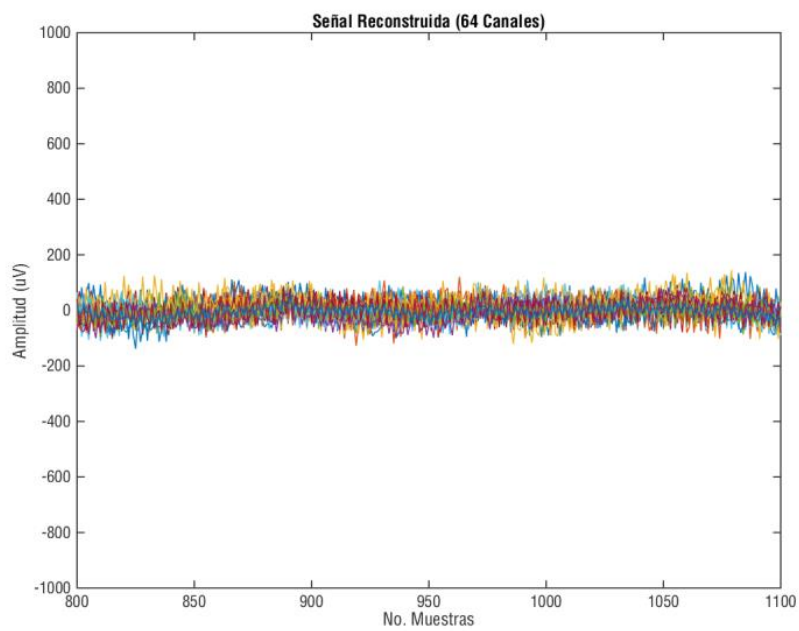
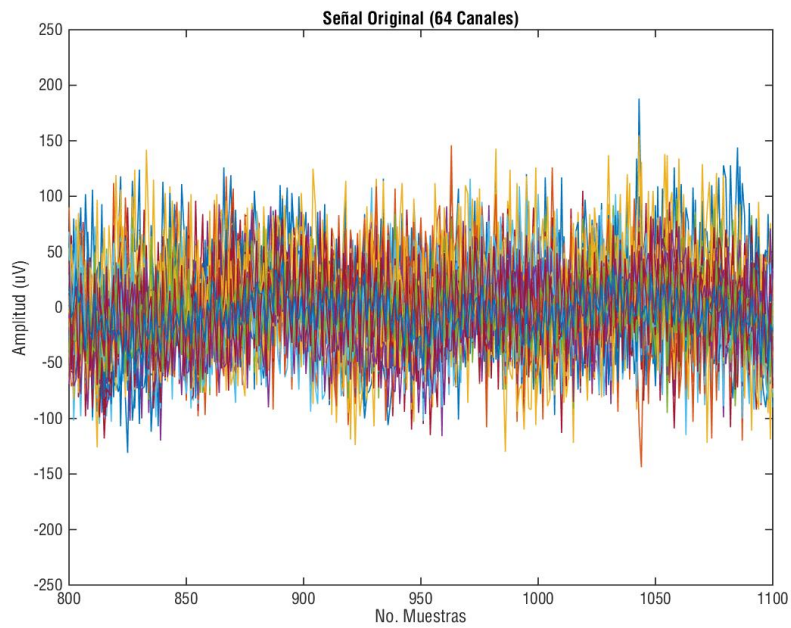
% Copyright (C) <2007> German Gomez-Herrero, http://germangh.com
```

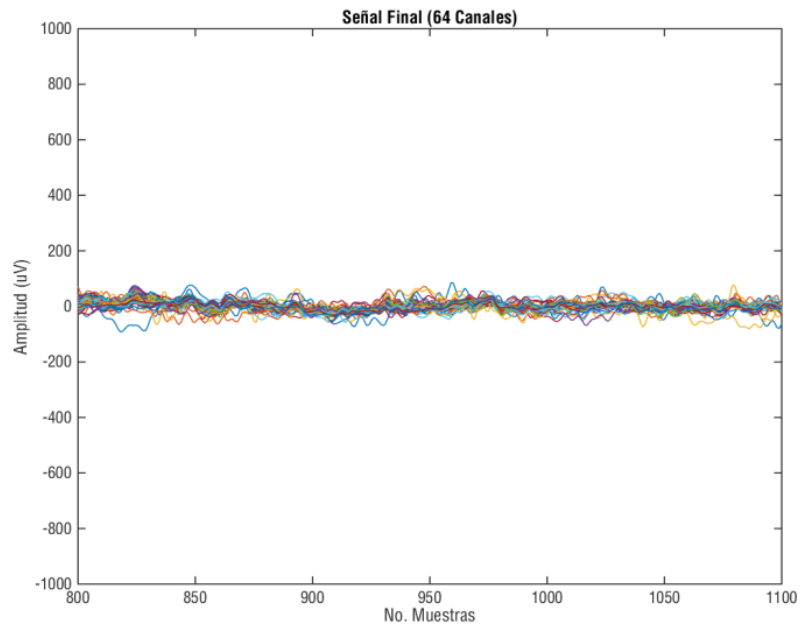
ANEXO 4.

RESULTADOS DE LA IMPLEMENTACIÓN DEL PROCESAMIENTO A LAS SEÑALES DE LA BASE DE DATOS PHYISIONET.

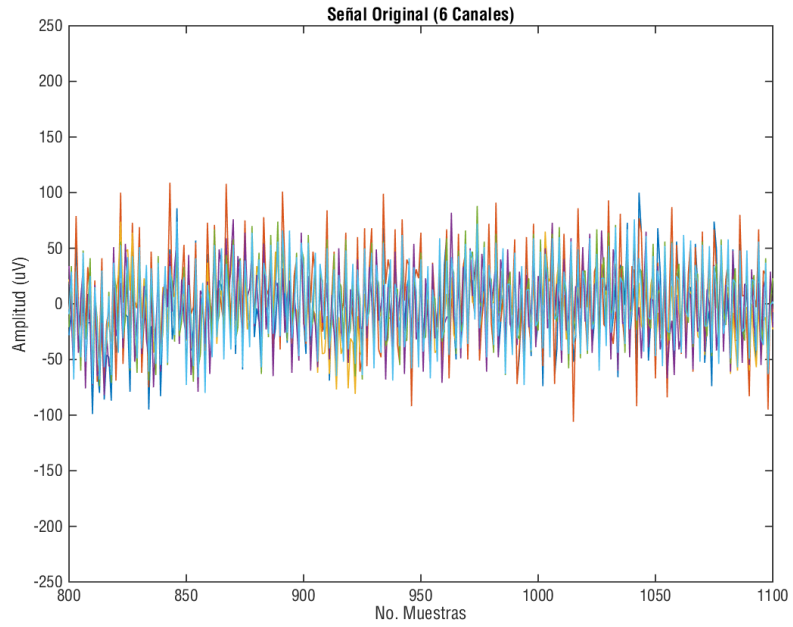
Registro Individuo número 05 tarea 05:

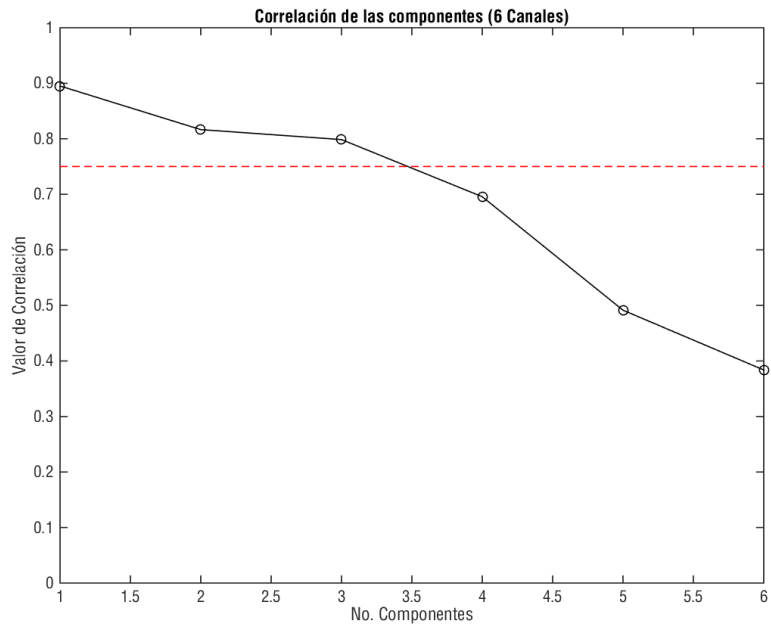
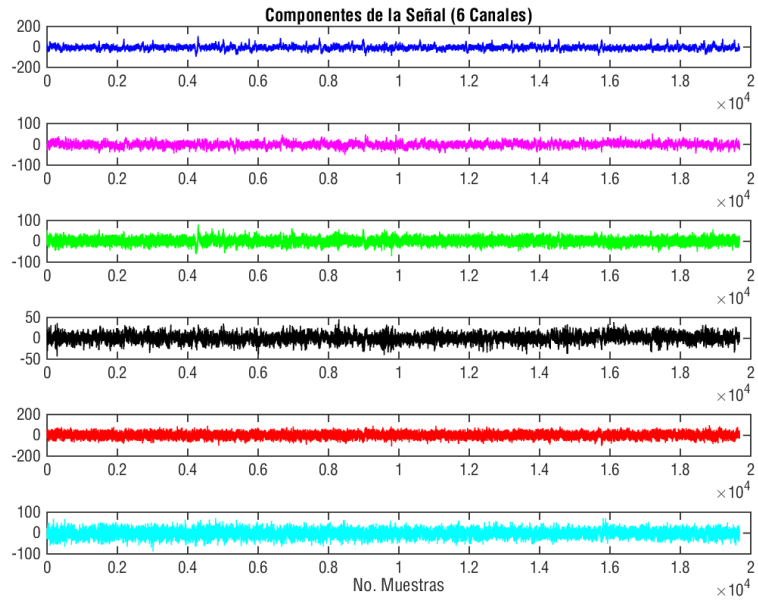
Análisis 64 Canales:

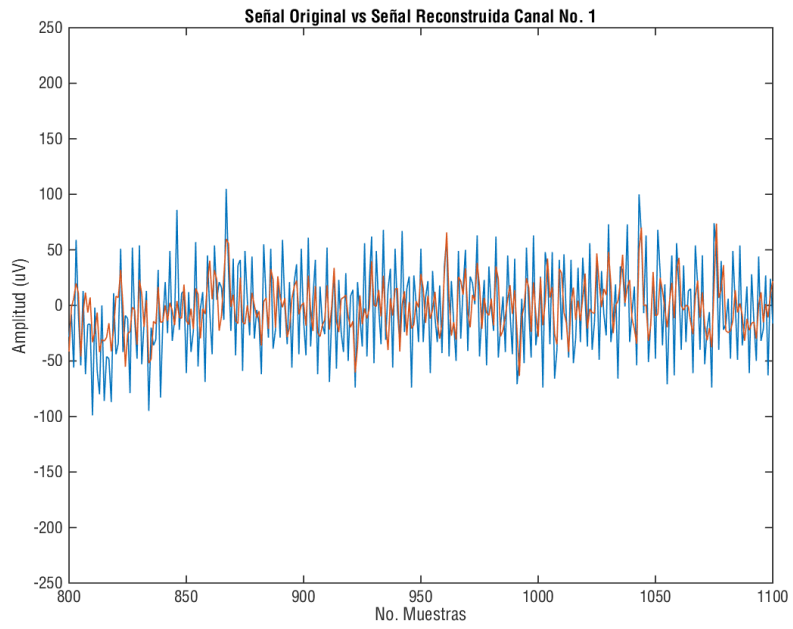
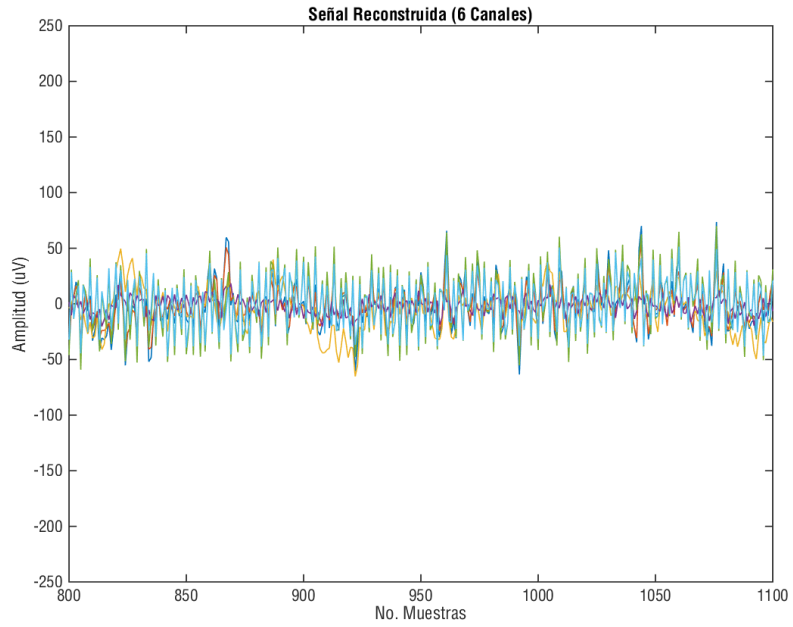


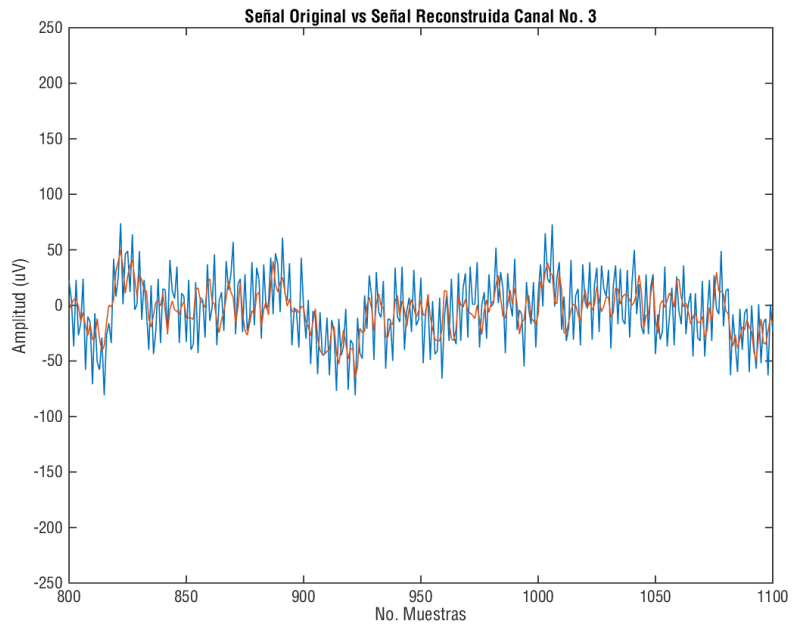
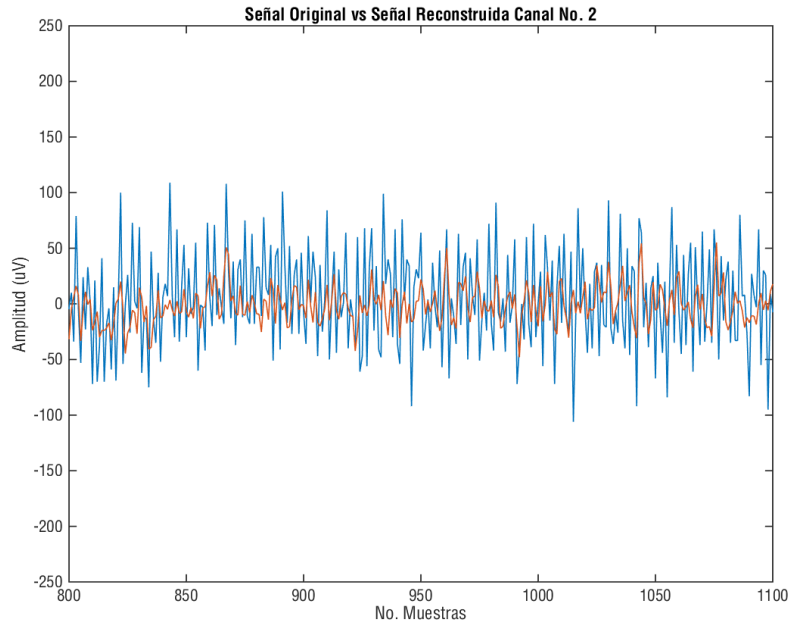


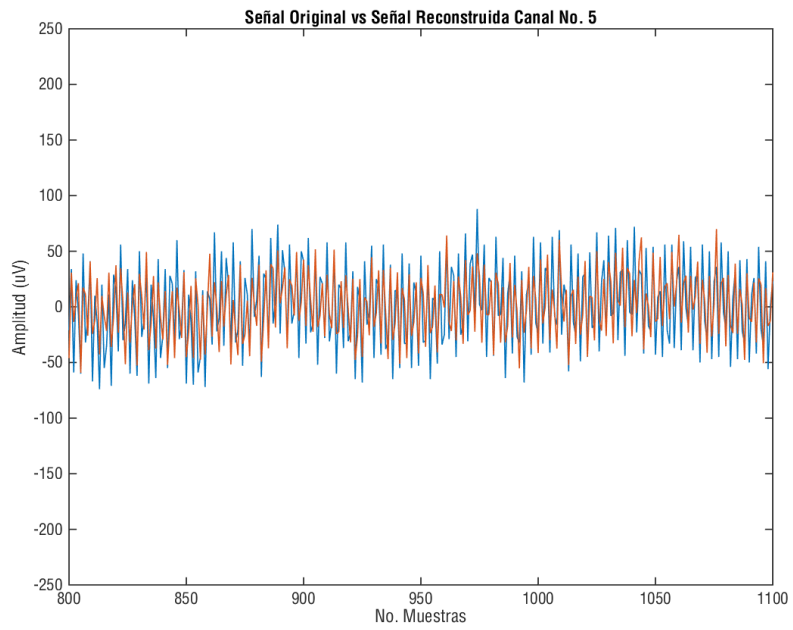
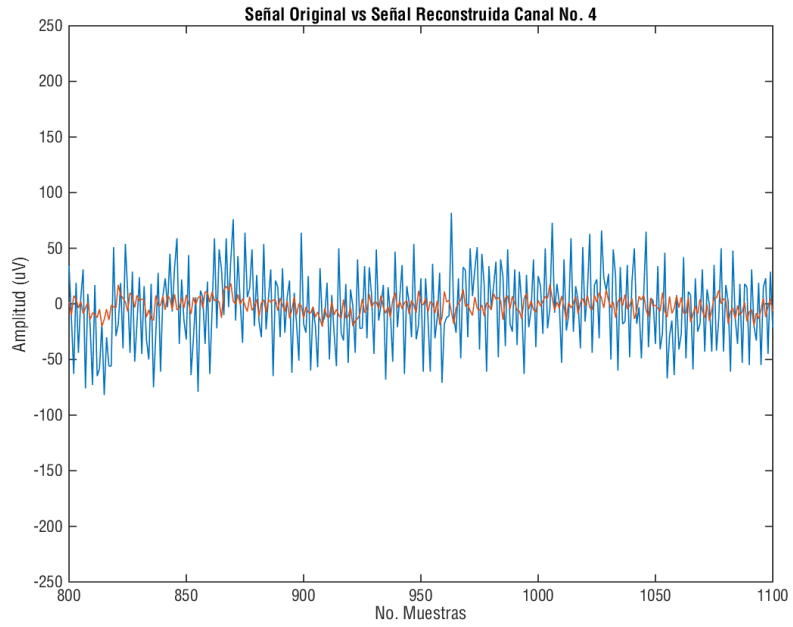
Análisis 6 Canales:



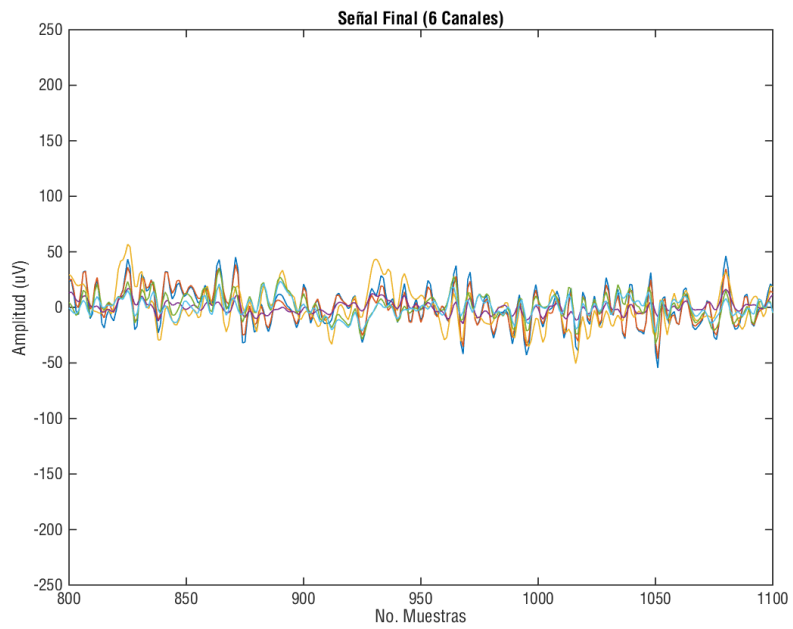
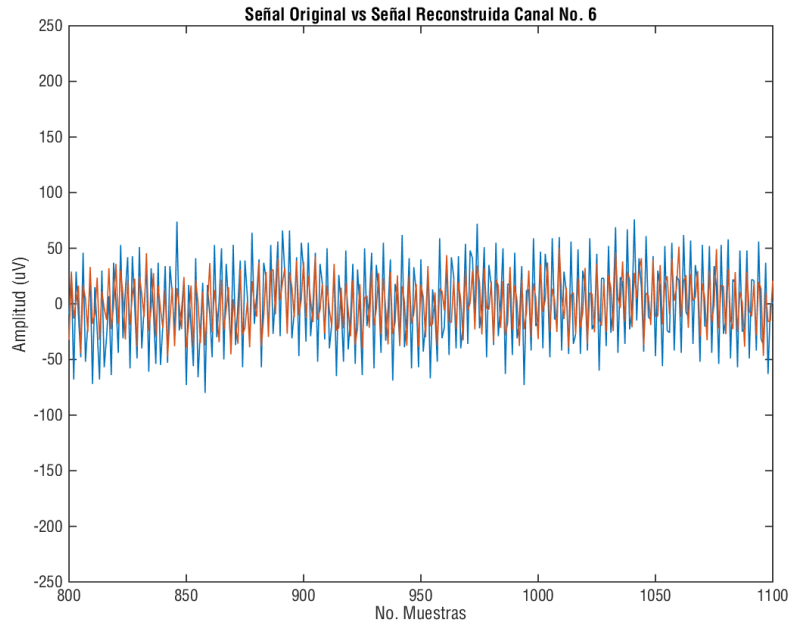






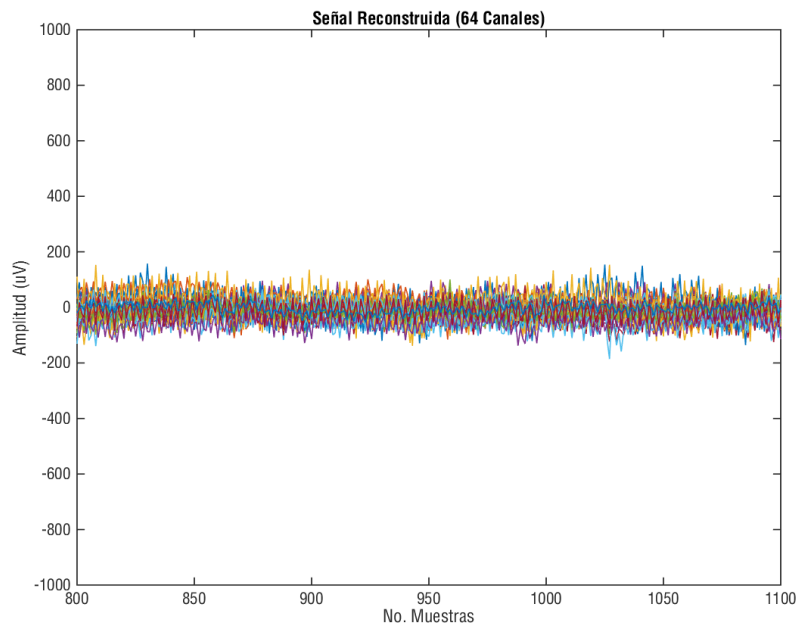
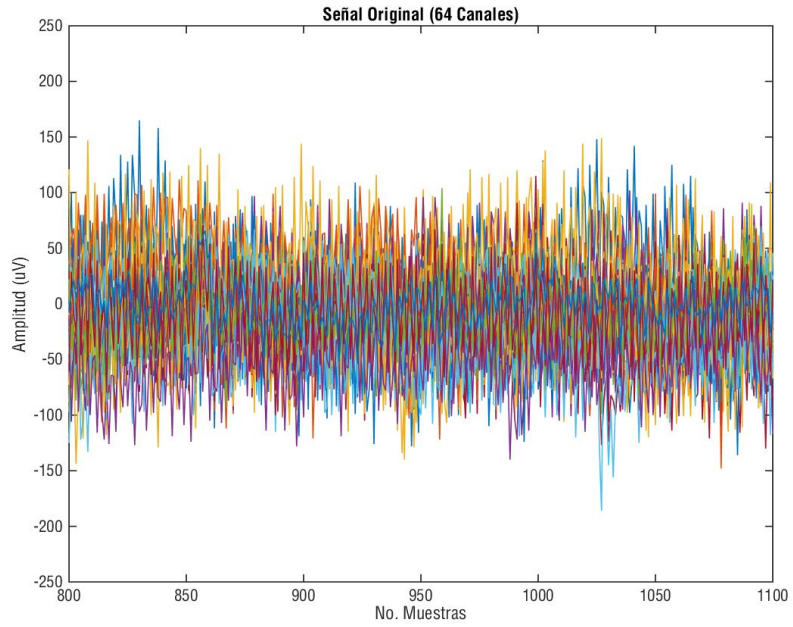


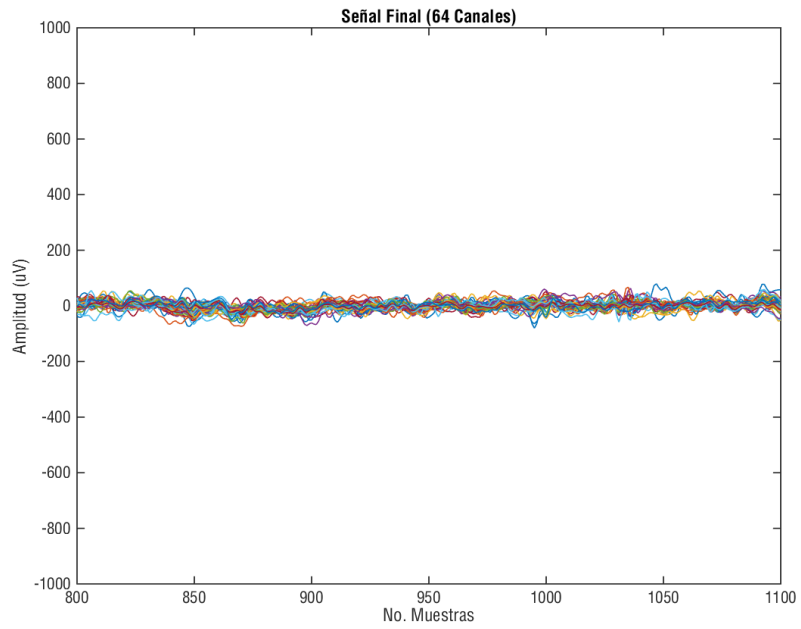
|



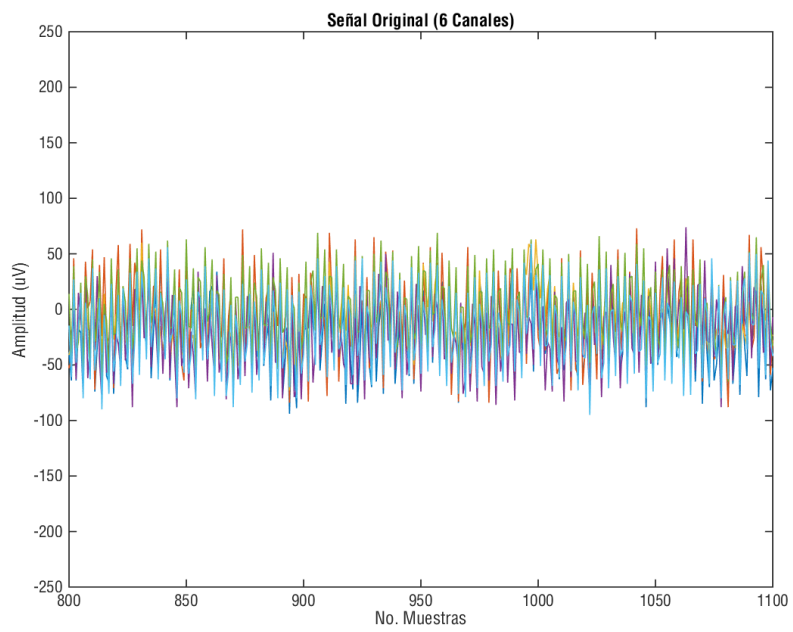
Registro Individuo número 05 tarea 09:

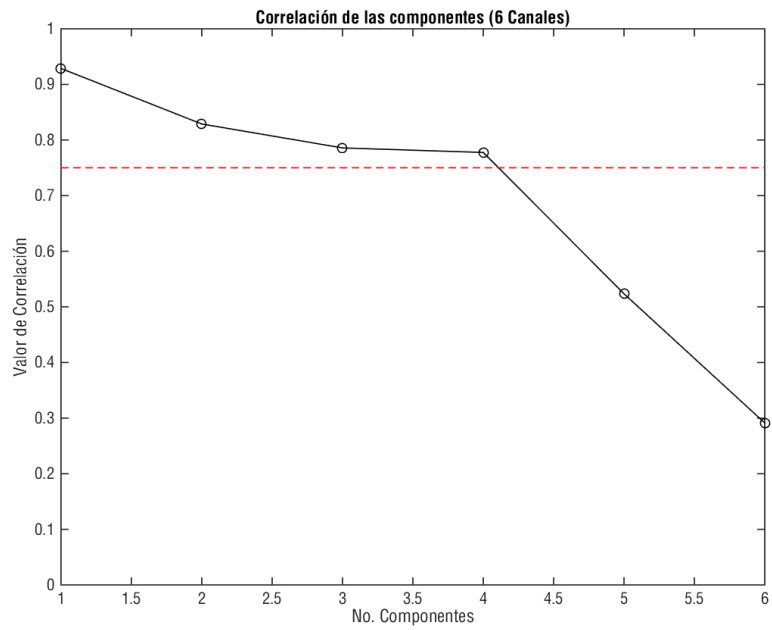
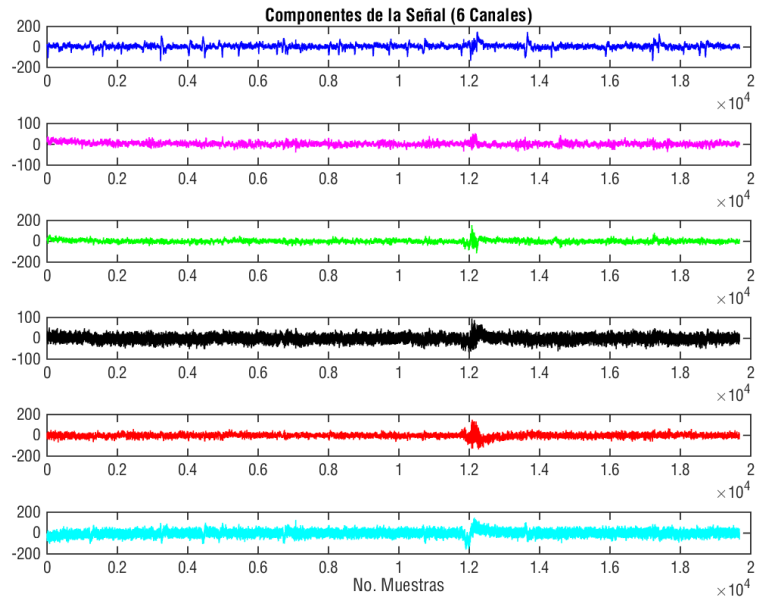
Análisis 64 Canales:

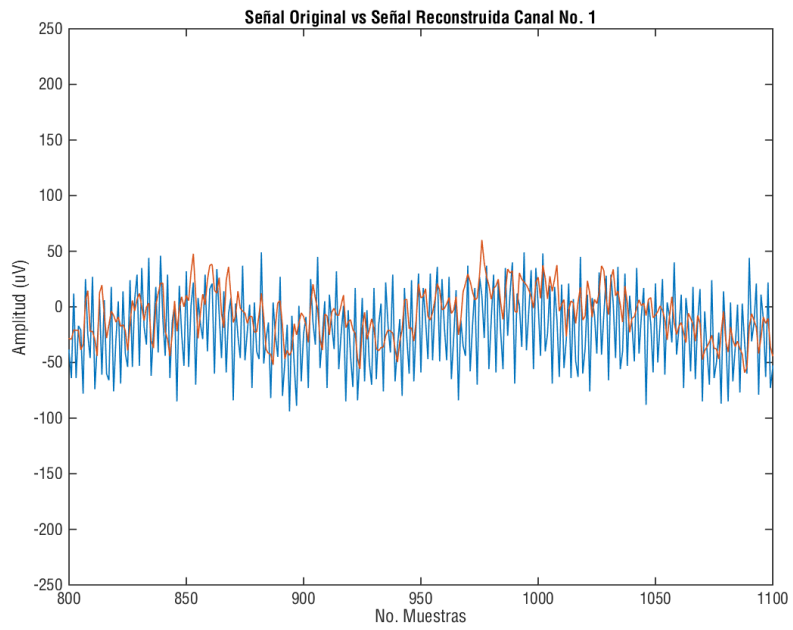
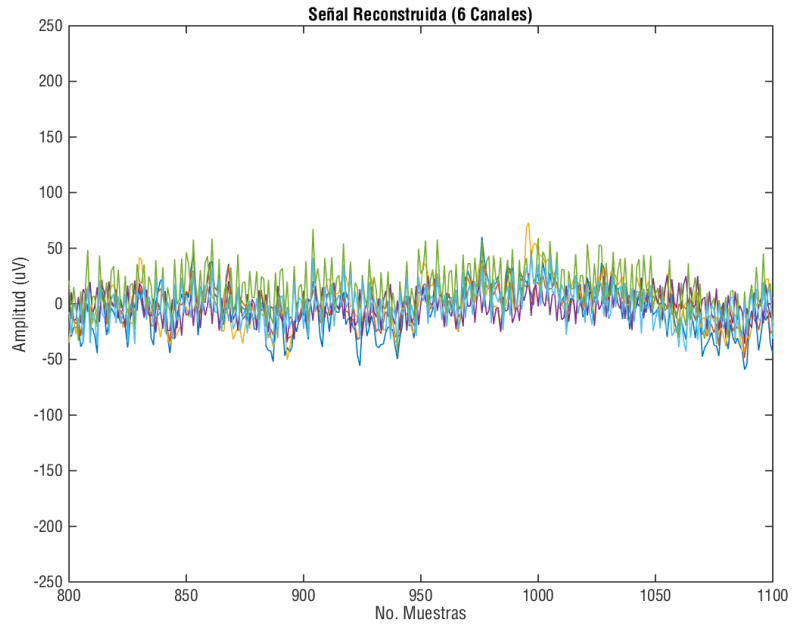


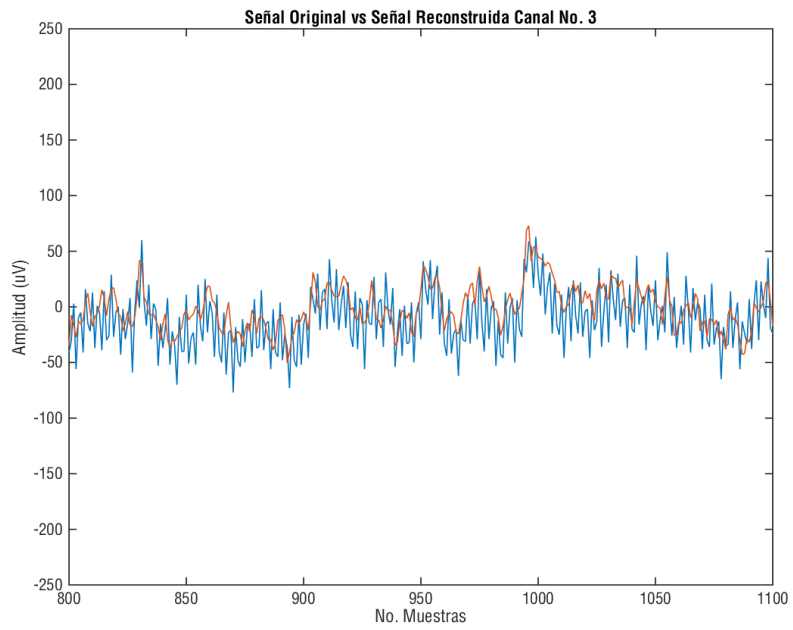
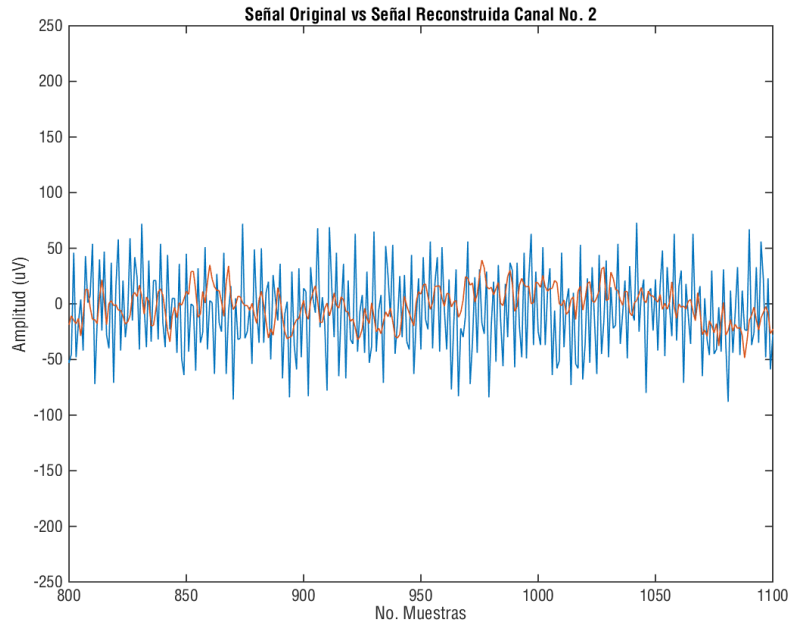


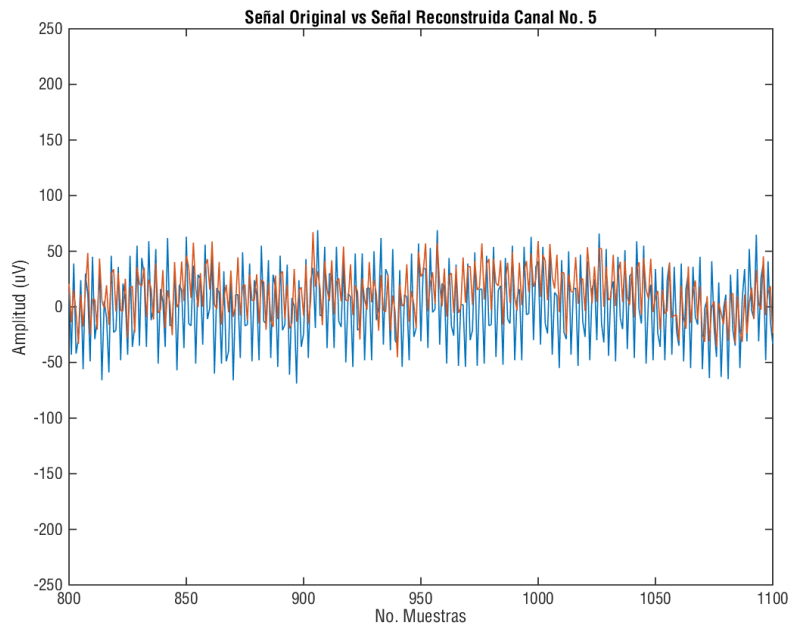
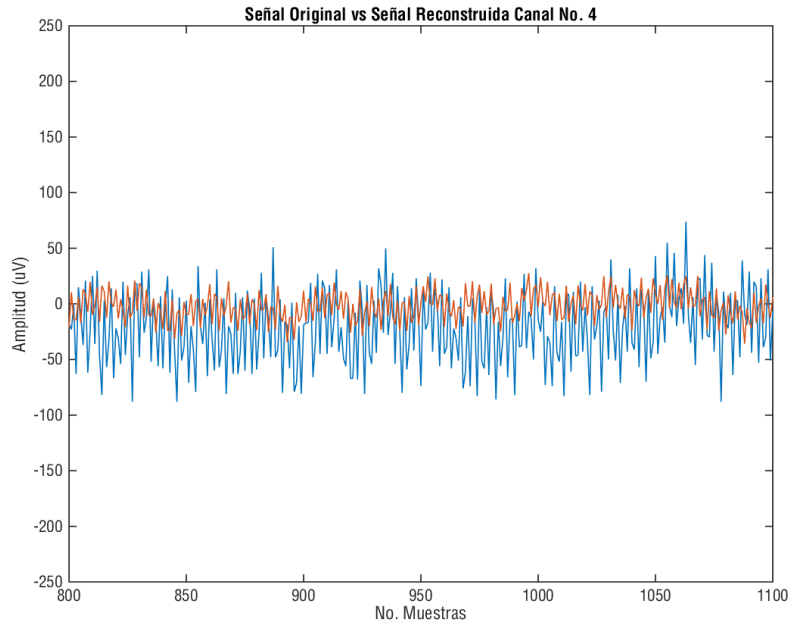
Análisis 6 Canales:

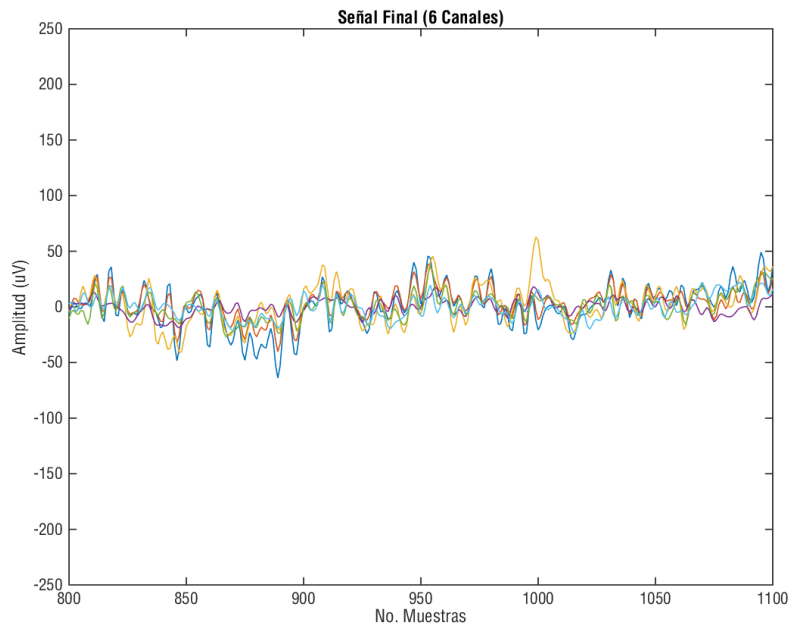
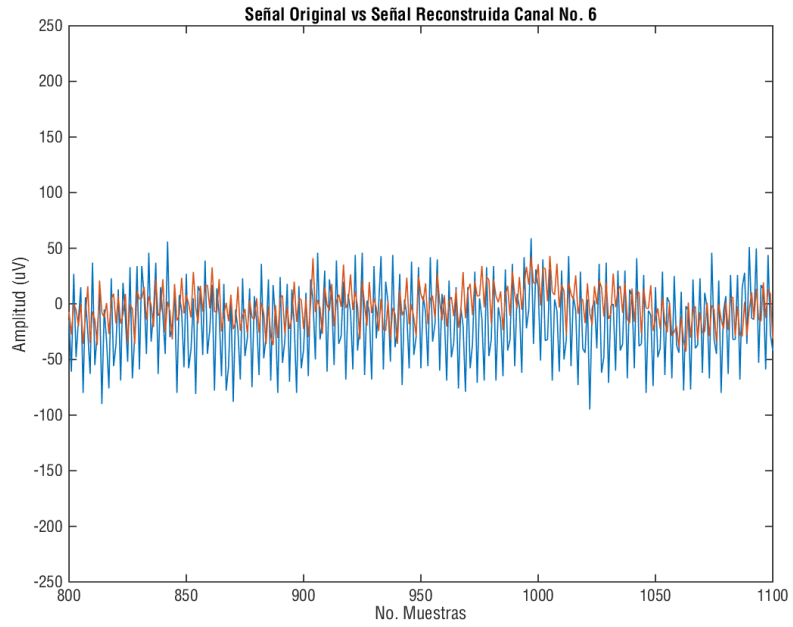






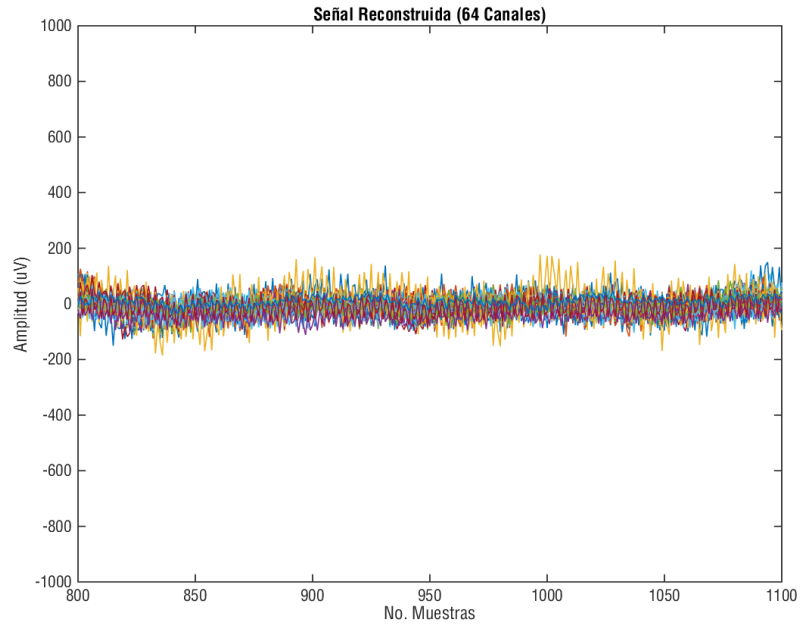
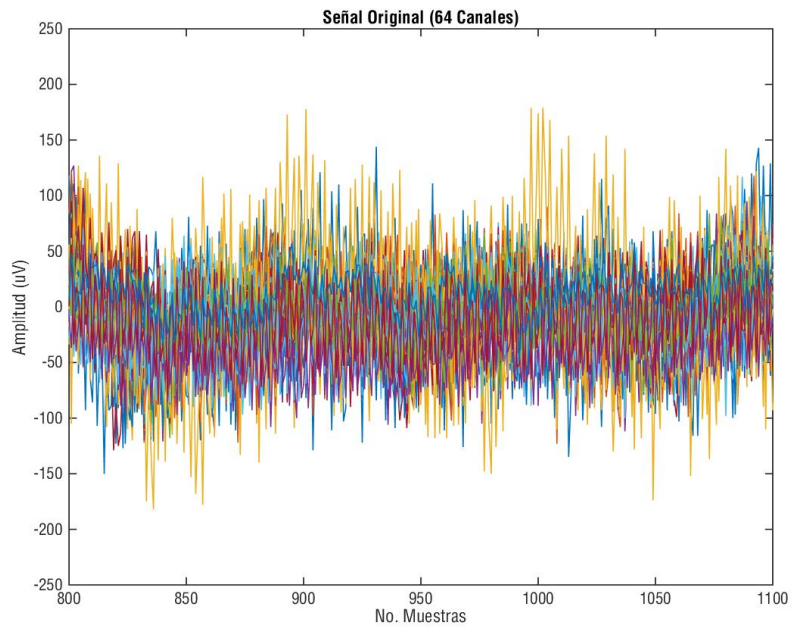


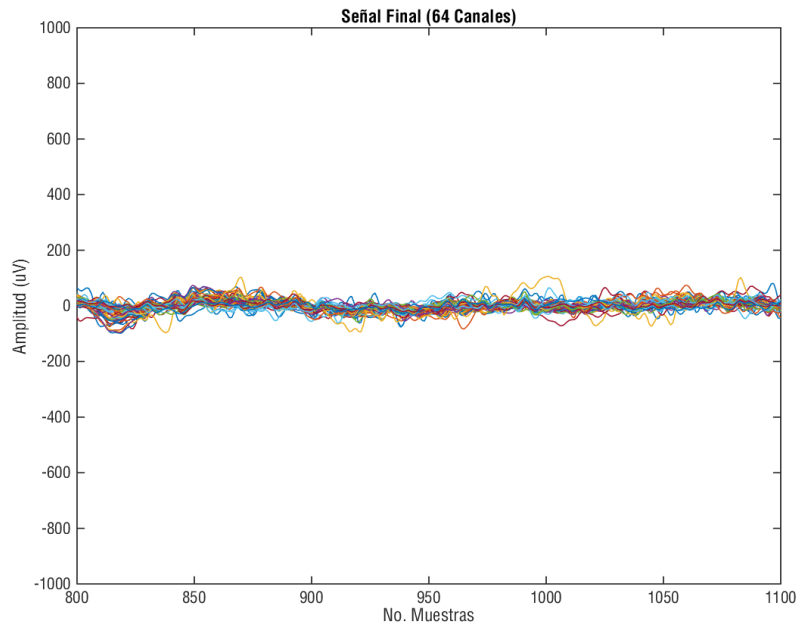




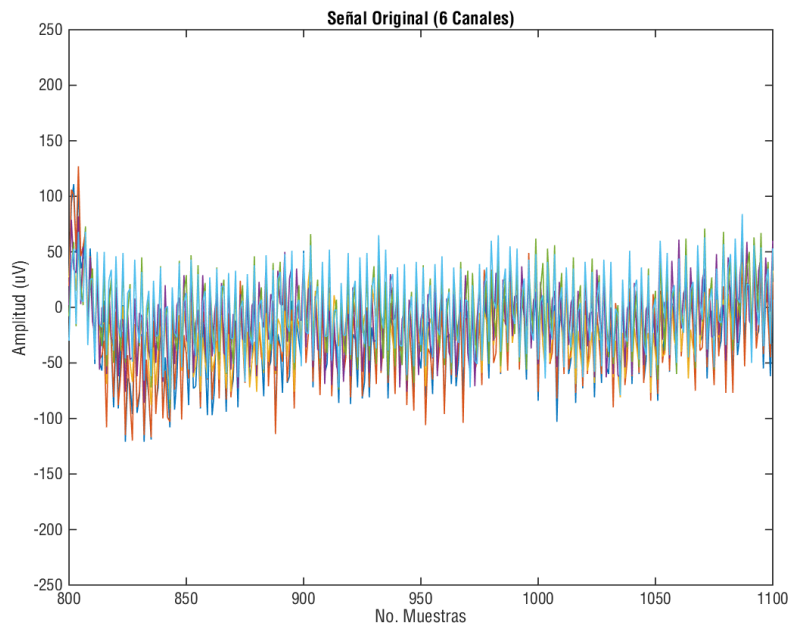
Registro Individuo número 05 tarea 13:

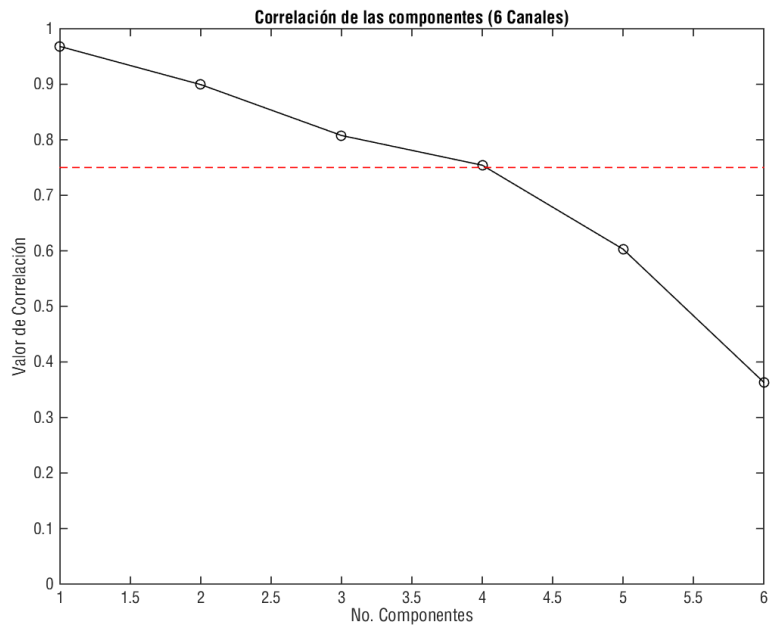
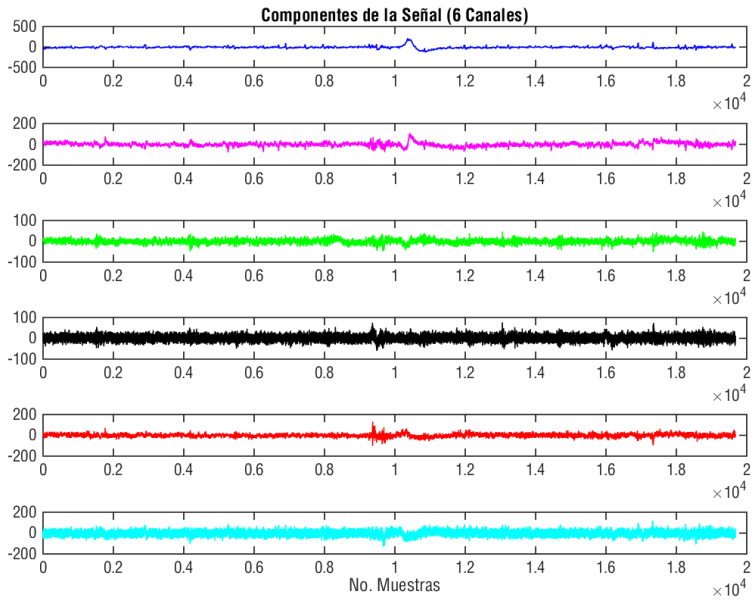
Análisis 64 Canales:

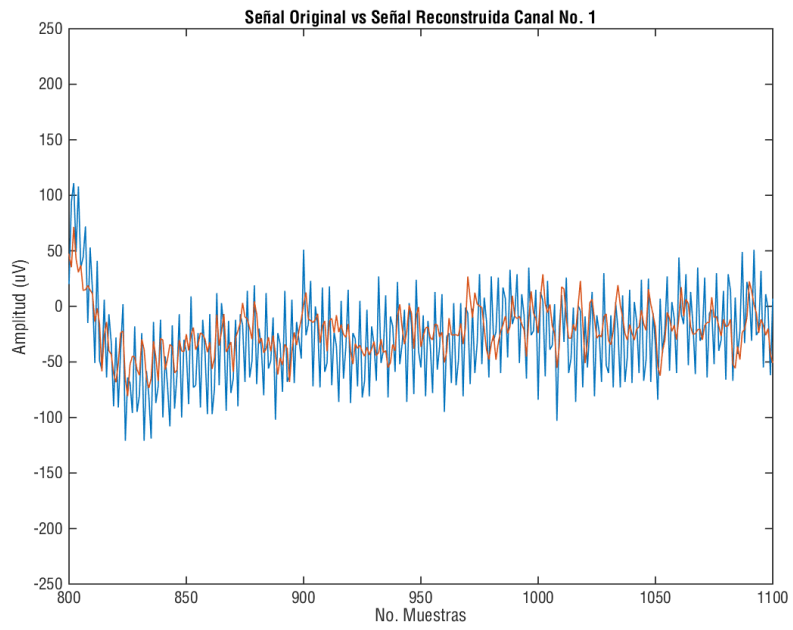
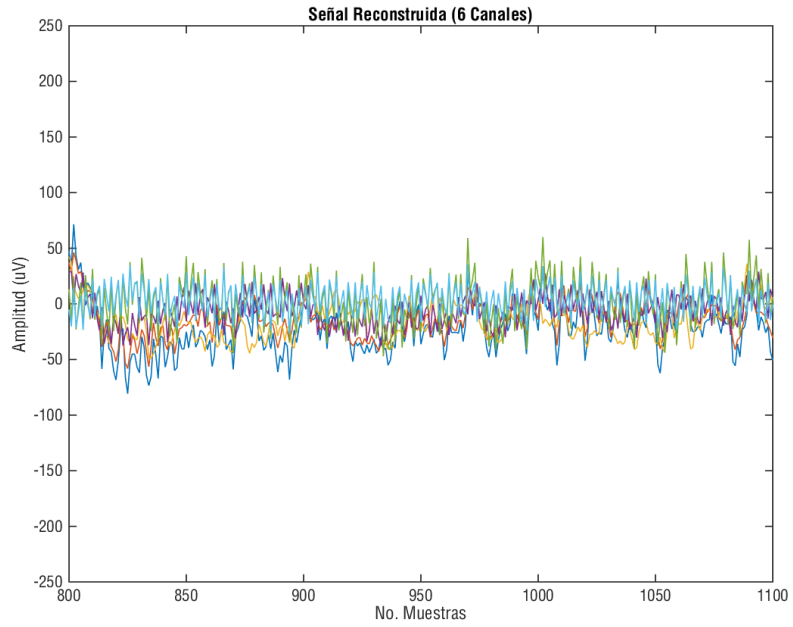


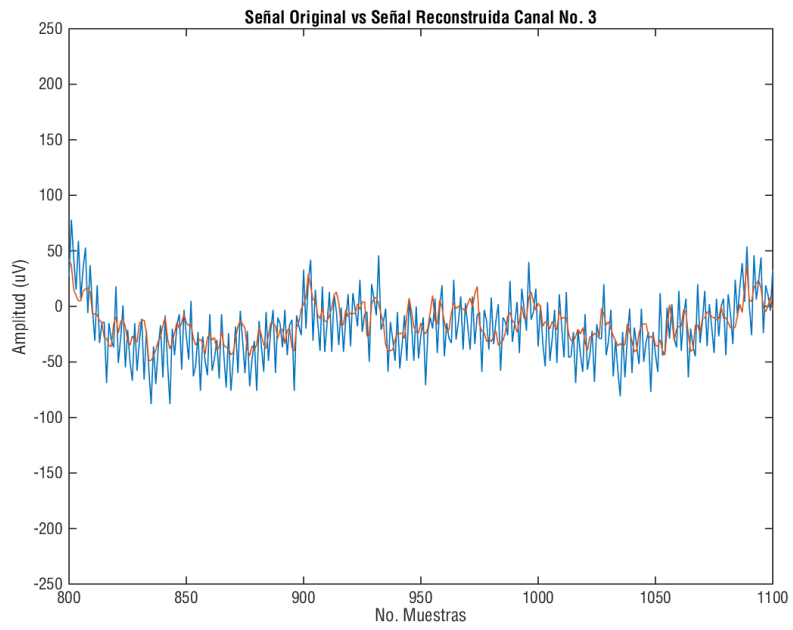
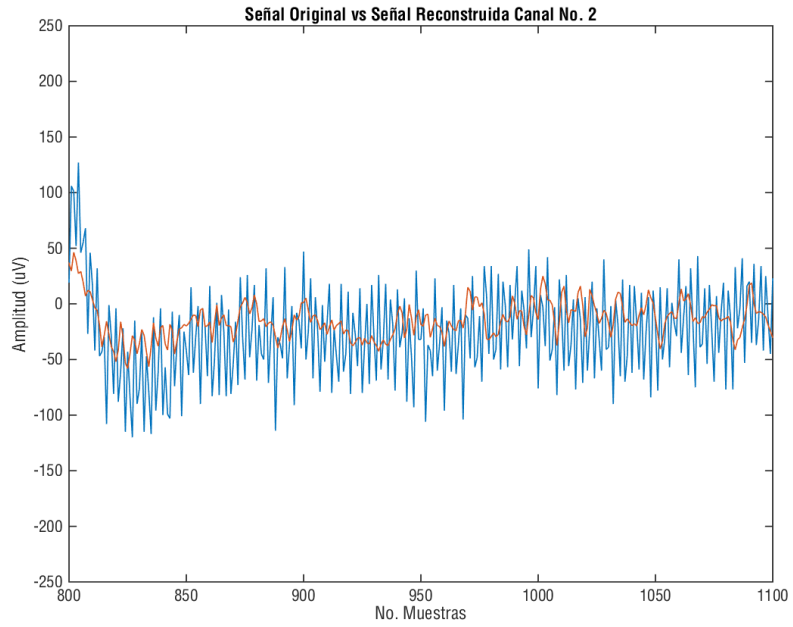


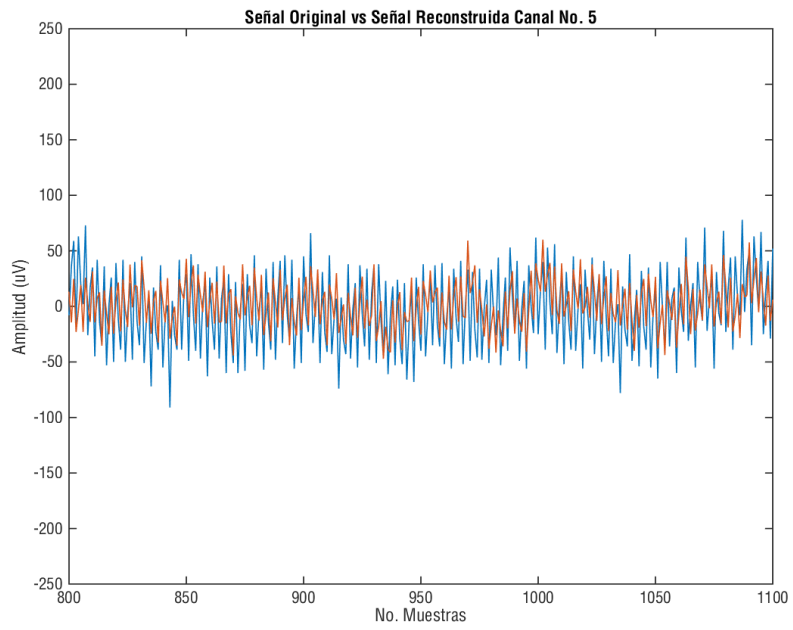
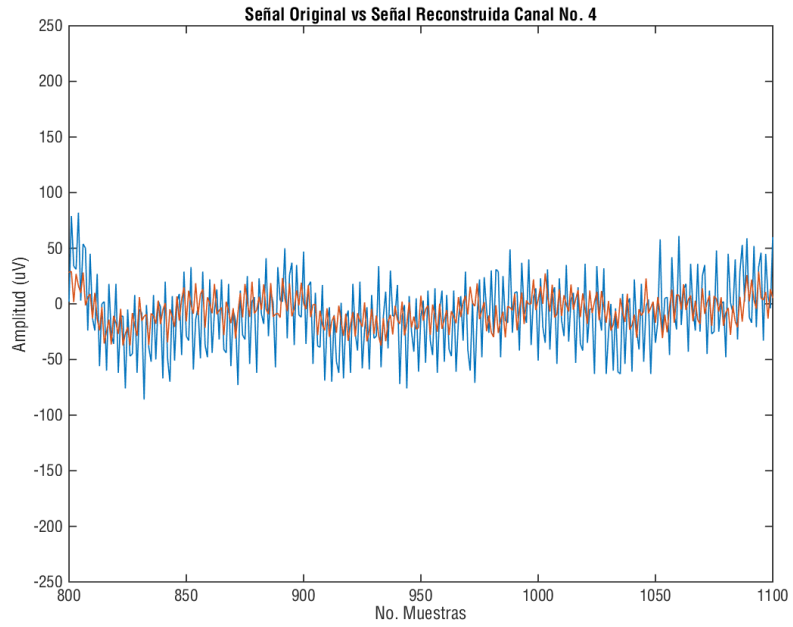
Análisis 6 Canales:

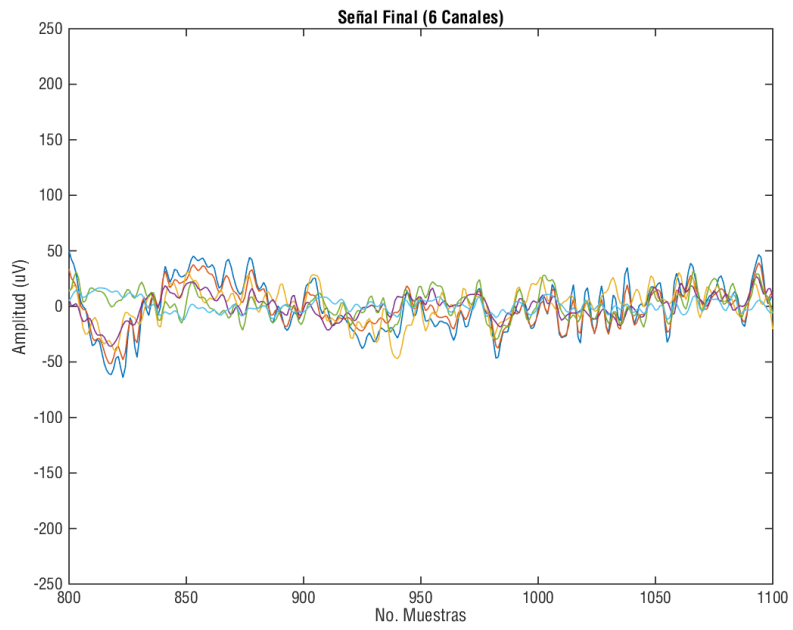
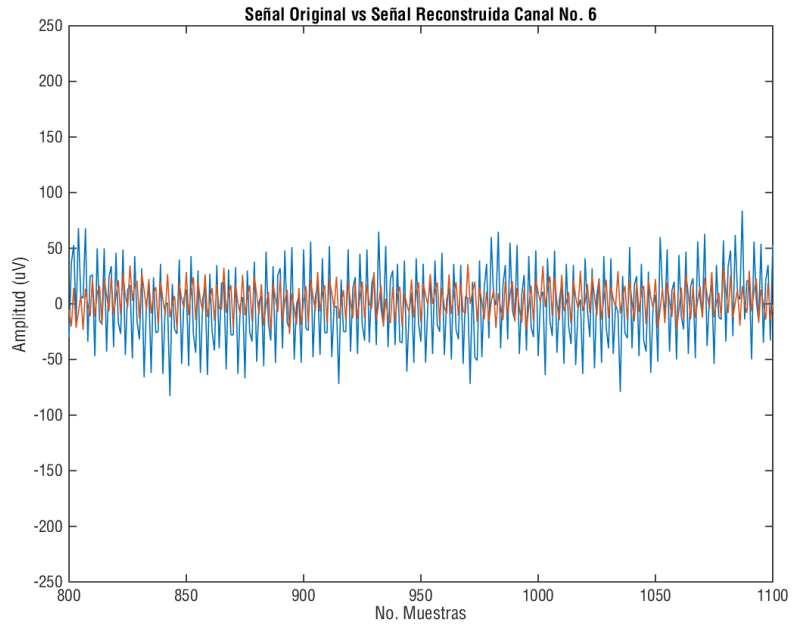






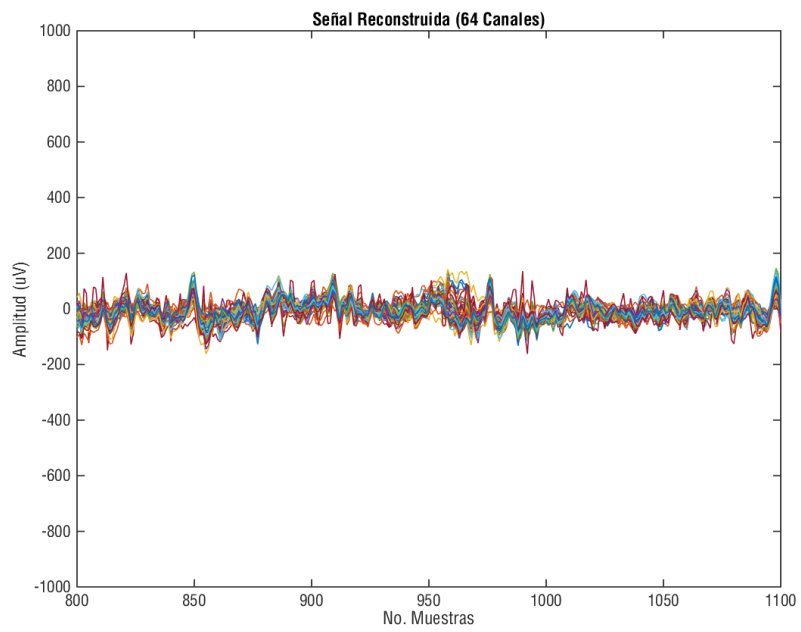
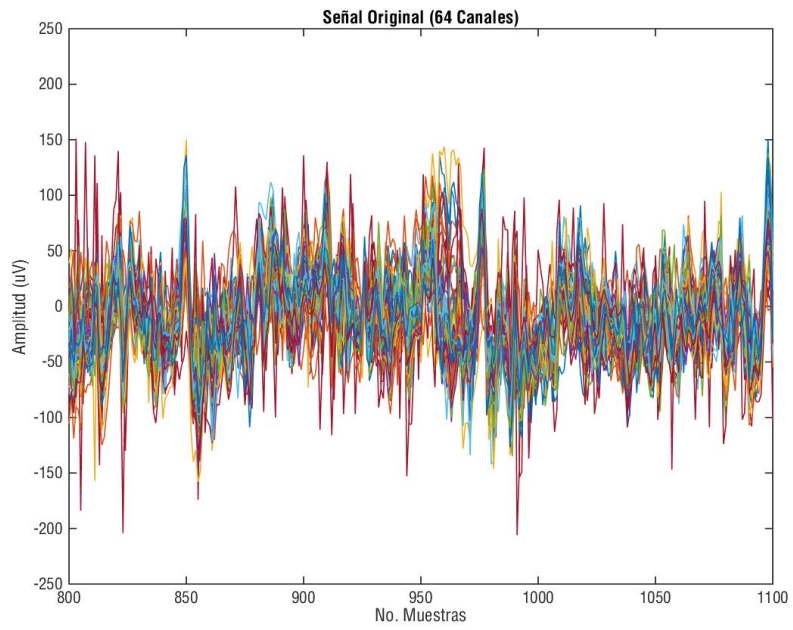


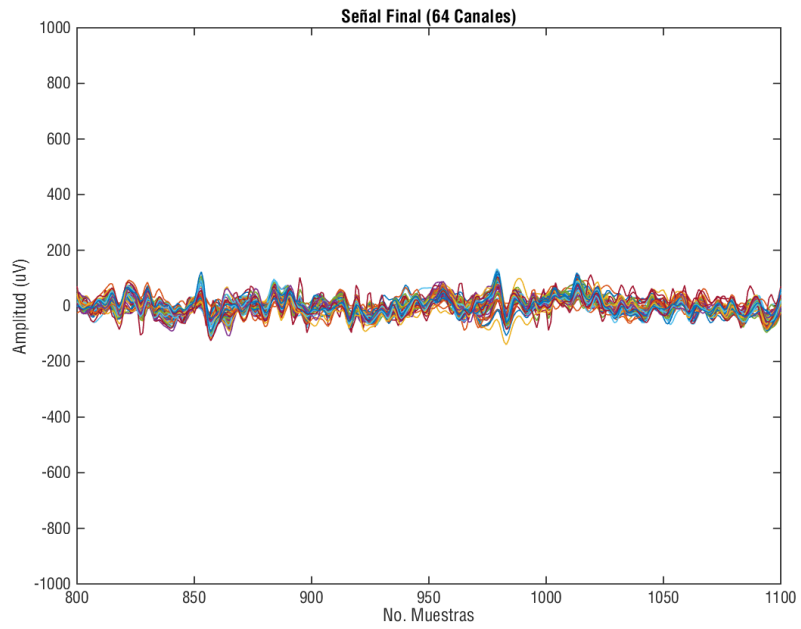




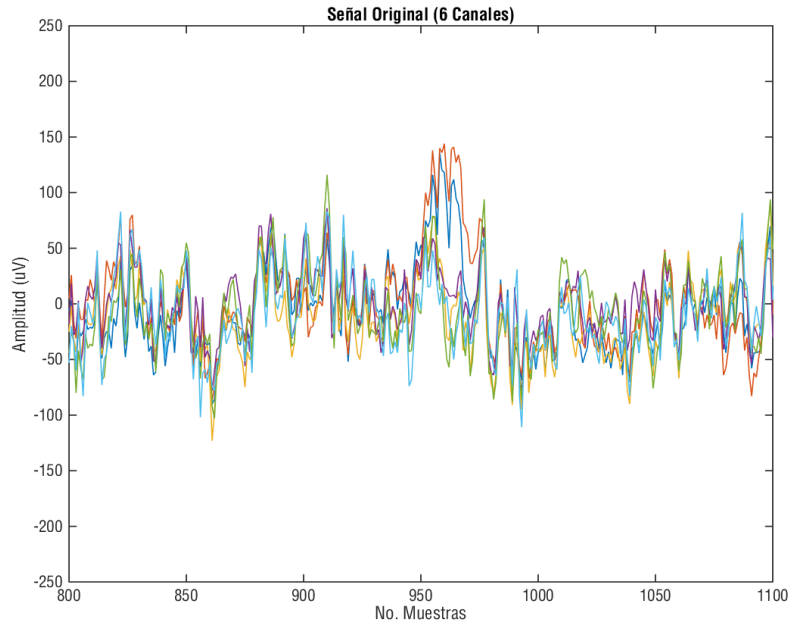
Registro Individuo número 15 tarea 5:

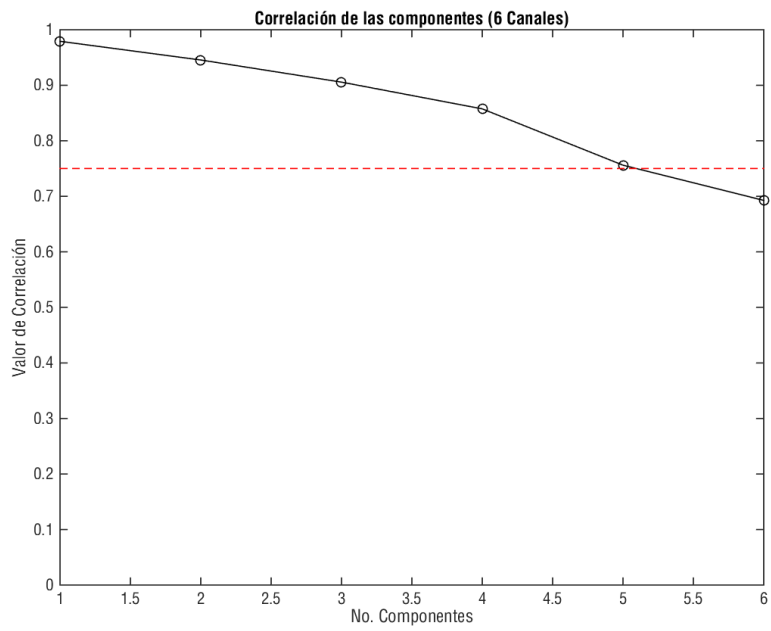
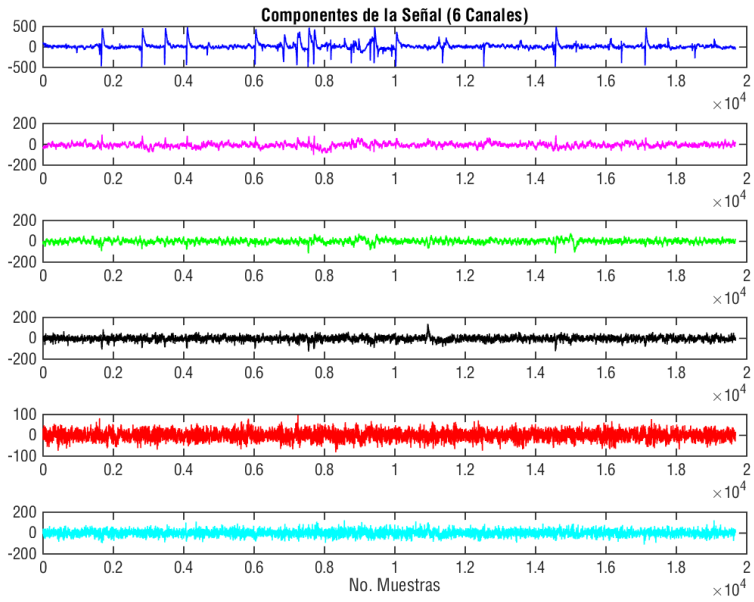
Análisis 64 Canales:

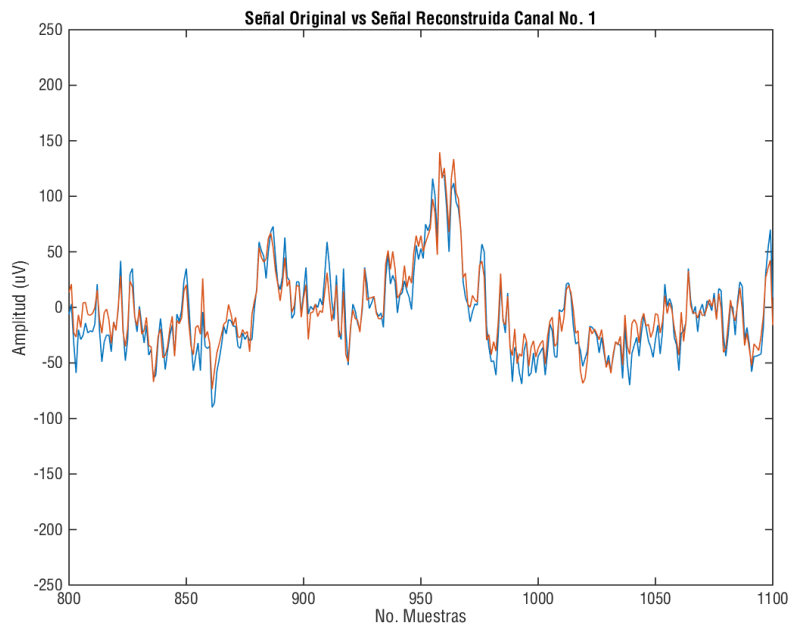
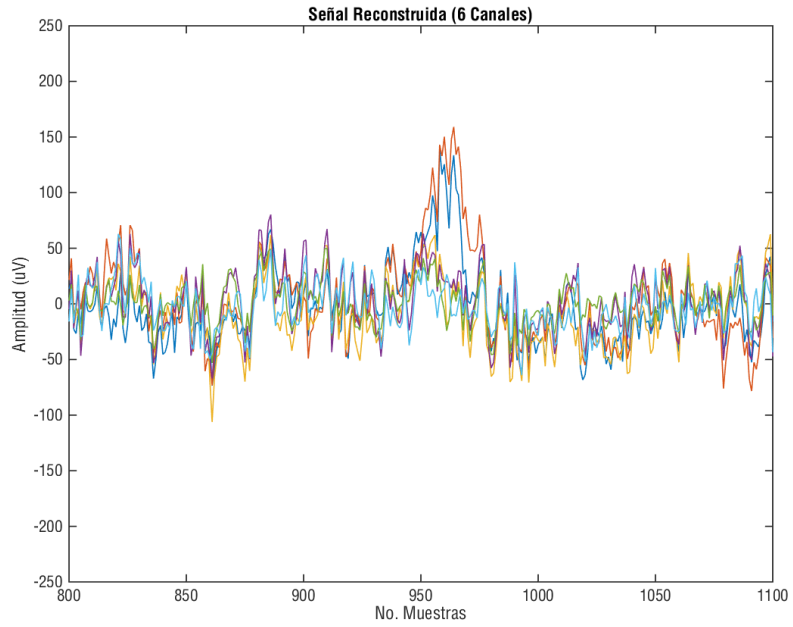


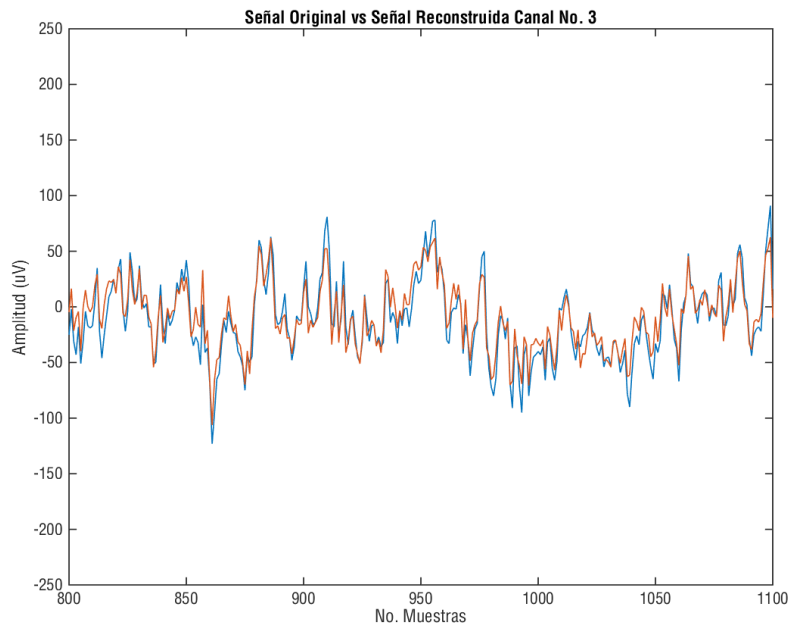
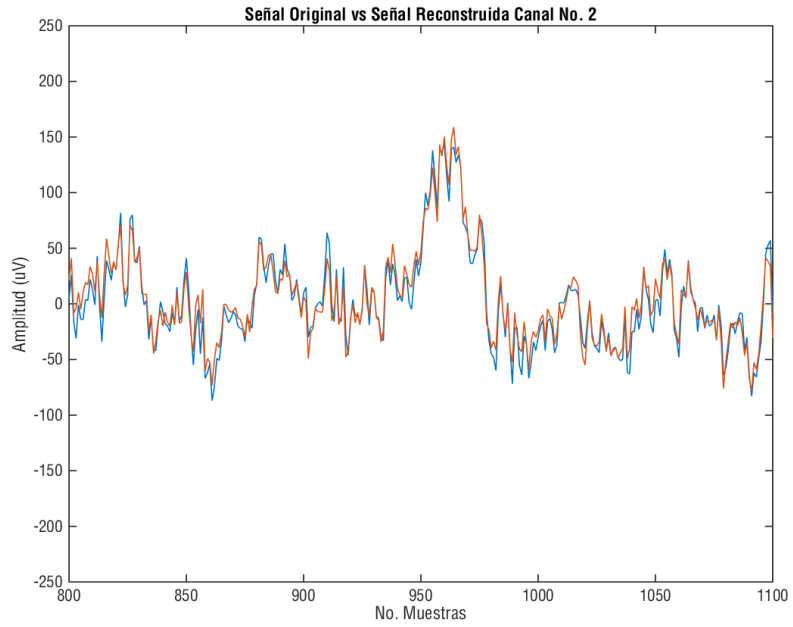


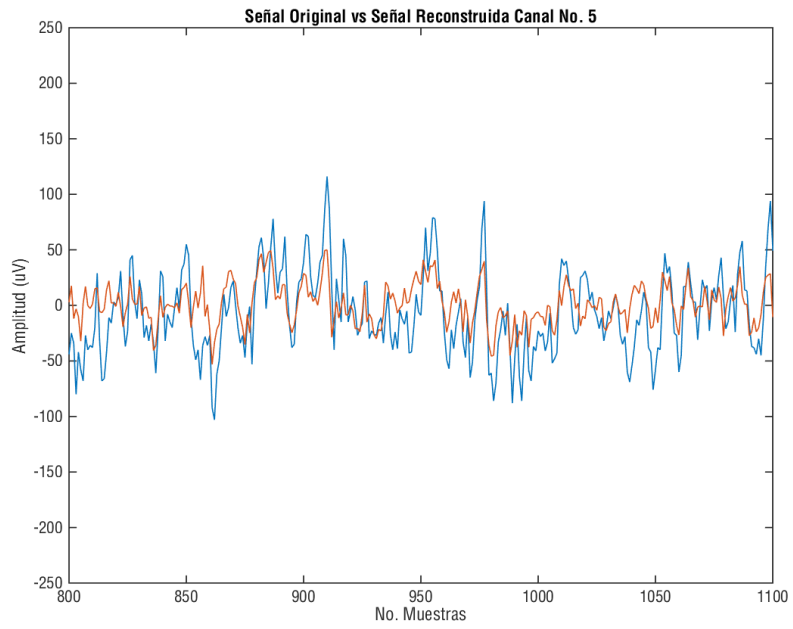
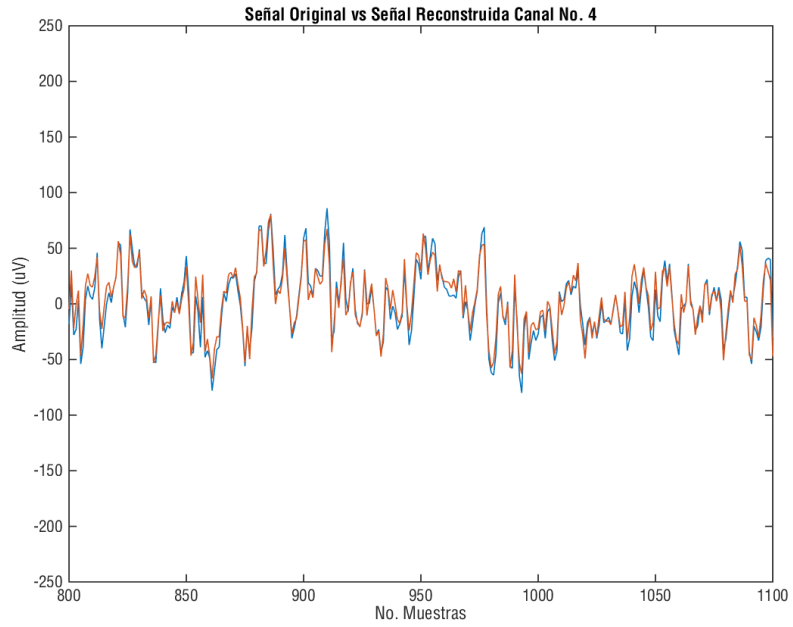
Análisis 6 Canales:

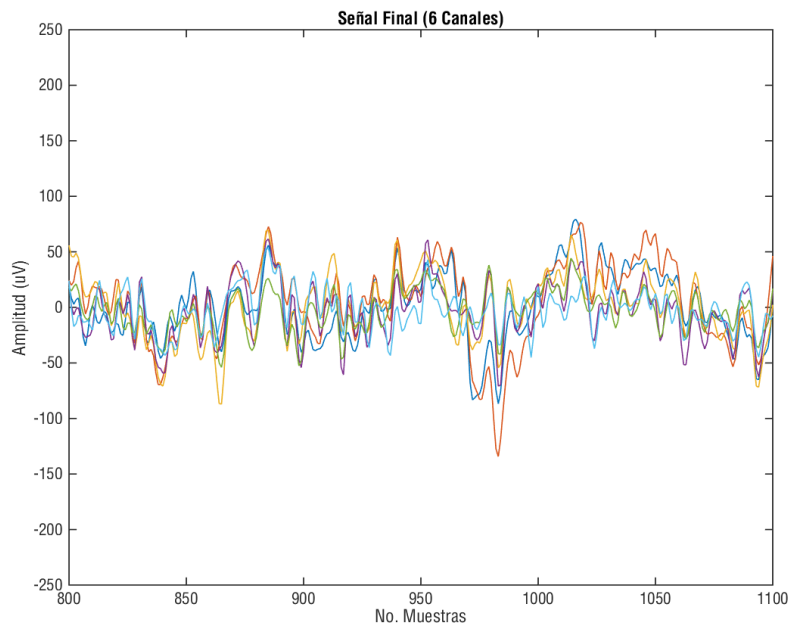
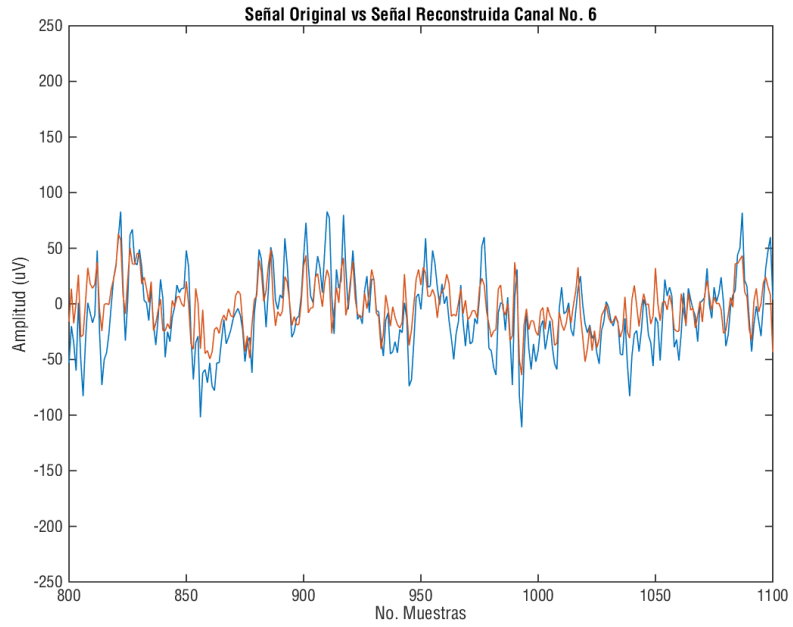






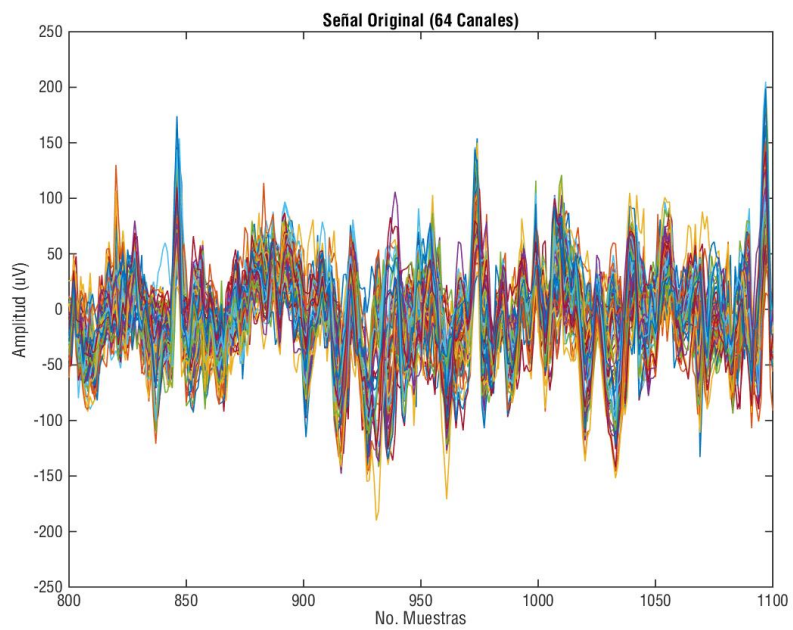
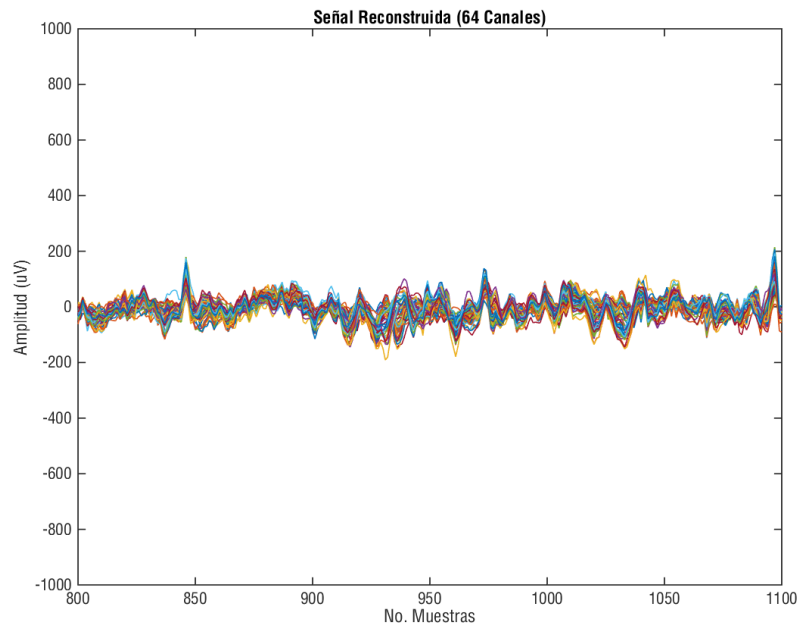


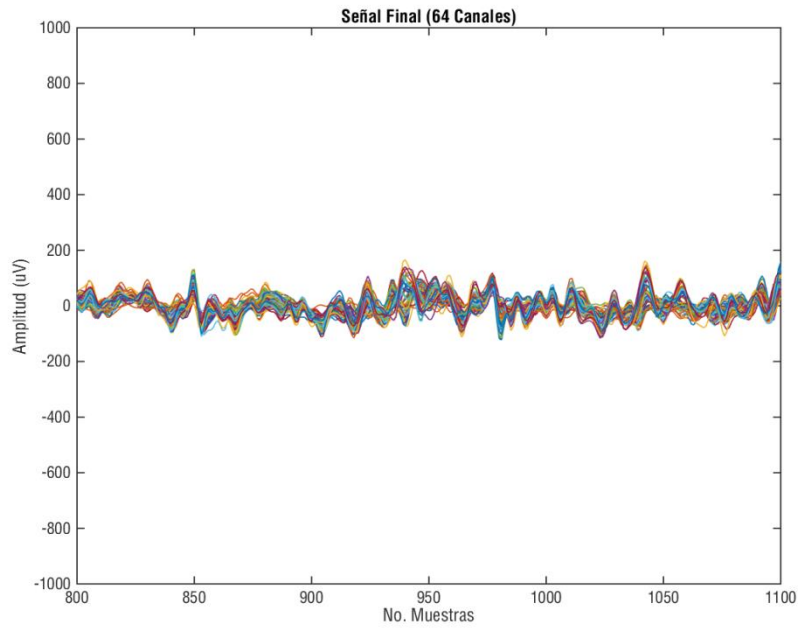




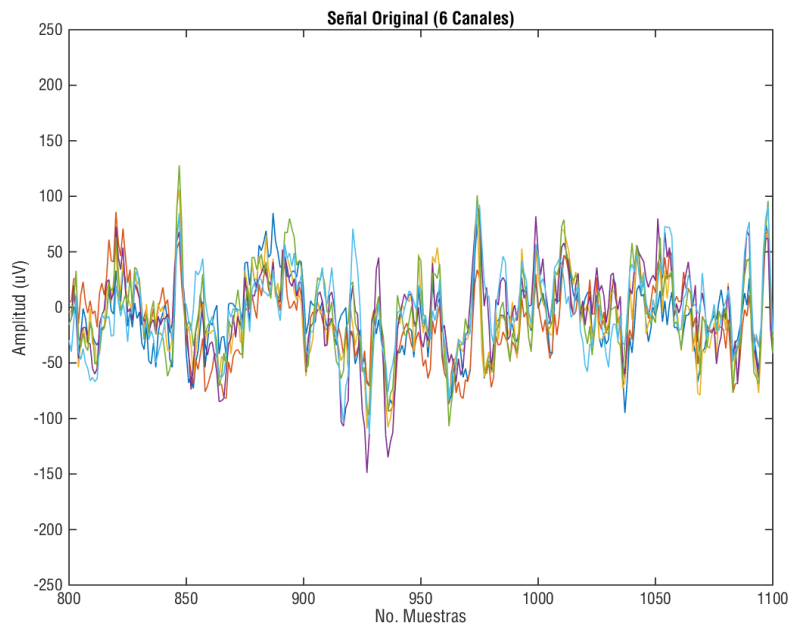
Registro Individuo número 15 tarea 9:

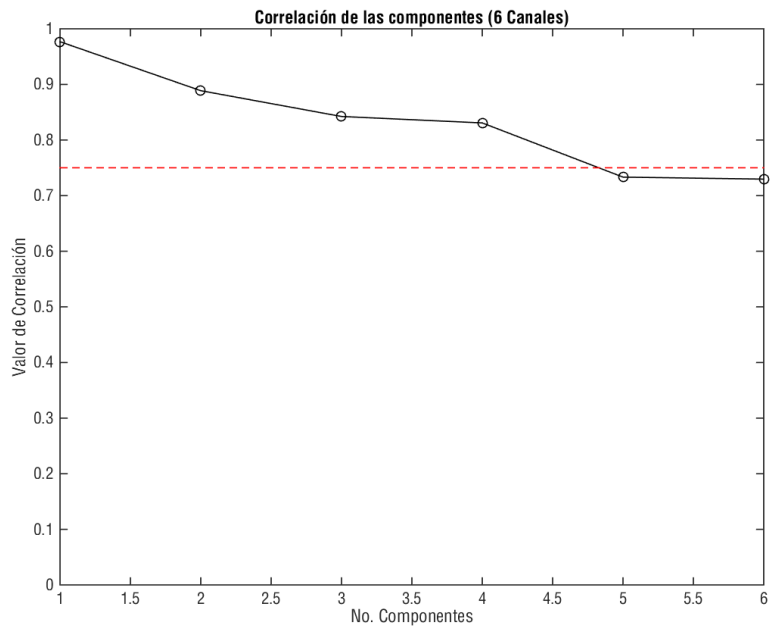
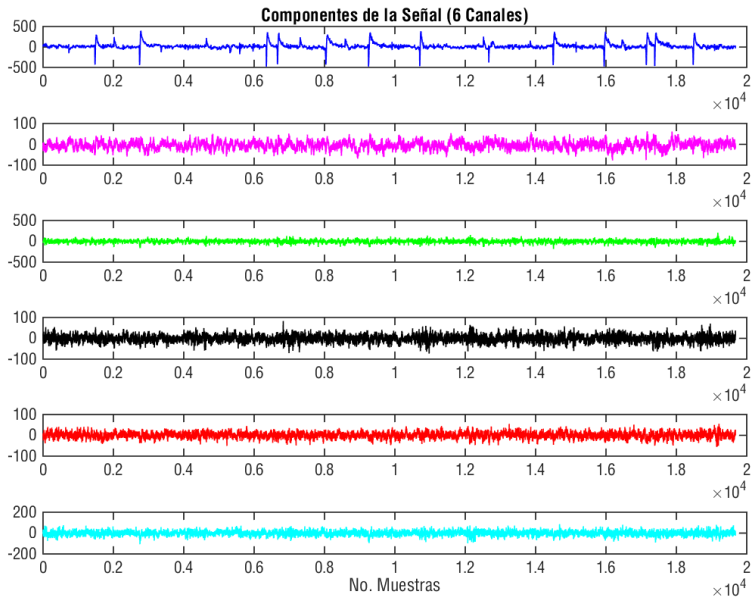
Análisis 64 Canales:

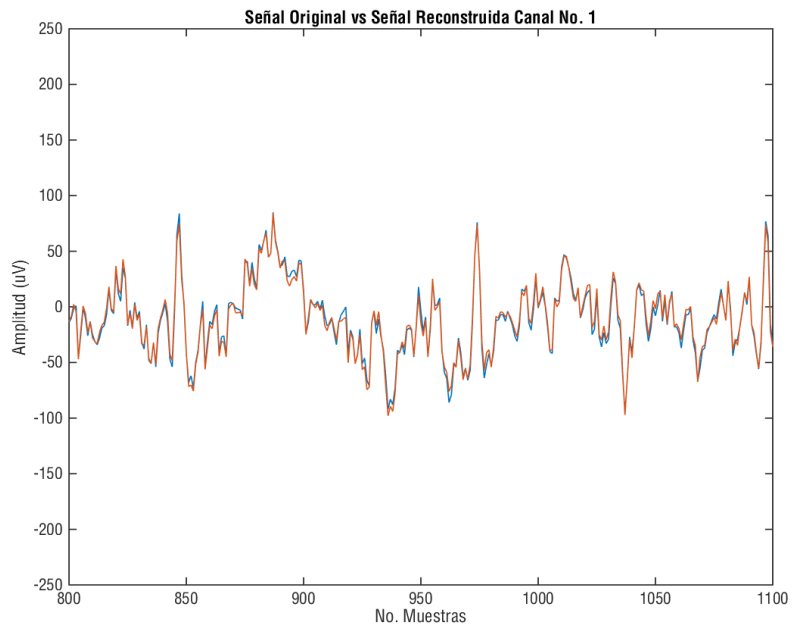
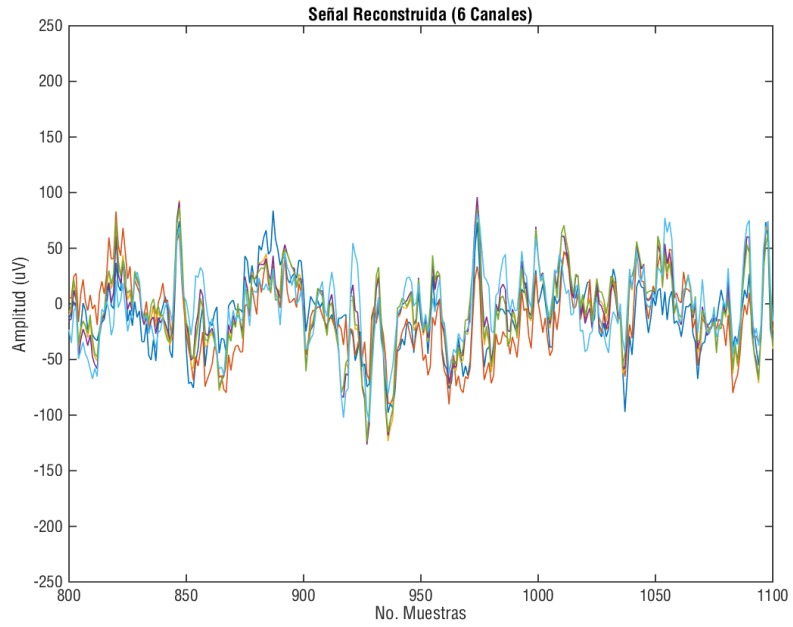


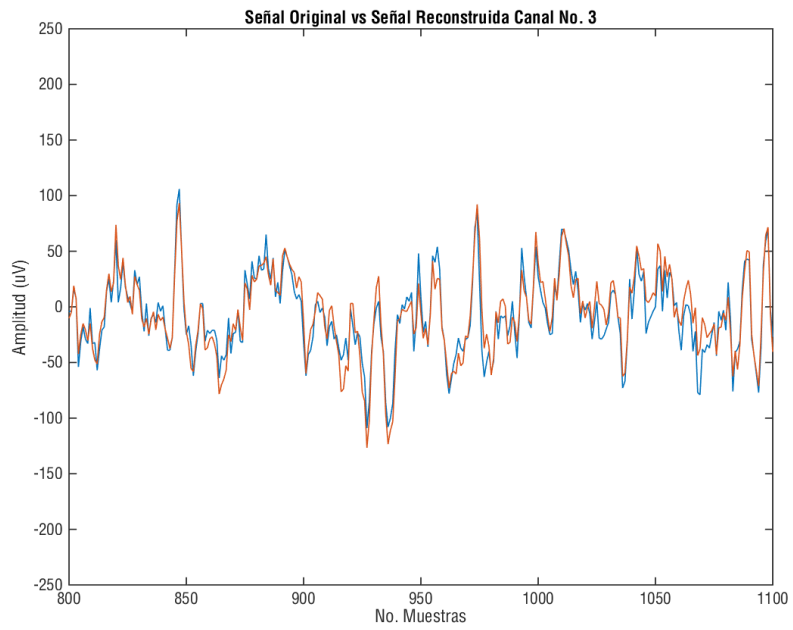
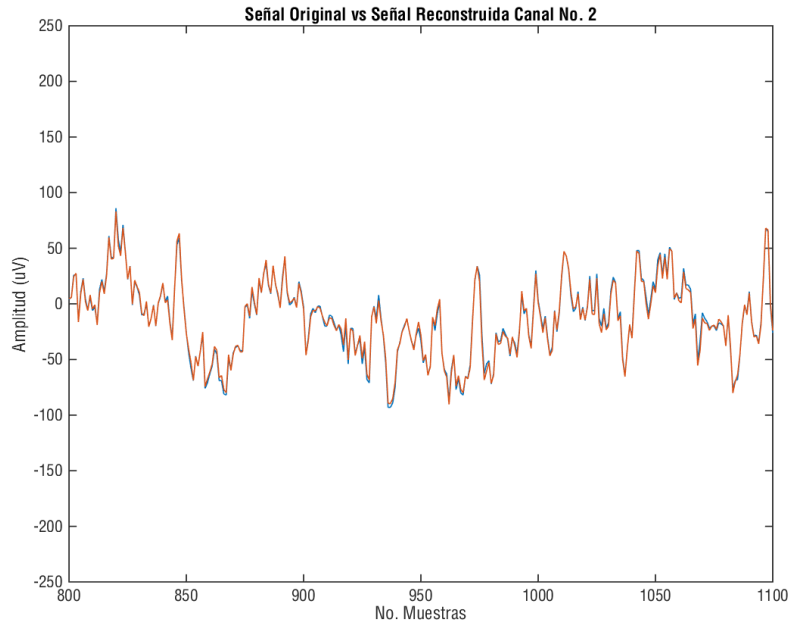


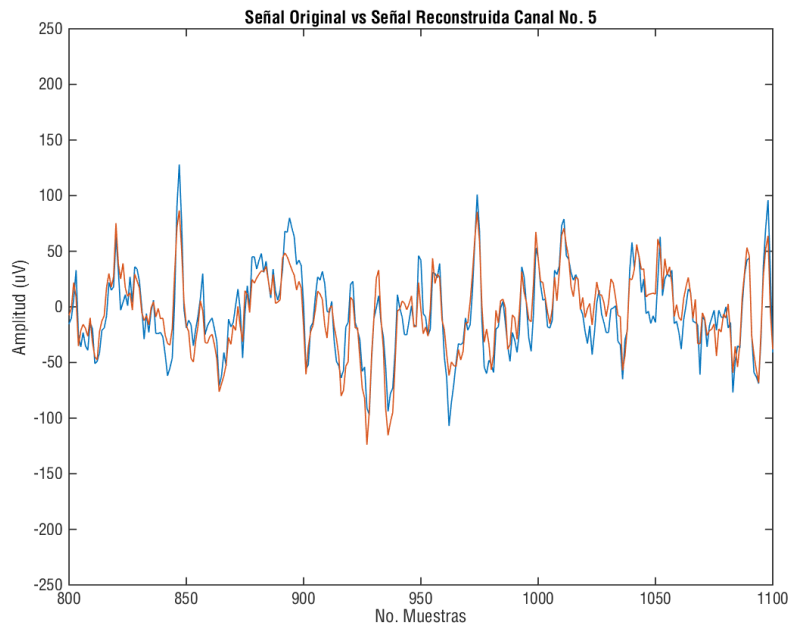
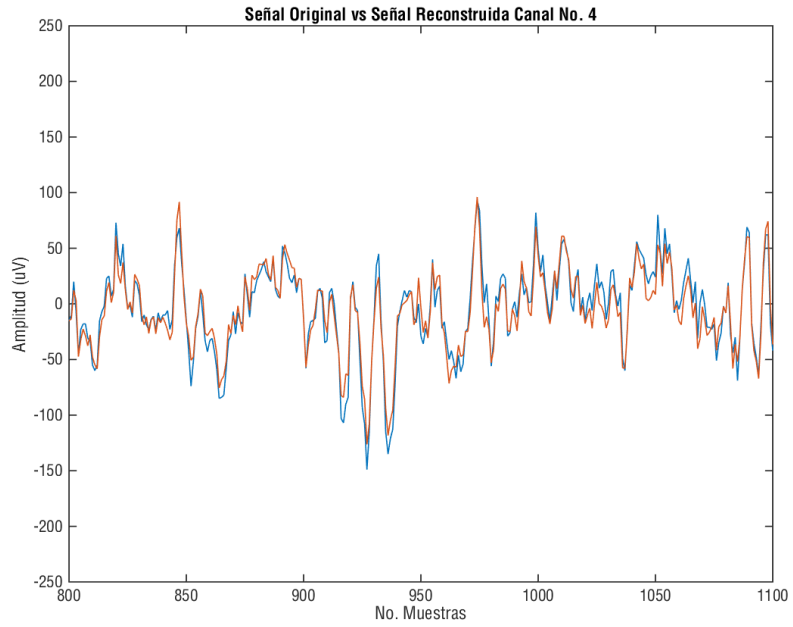
Análisis 6 Canales:

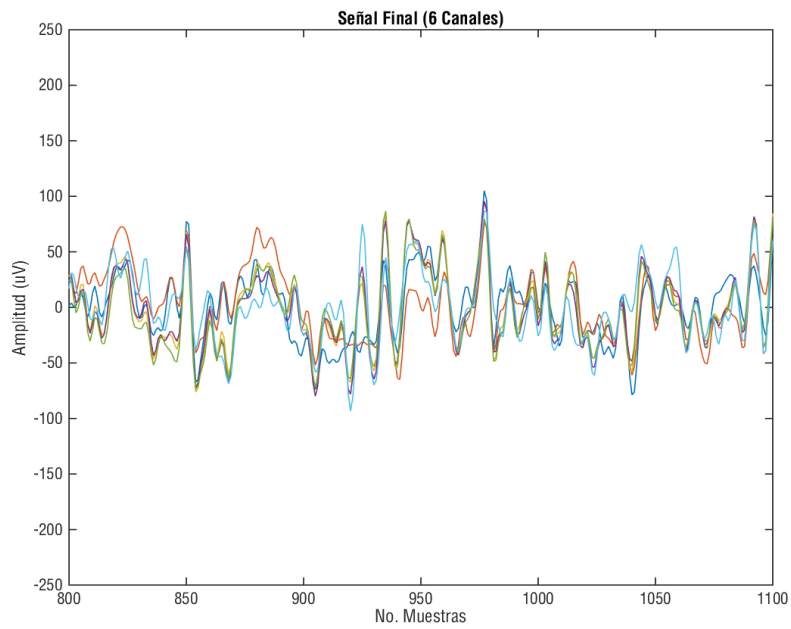
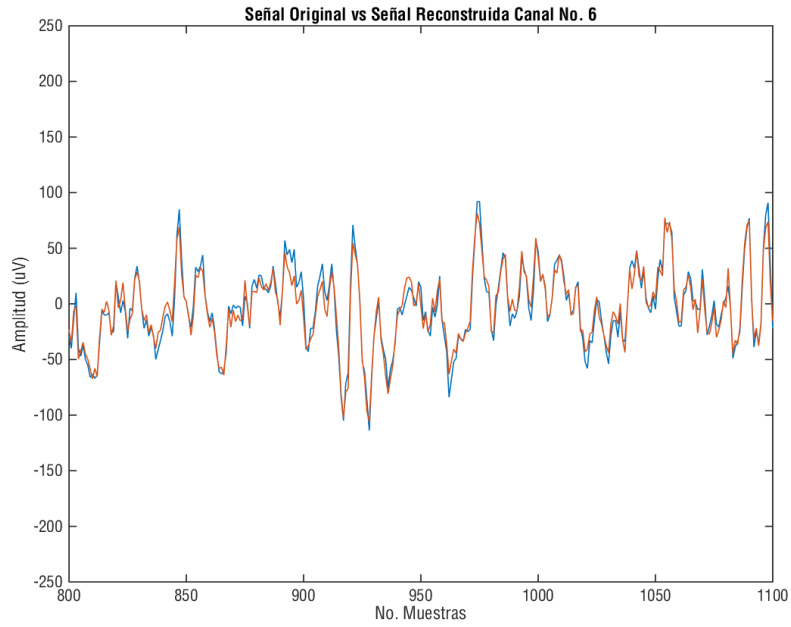






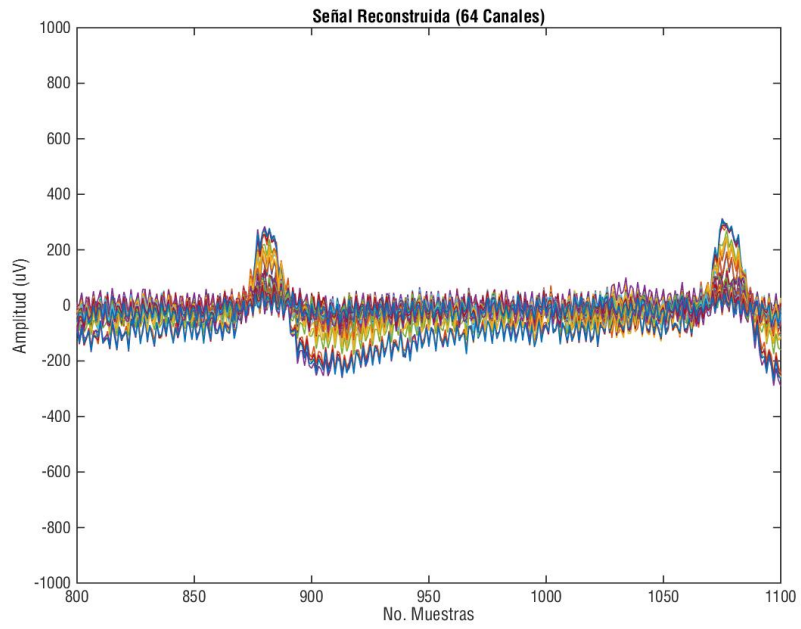
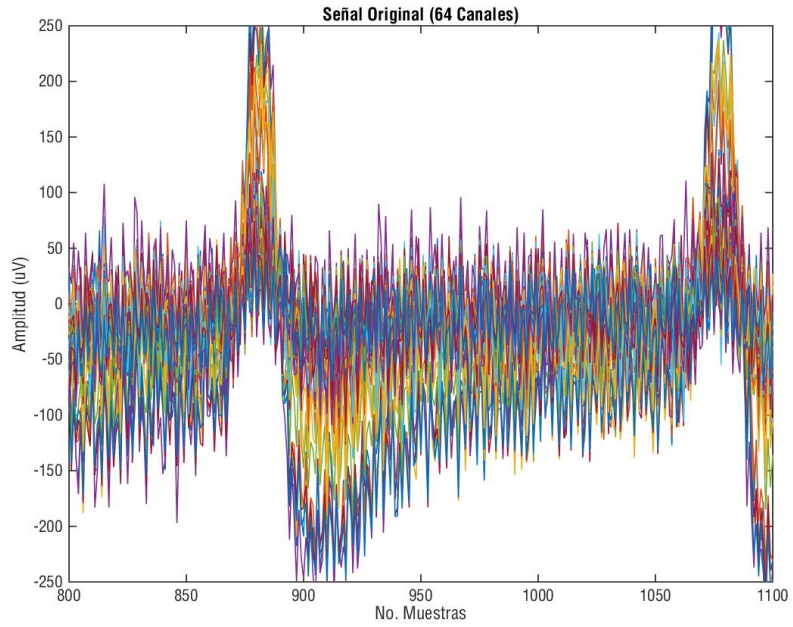


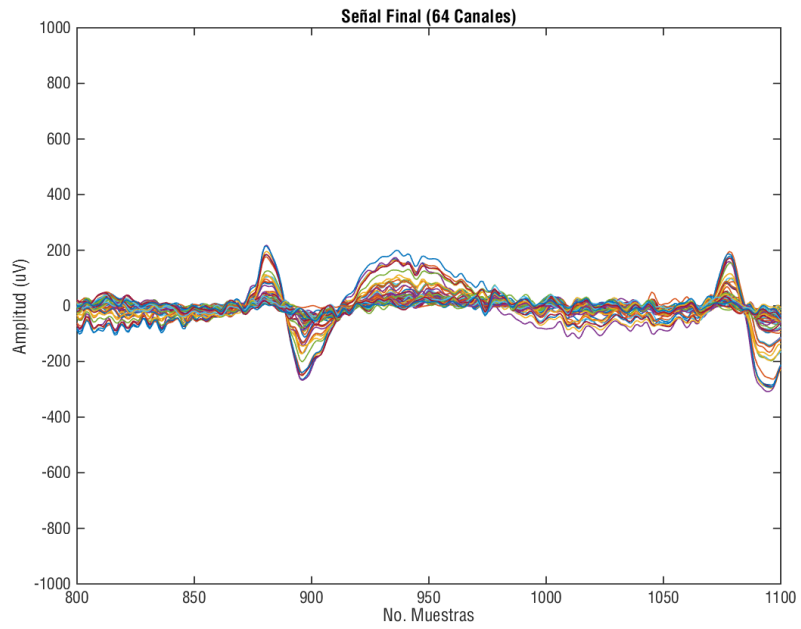




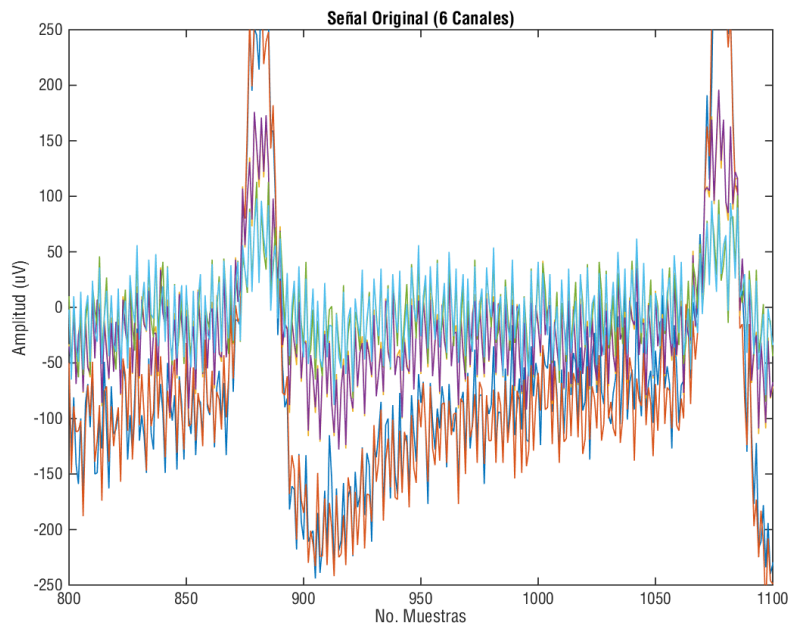
Registro Individuo número 15 tarea 13:

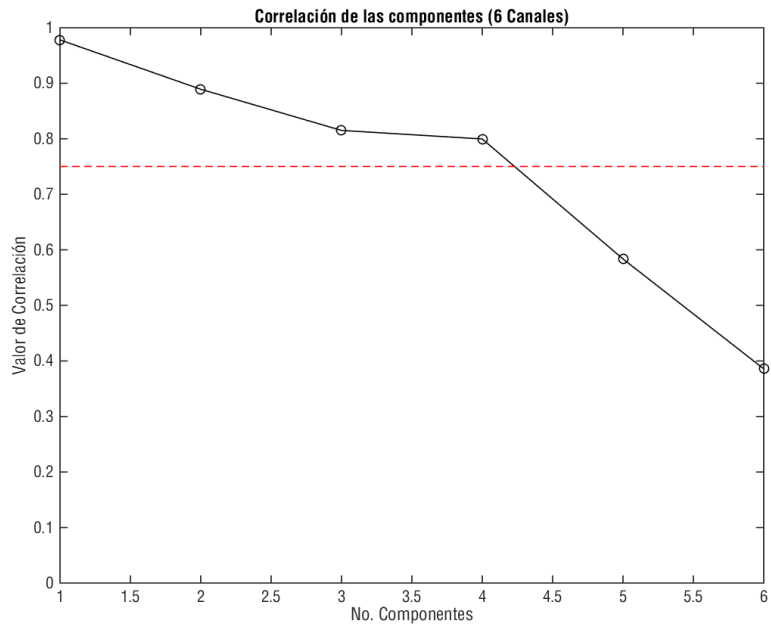
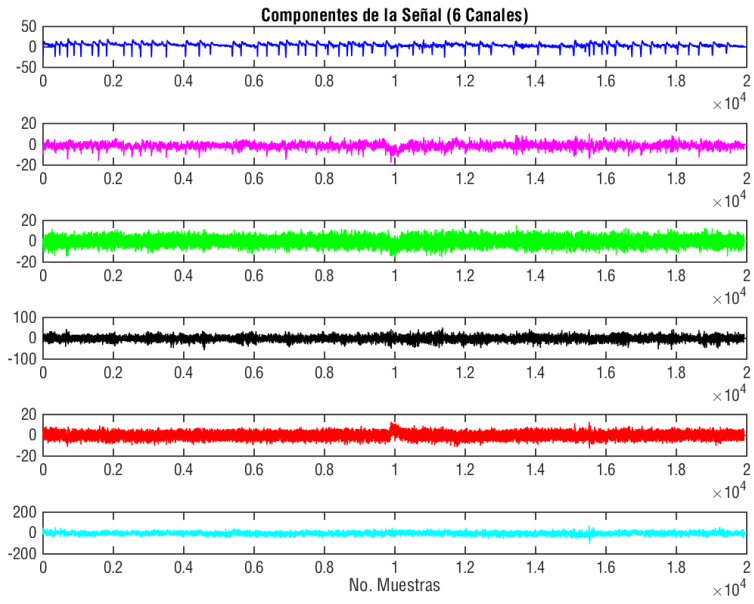
Análisis 64 Canales:

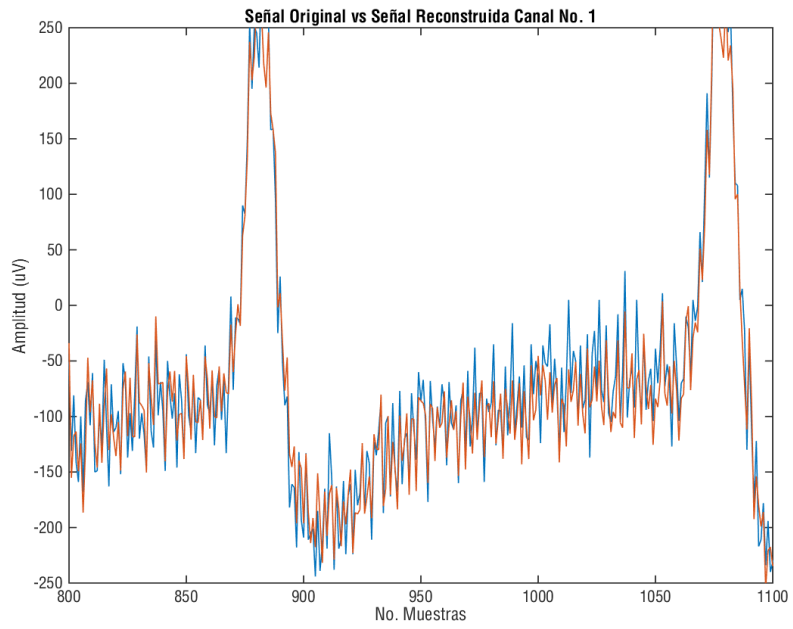
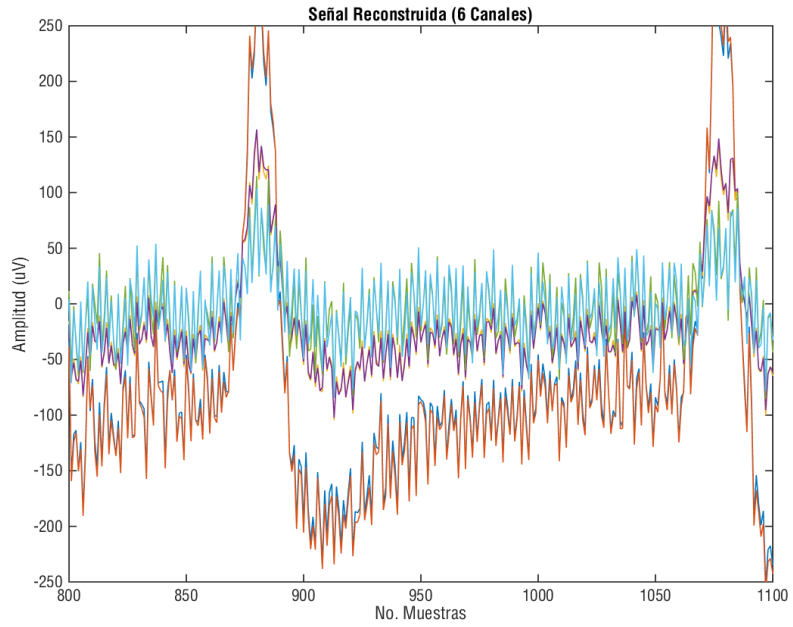


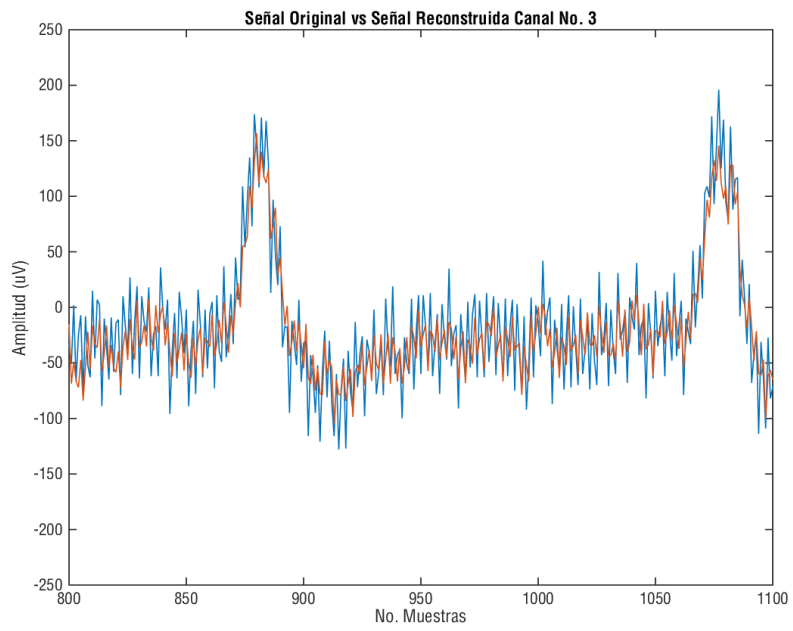
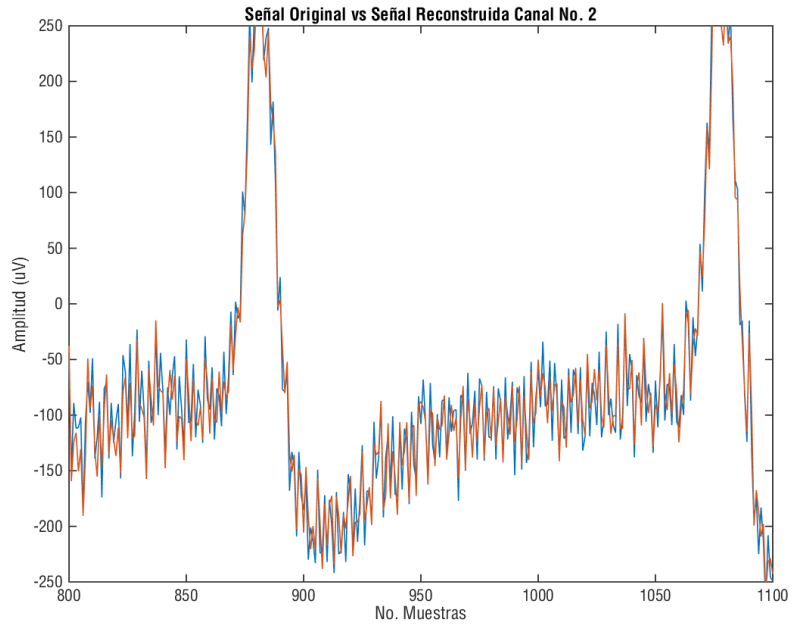


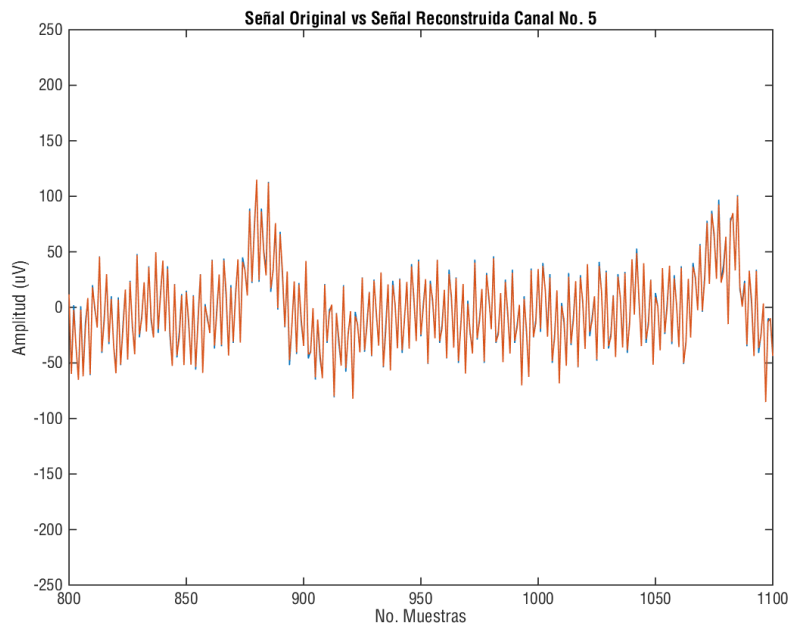
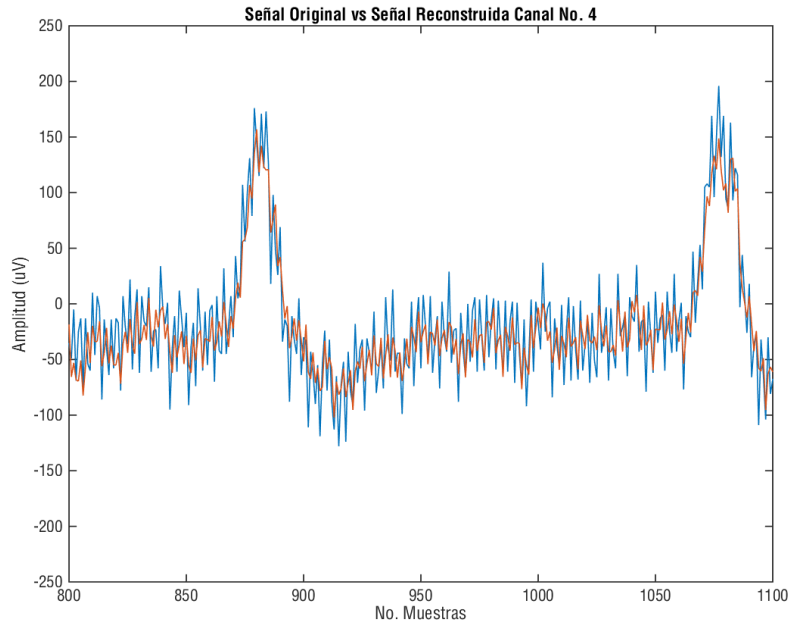
Análisis 6 Canales:

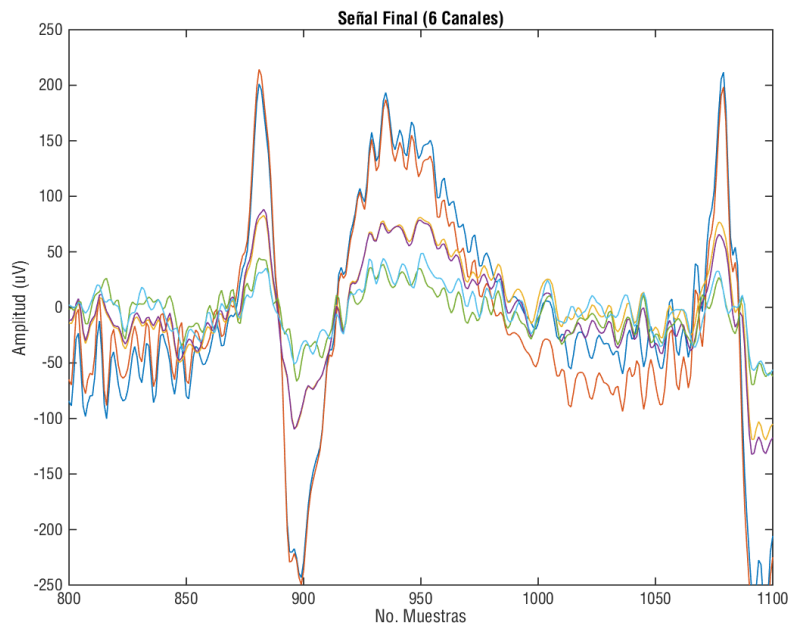
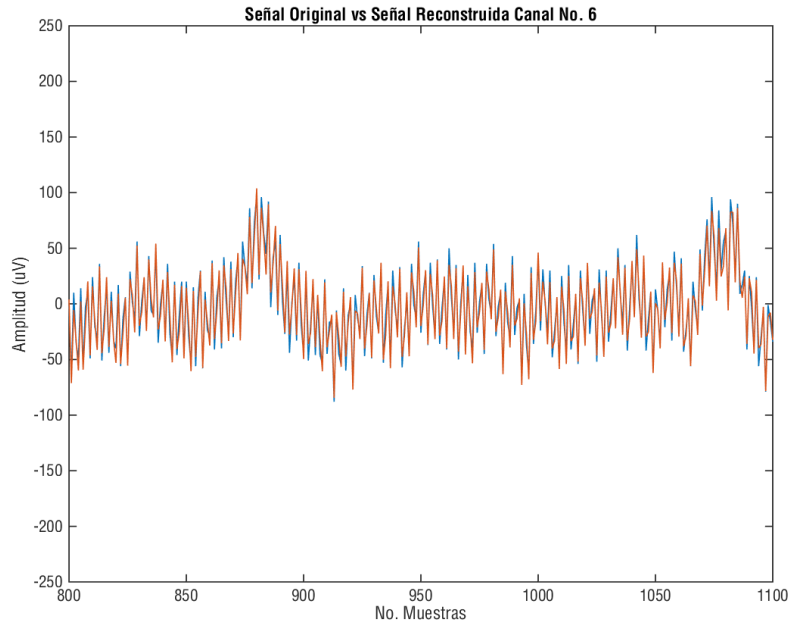






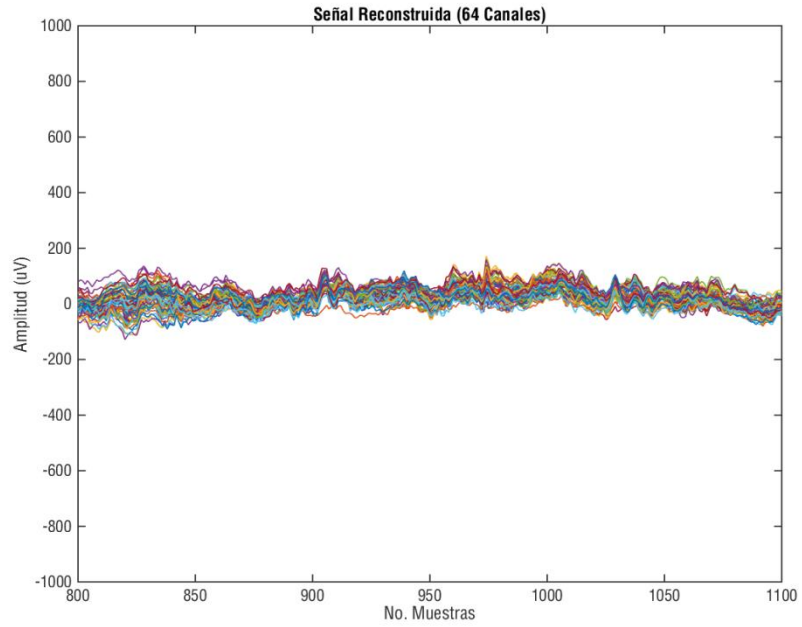
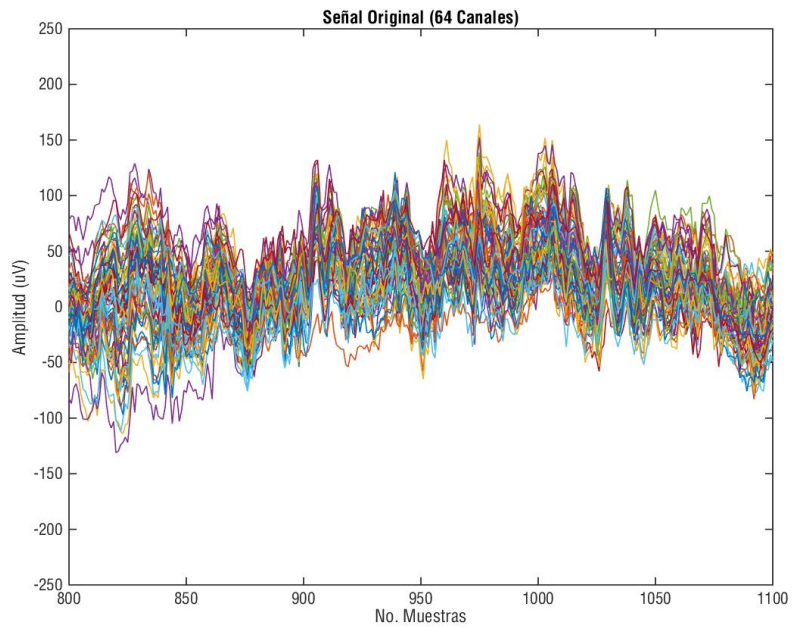


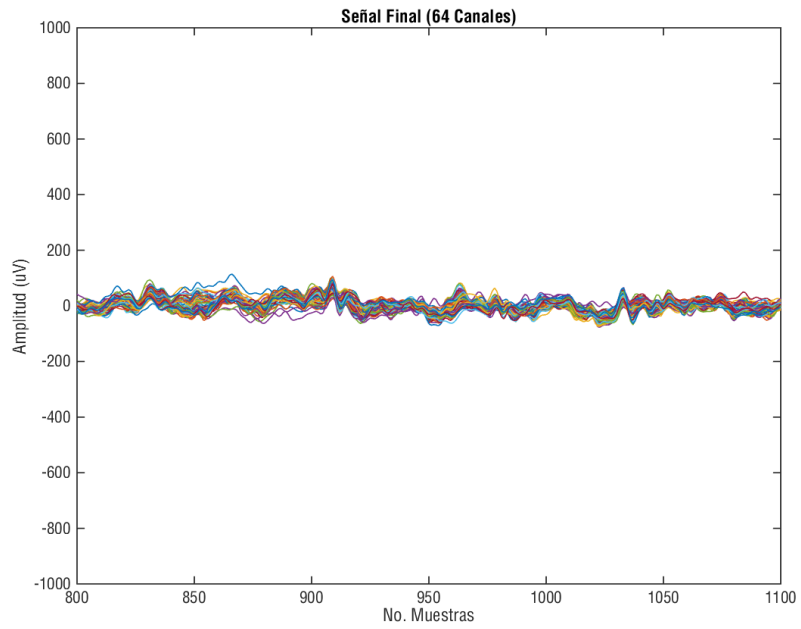




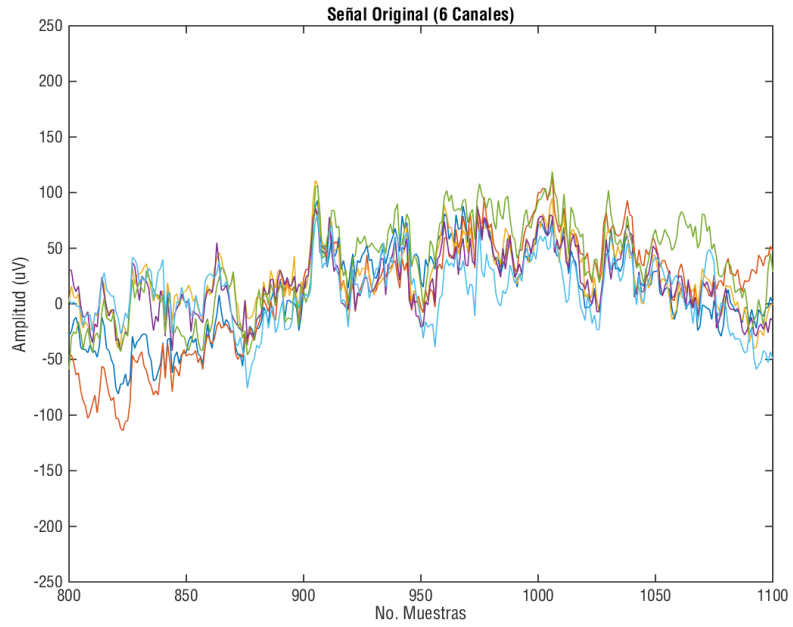
Registro Individuo número 57 tarea 5:

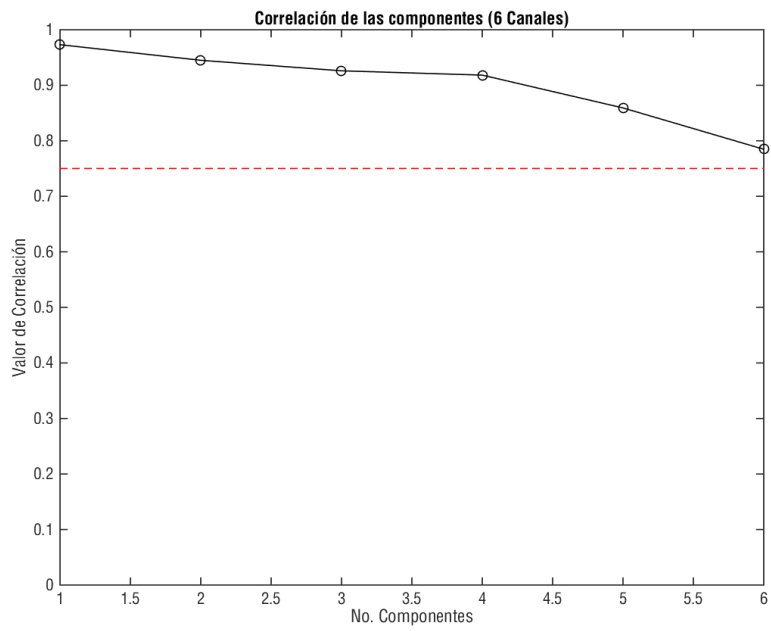
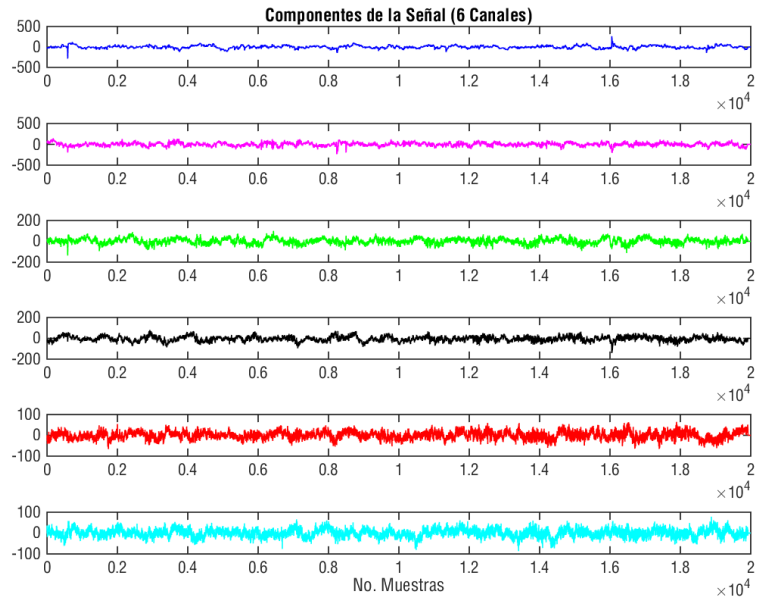
Análisis 64 Canales:

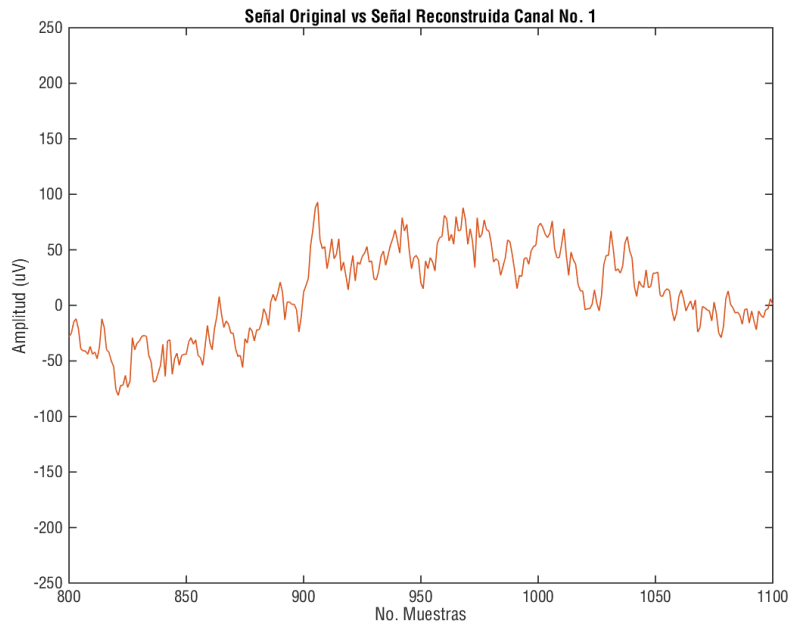
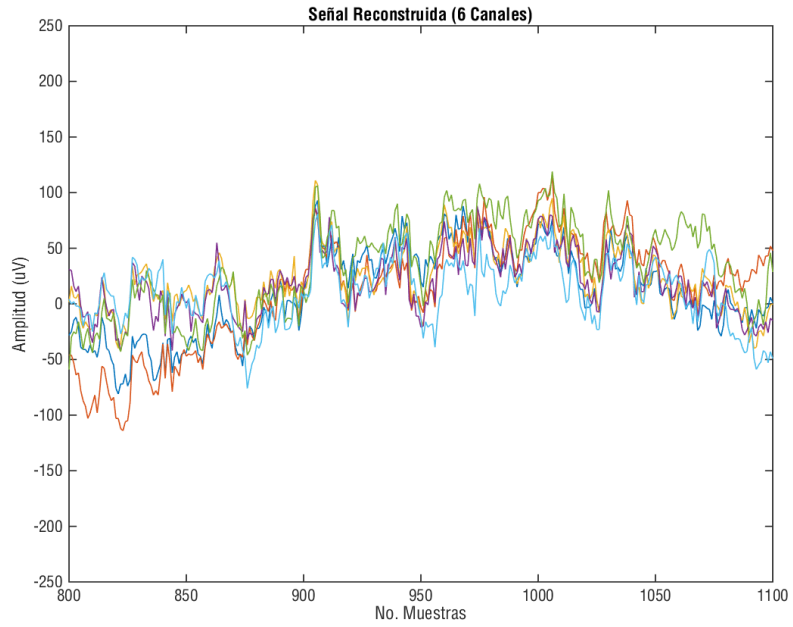


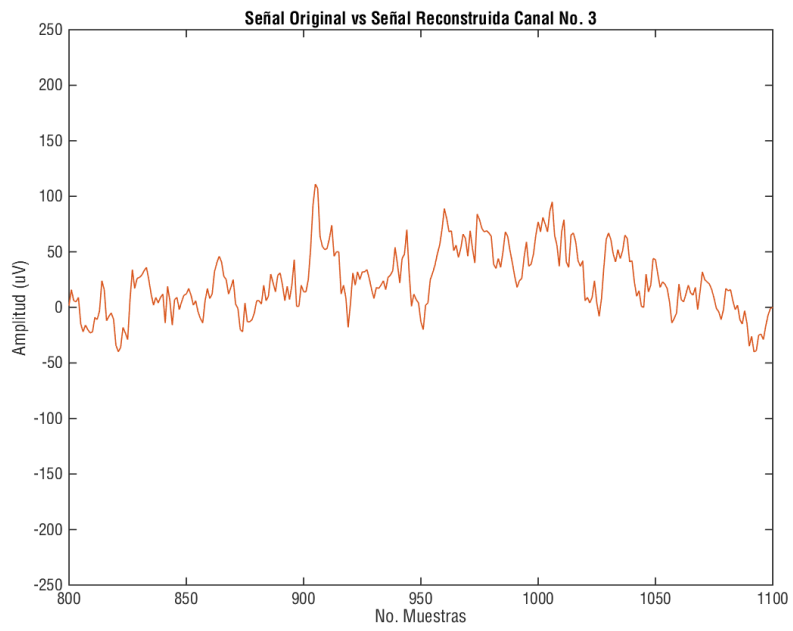
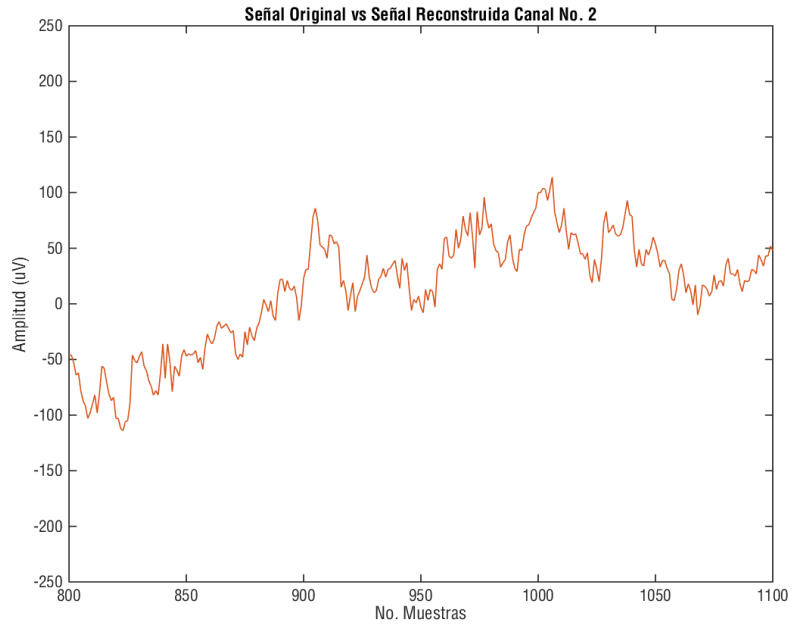


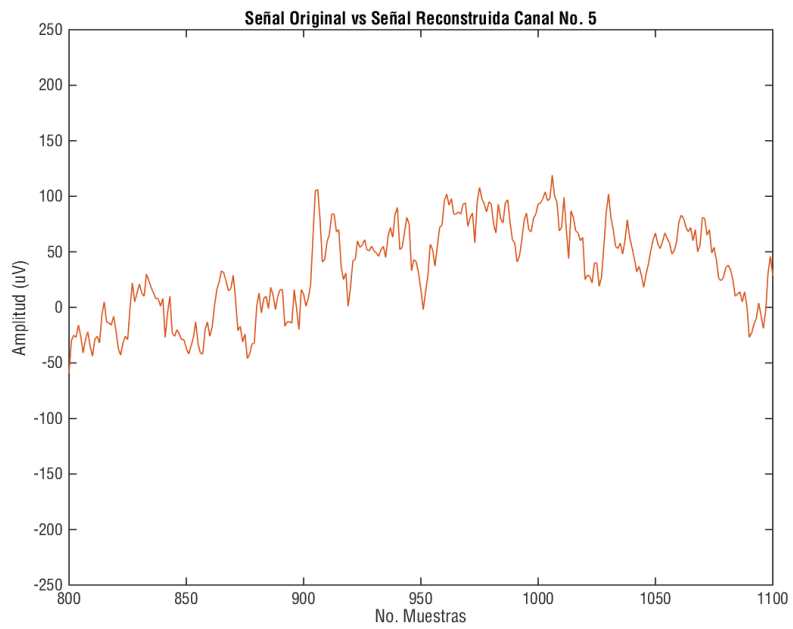
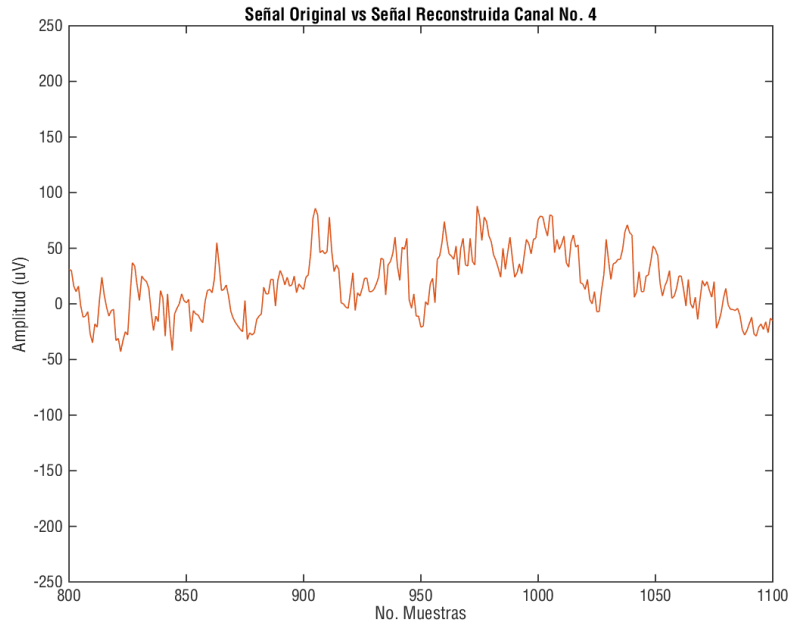
Análisis 6 Canales:

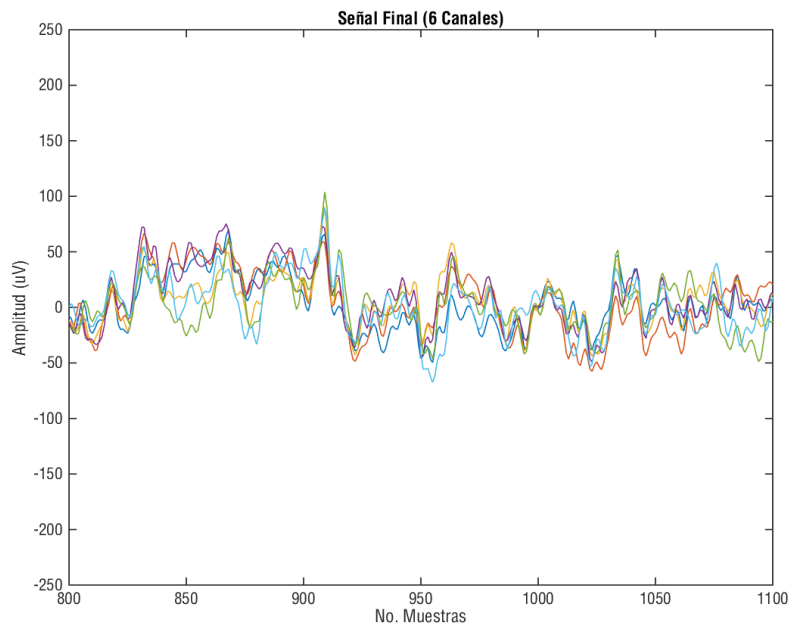
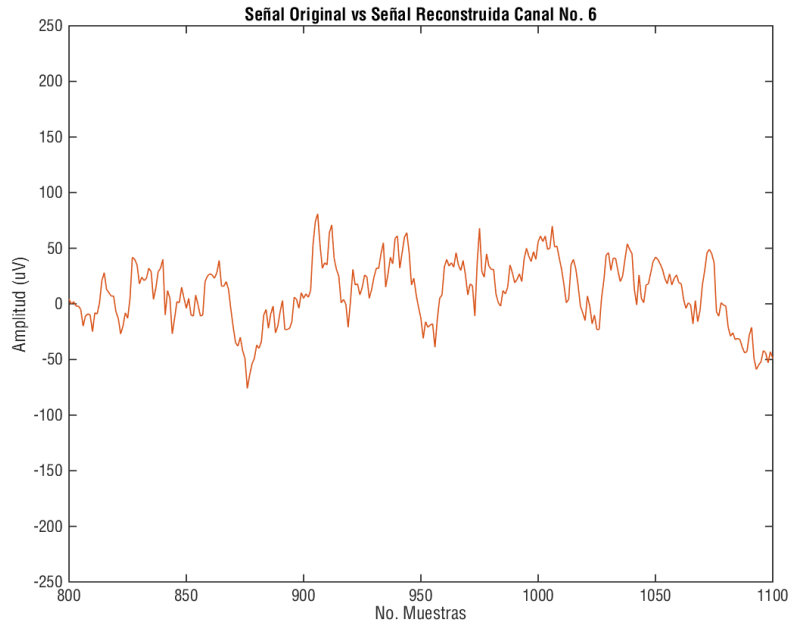






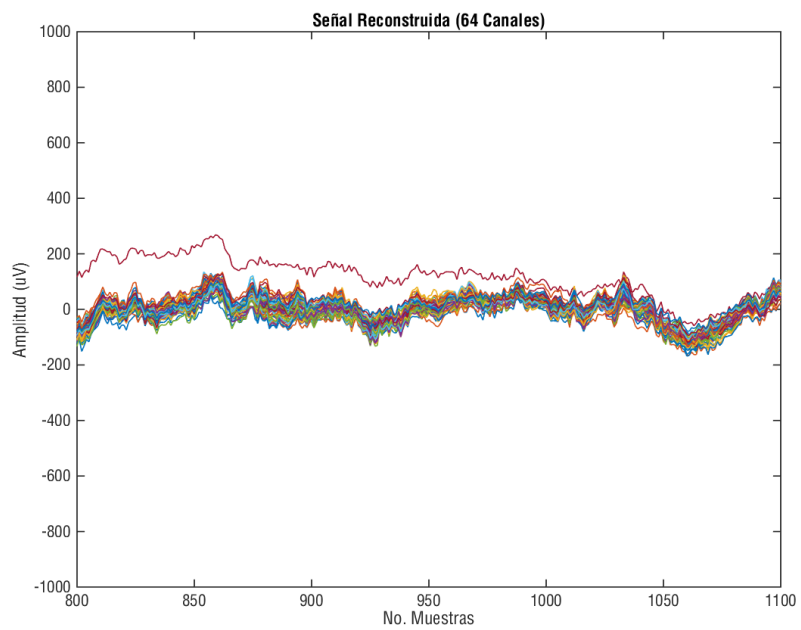
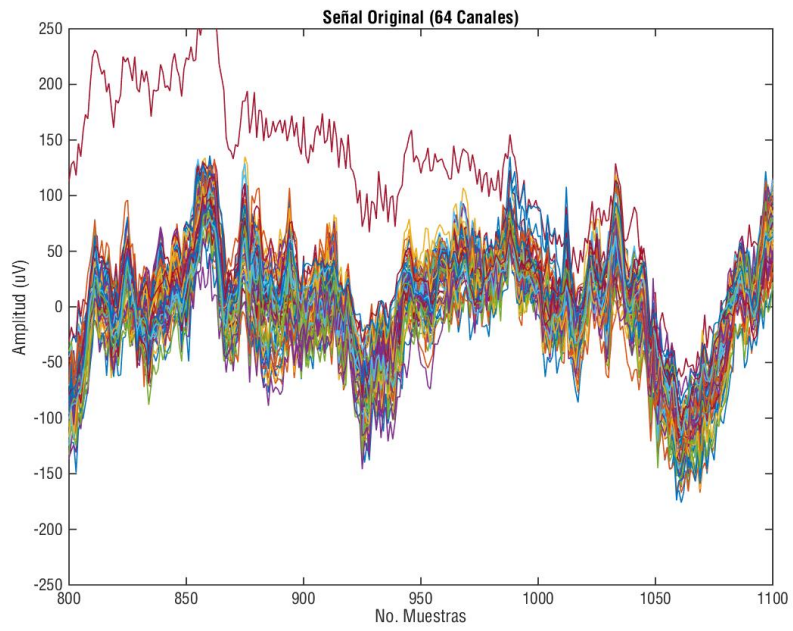


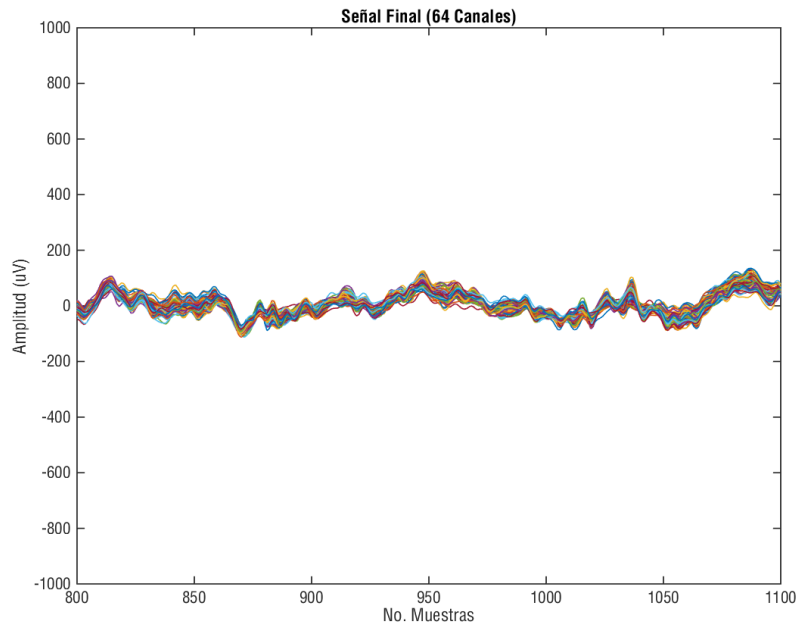




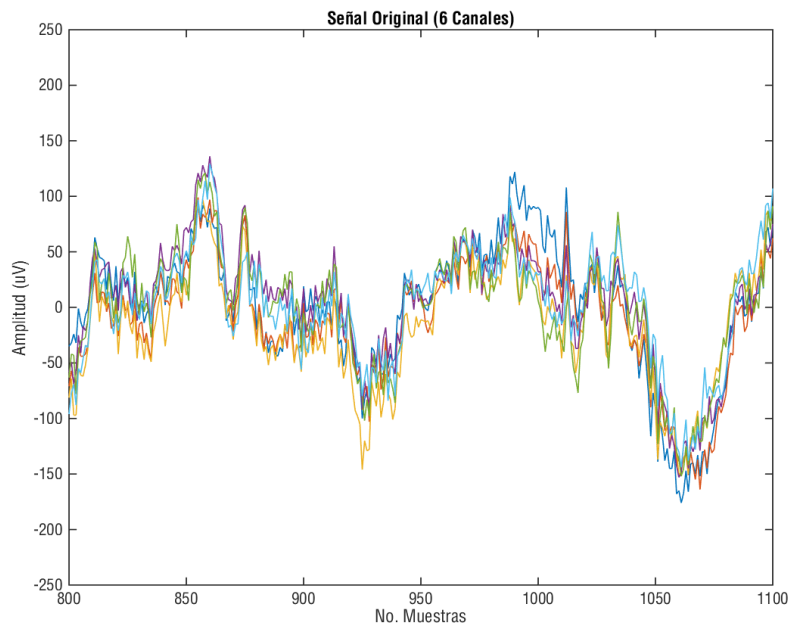
Registro Individuo número 57 tarea 9:

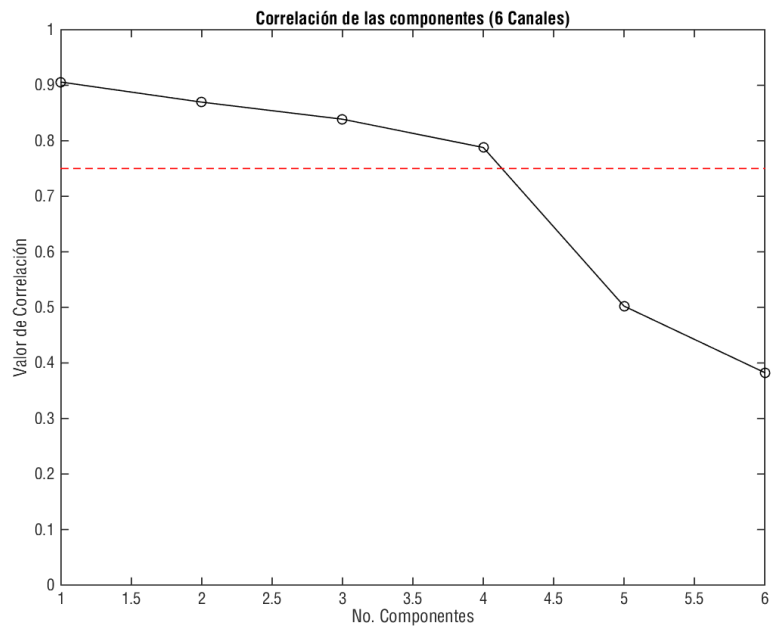
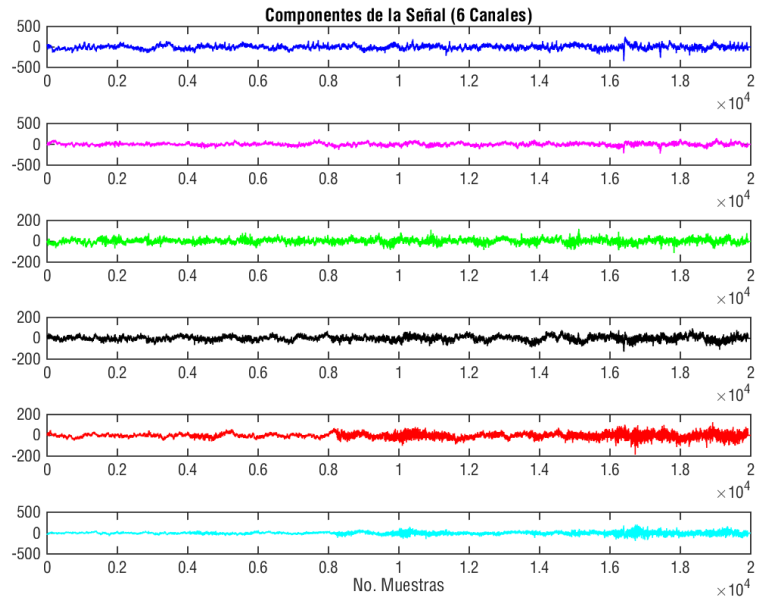
Análisis 64 Canales:

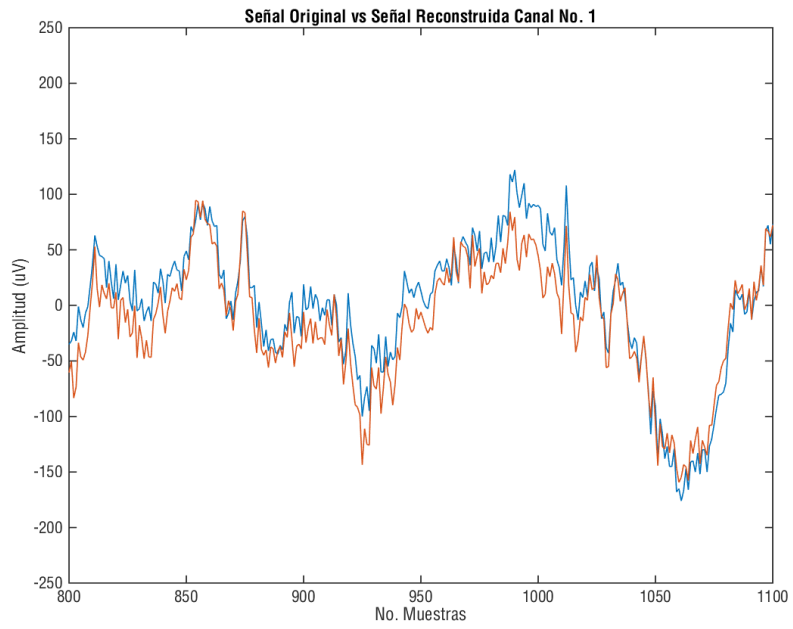
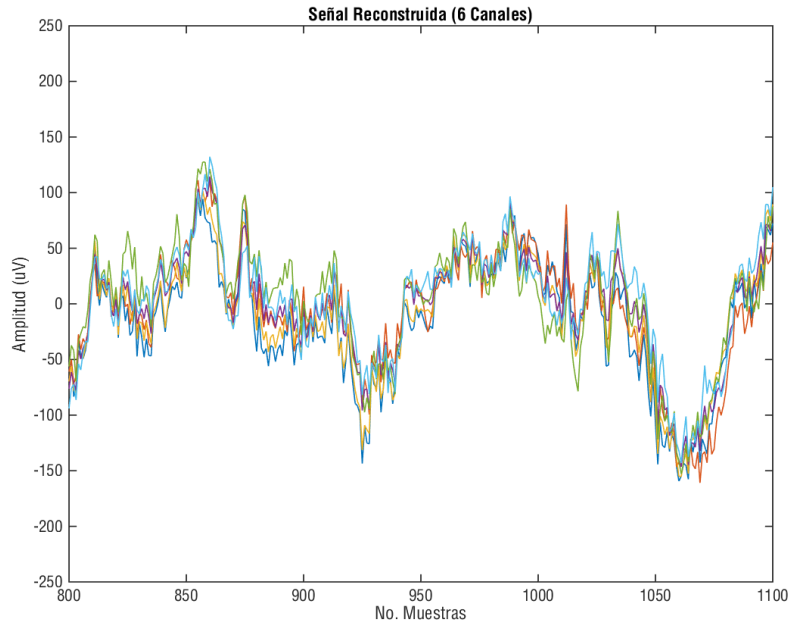


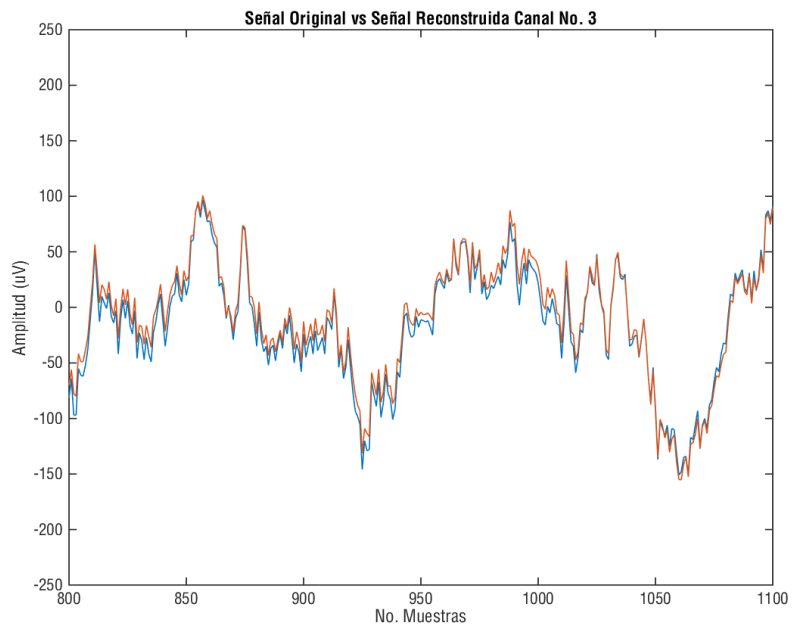
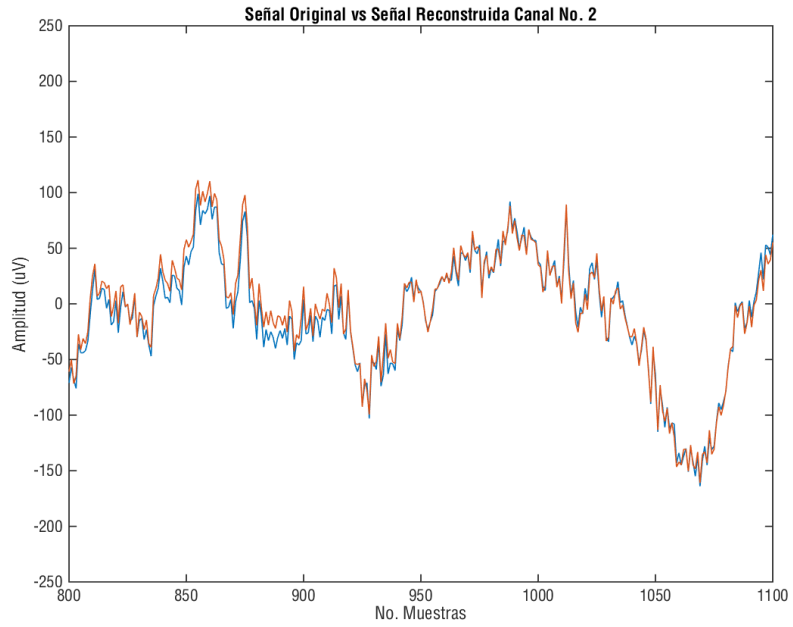


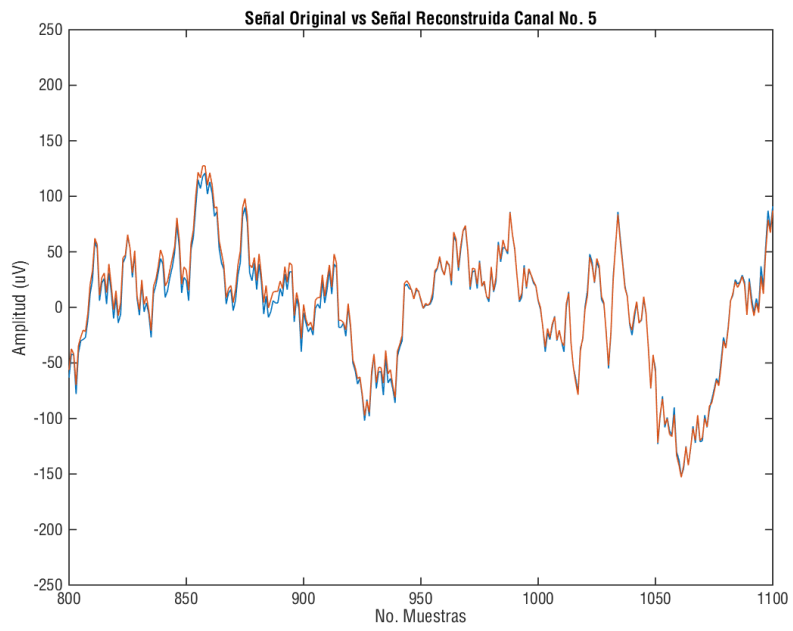
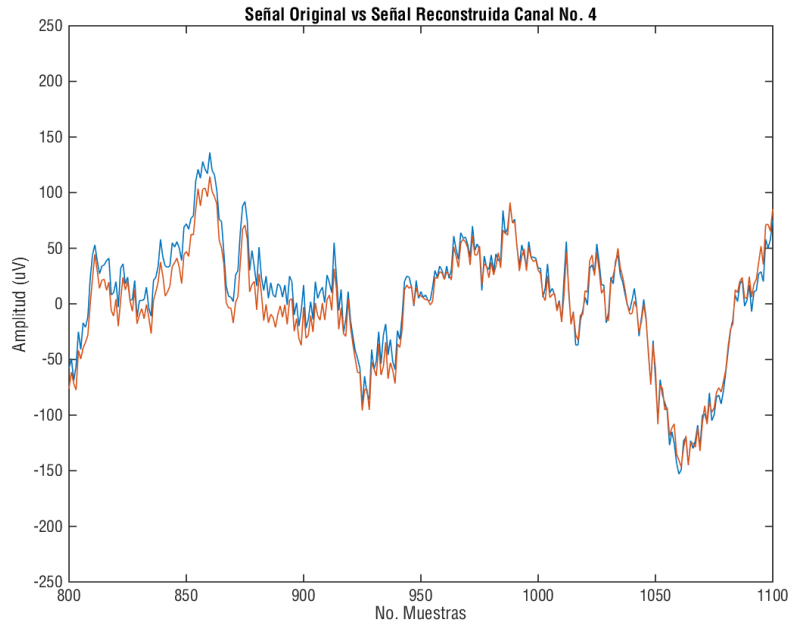
Análisis 6 Canales:

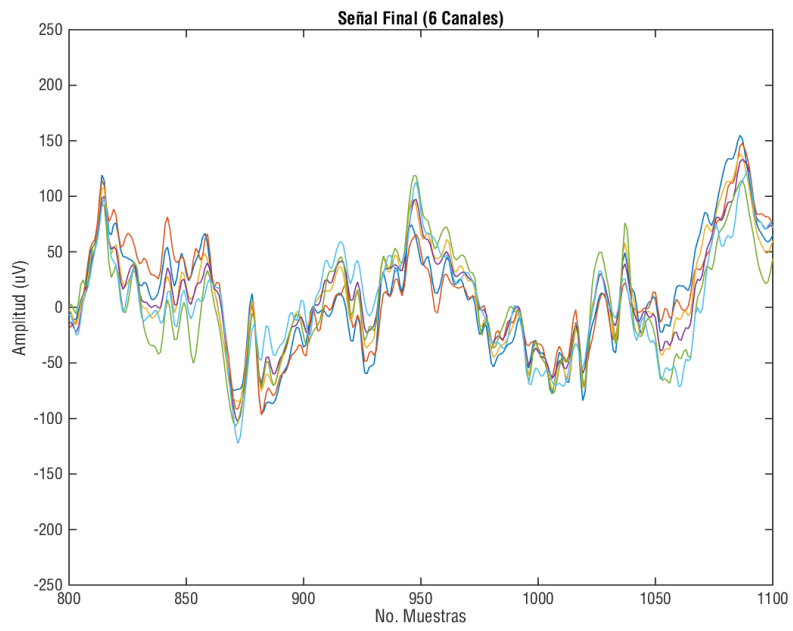
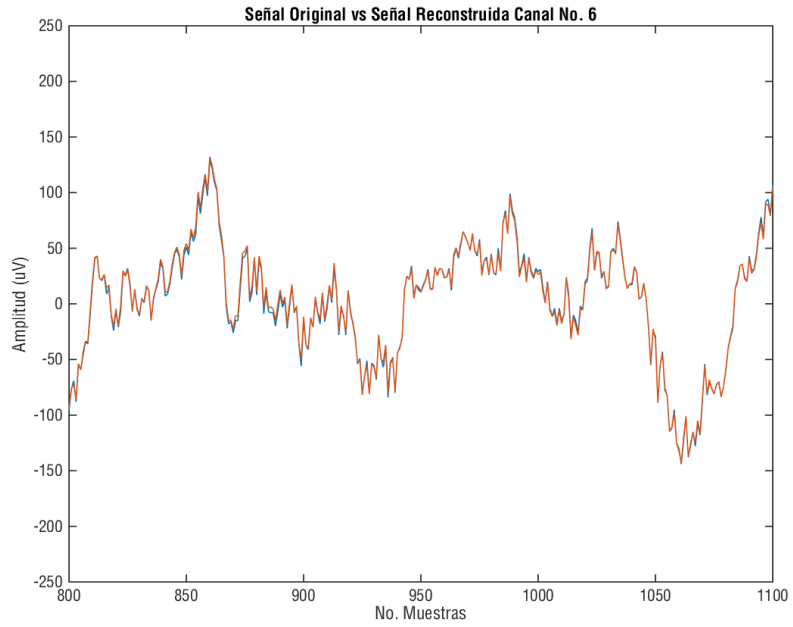






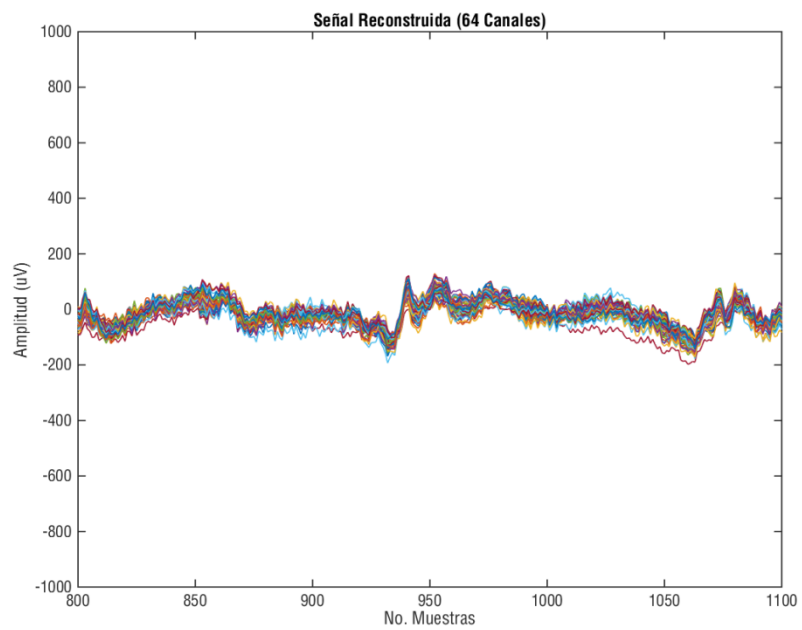
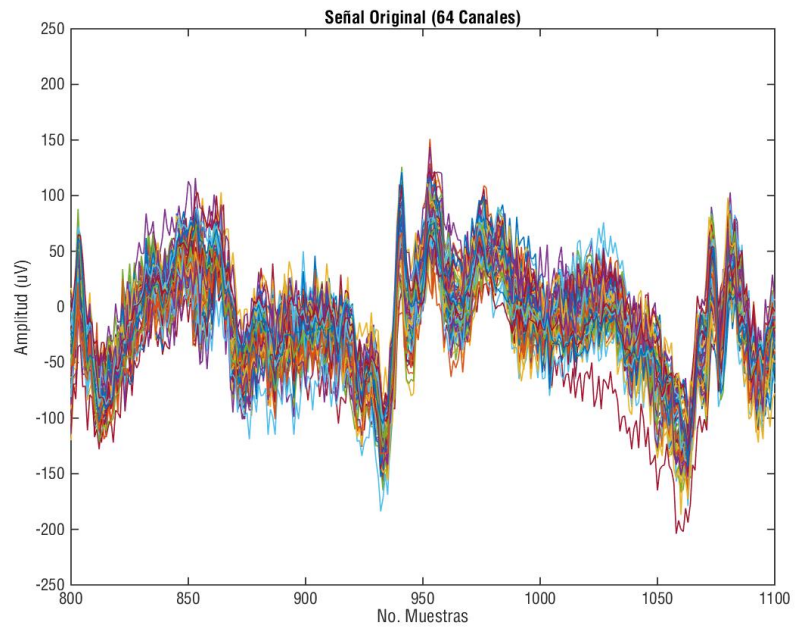


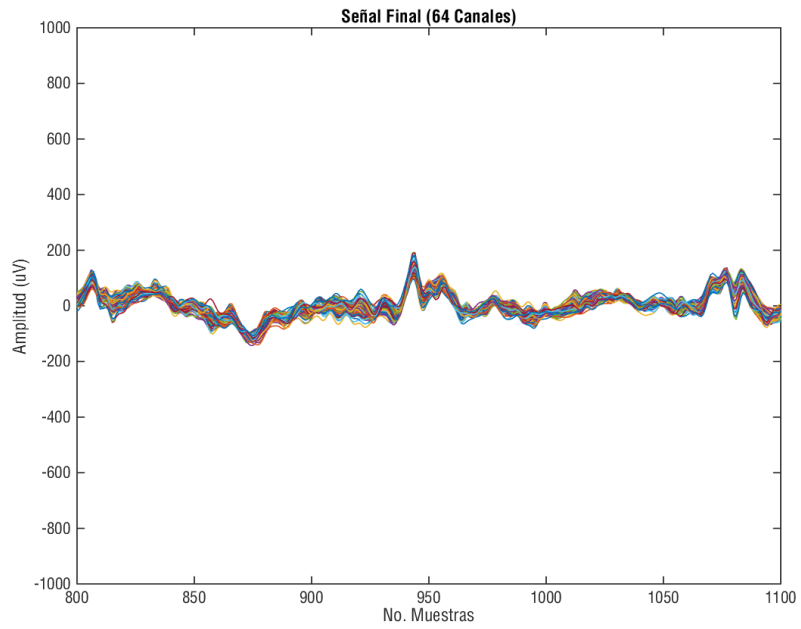




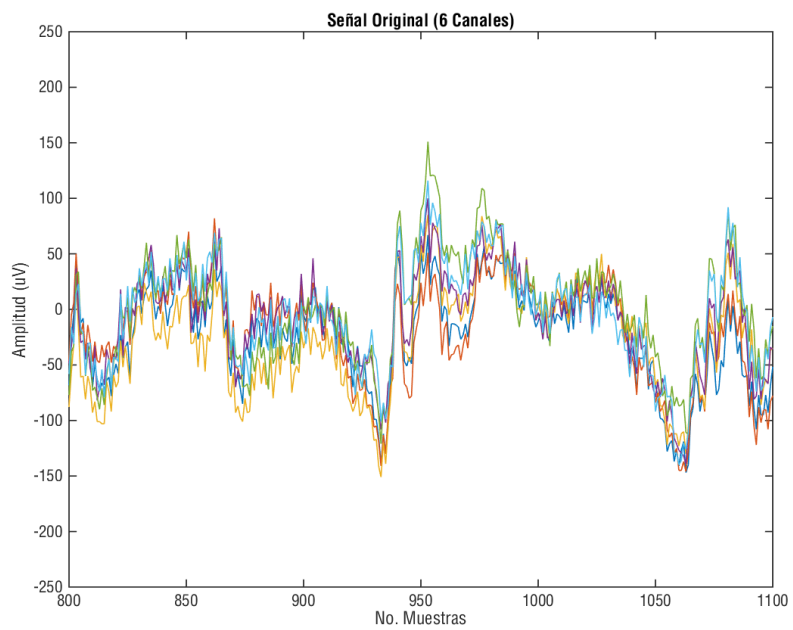
Registro Individuo número 57 tarea 13:

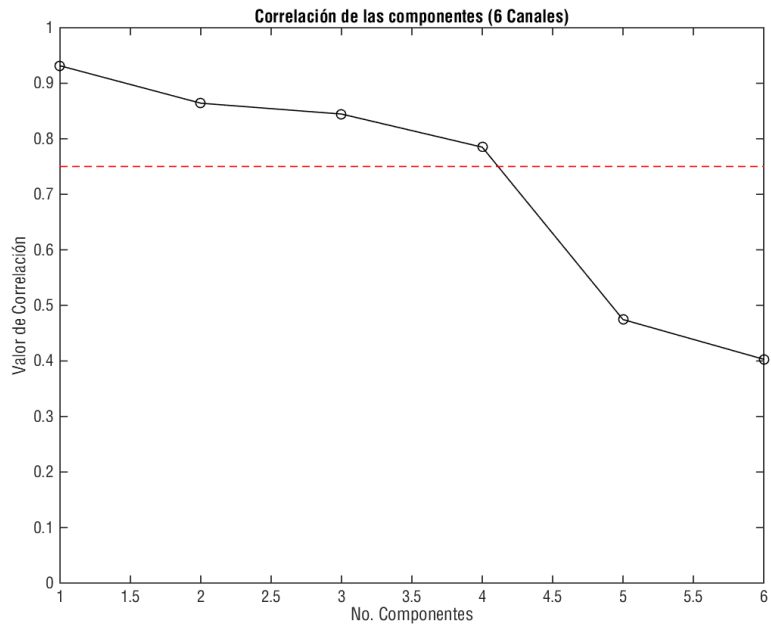
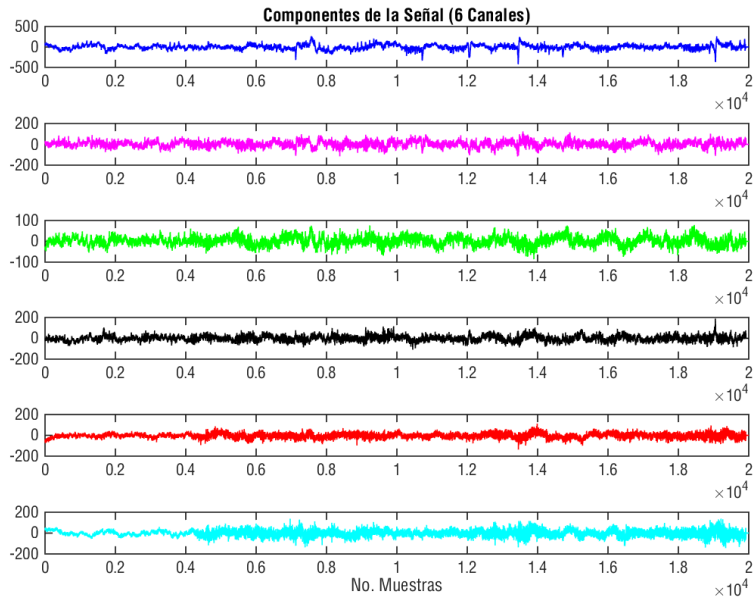
Análisis 64 Canales:

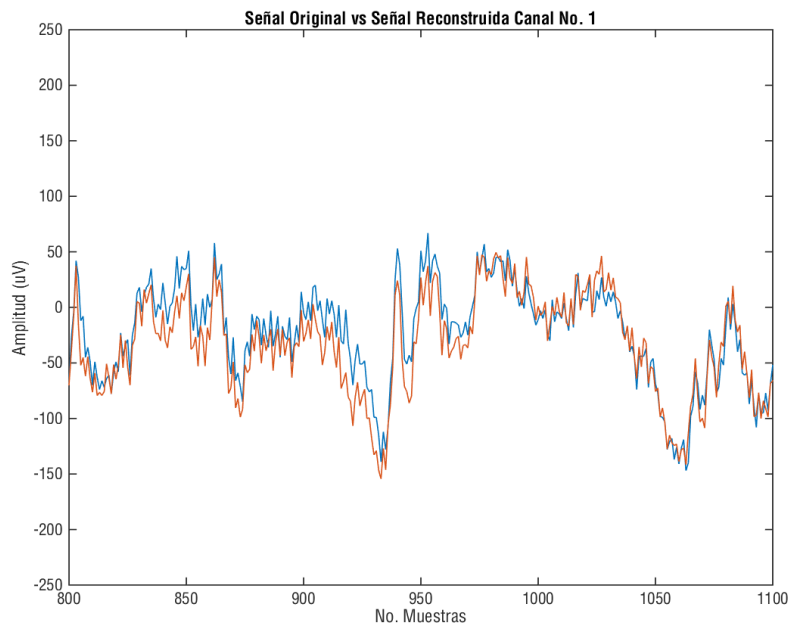
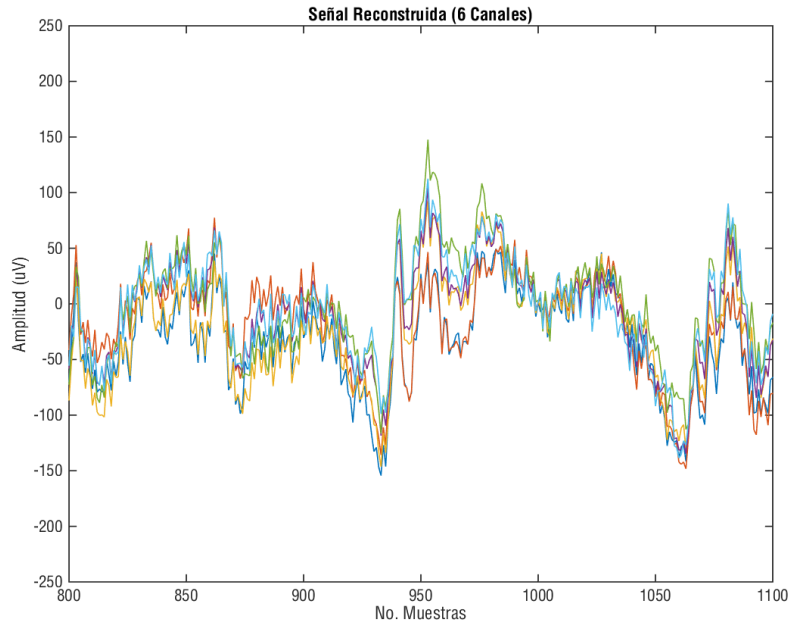


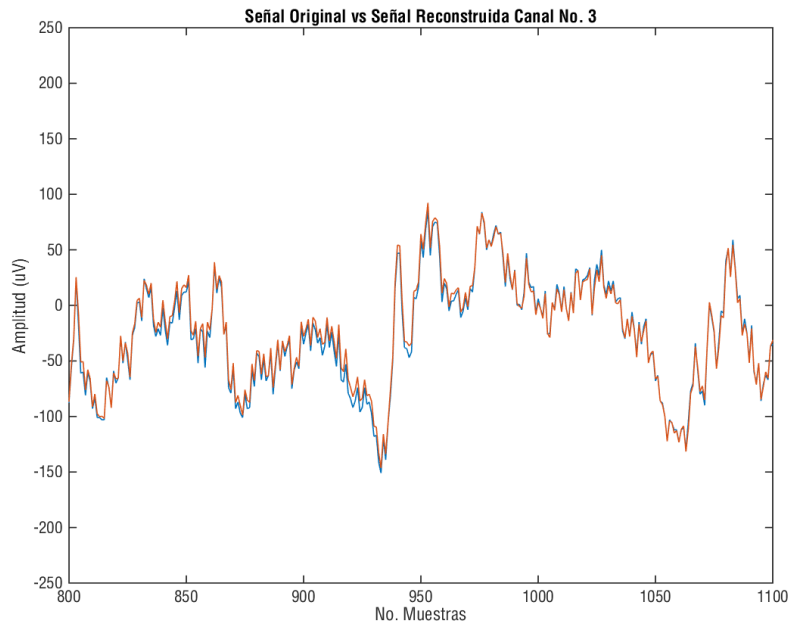
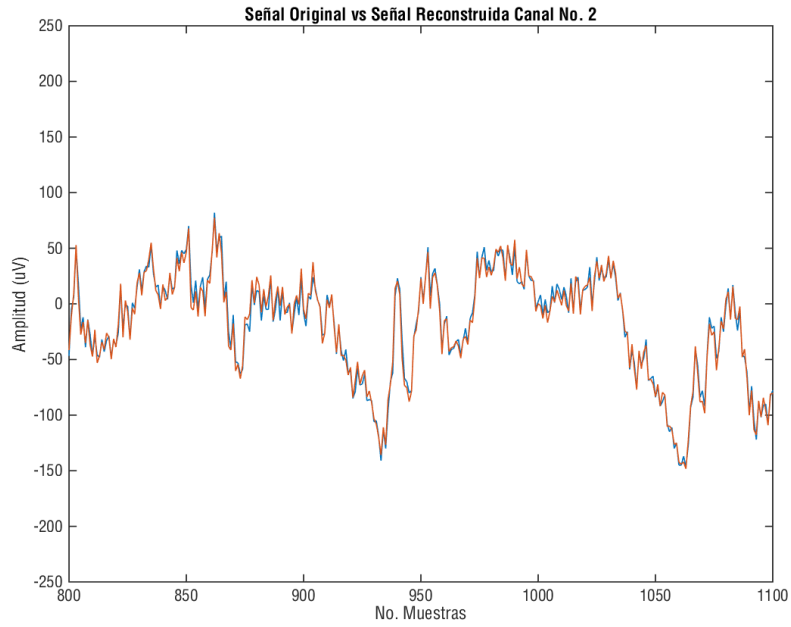


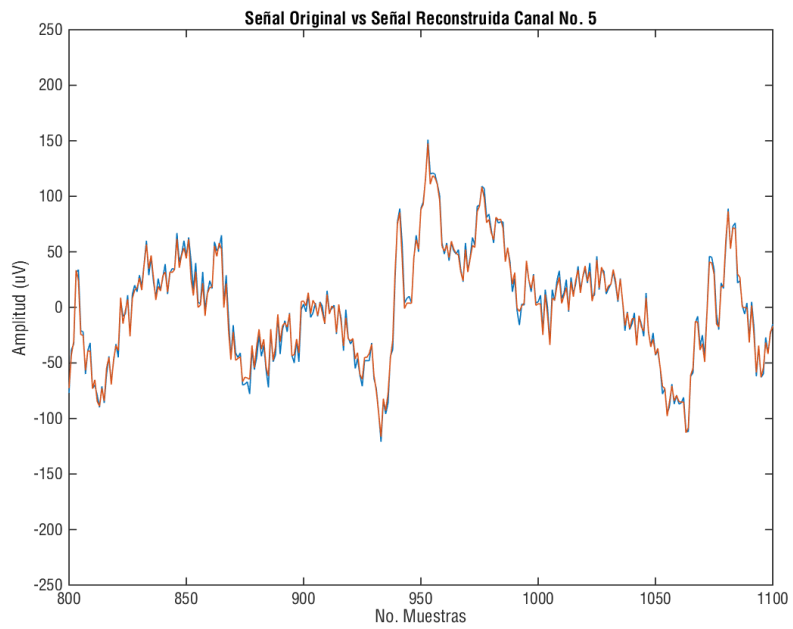
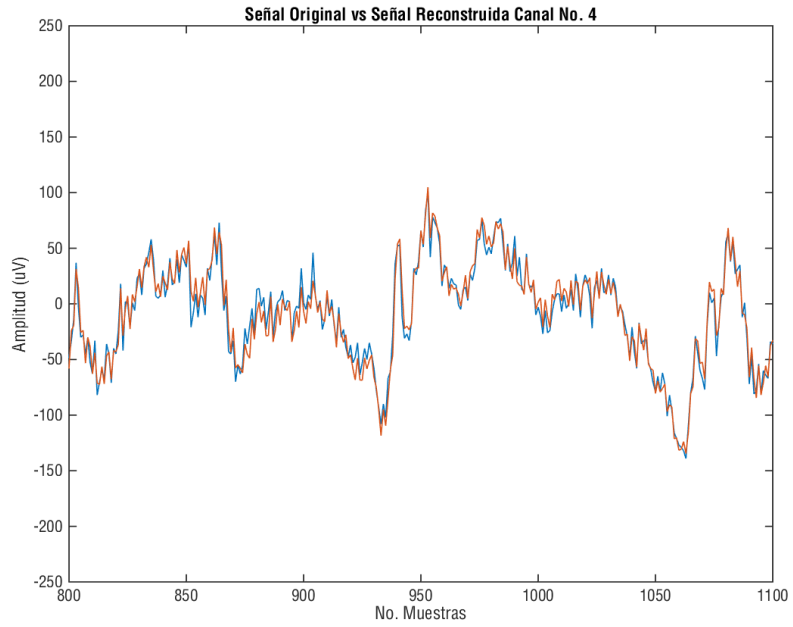
Análisis 6 Canales:

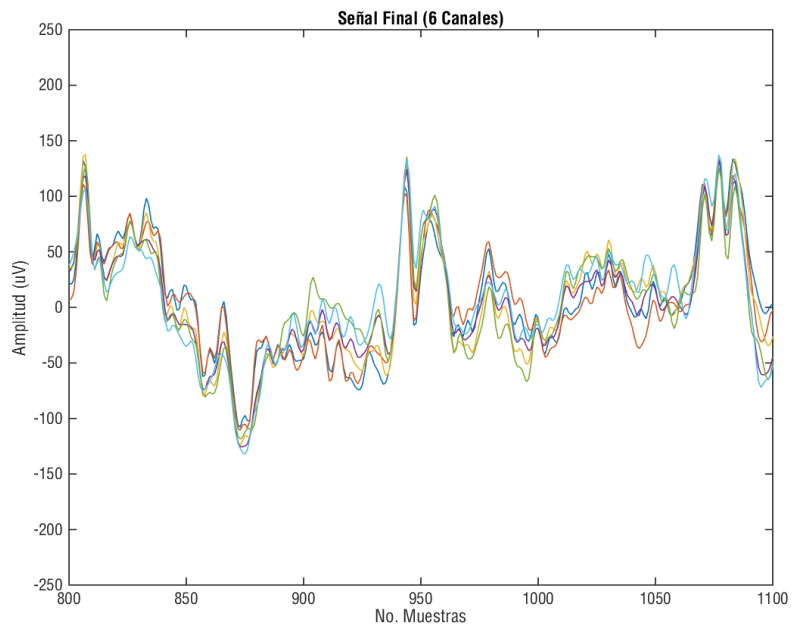
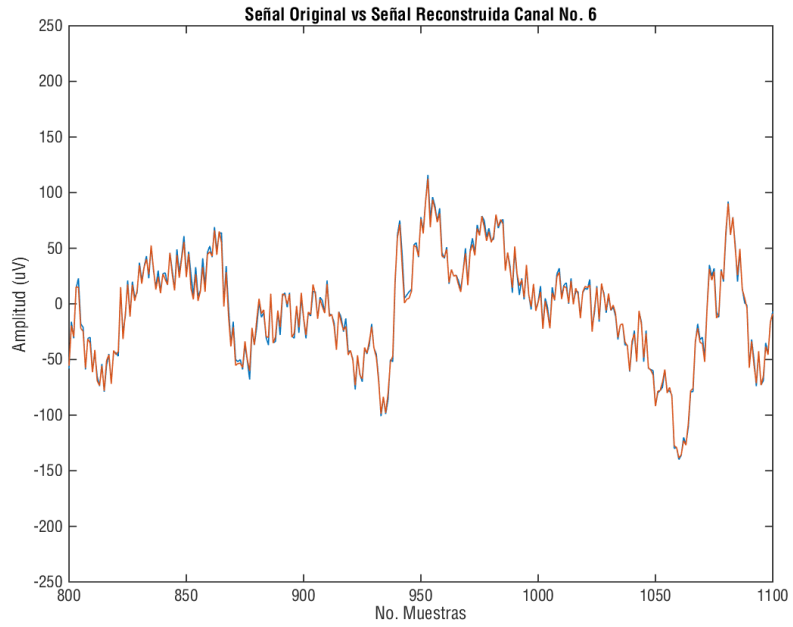






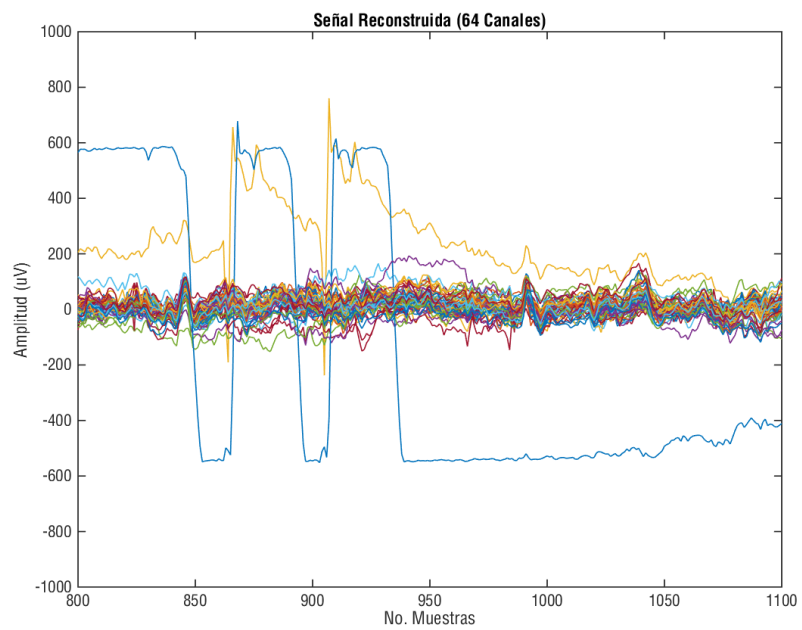
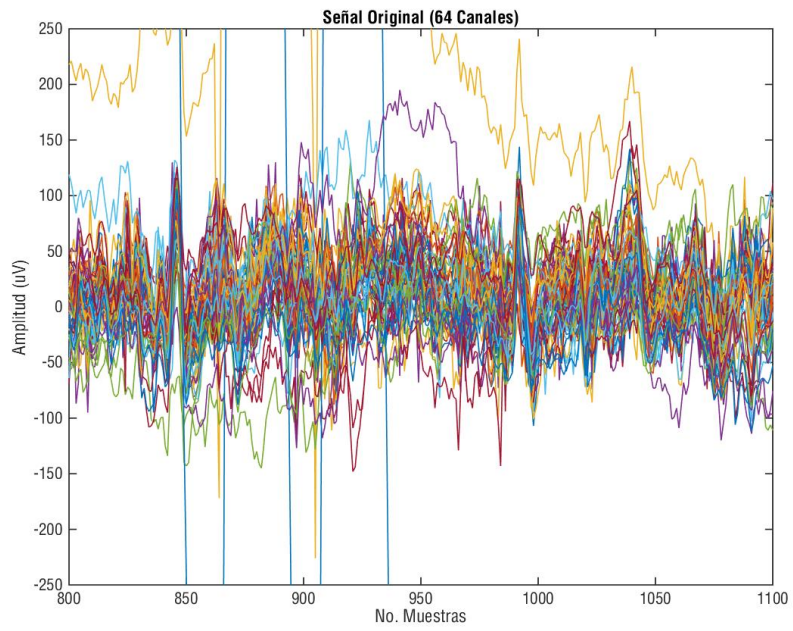


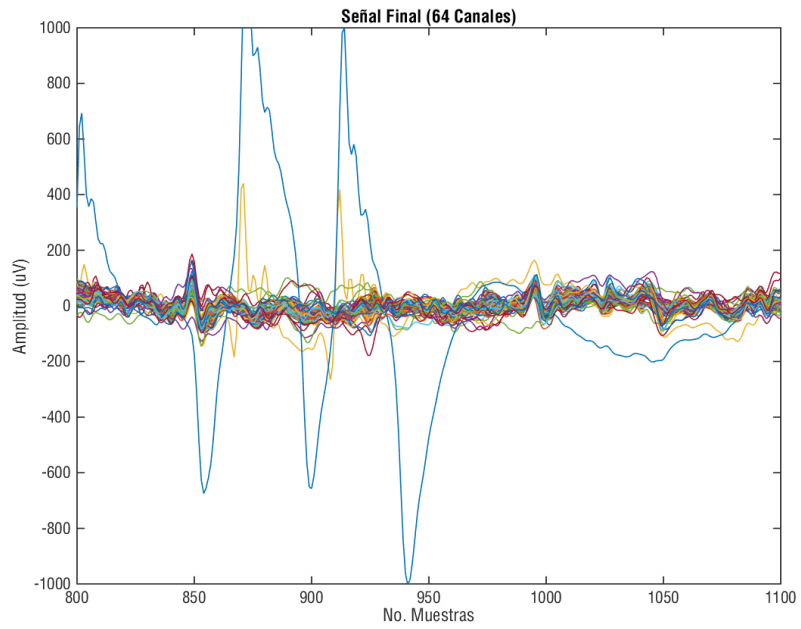




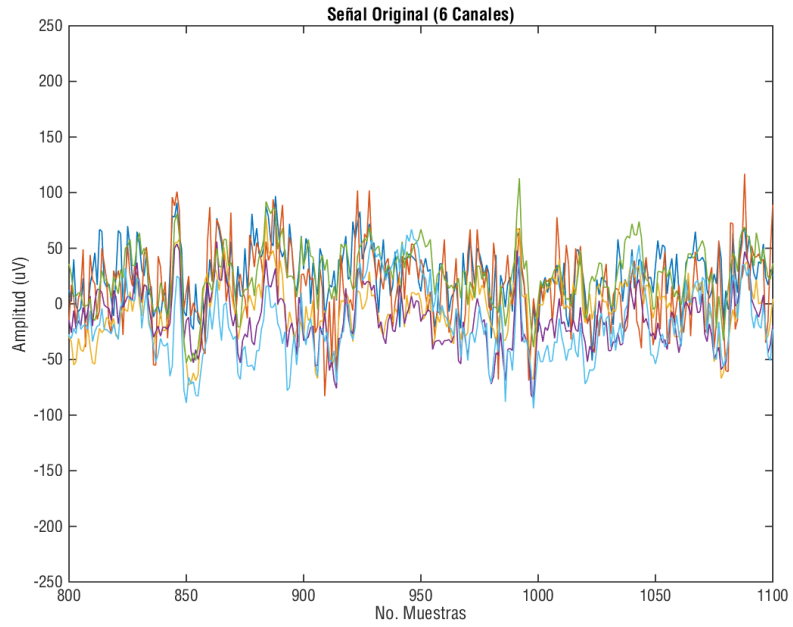
Registro Individuo número 65 tarea 5:

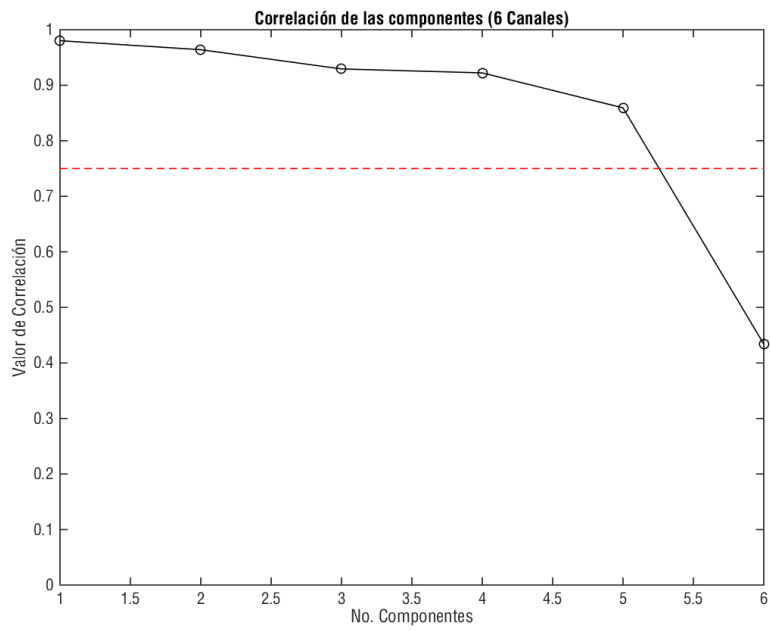
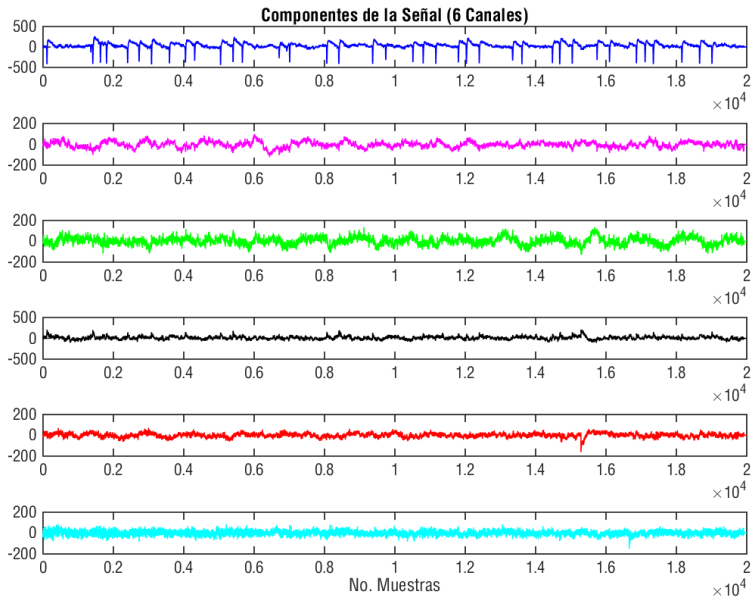
Análisis 64 Canales:

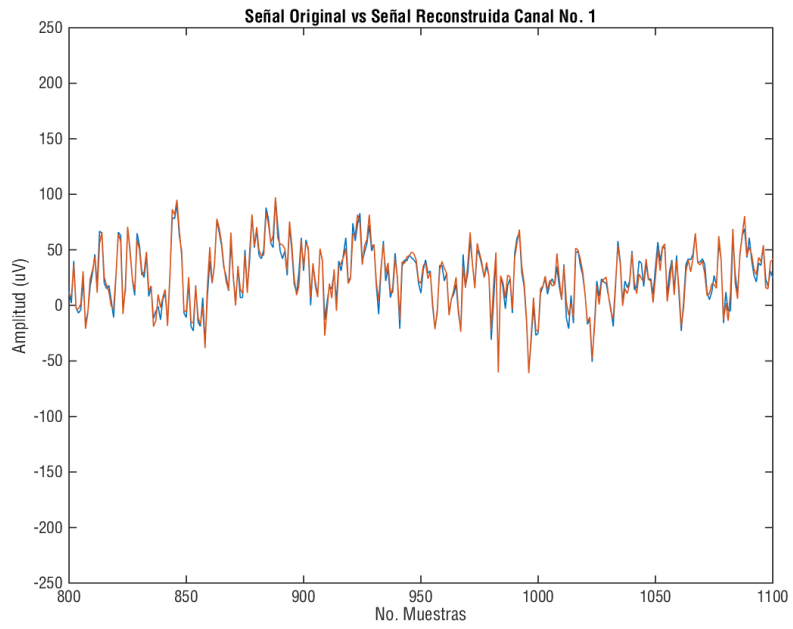
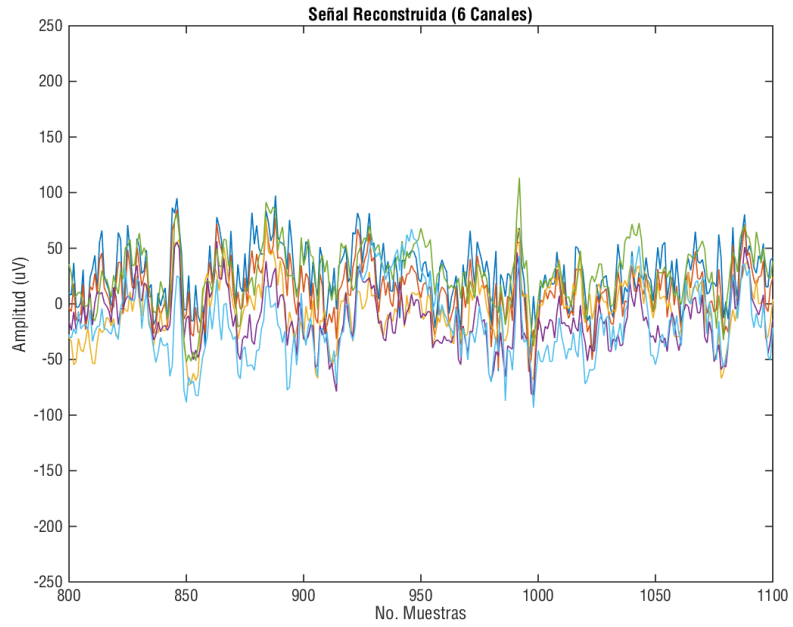


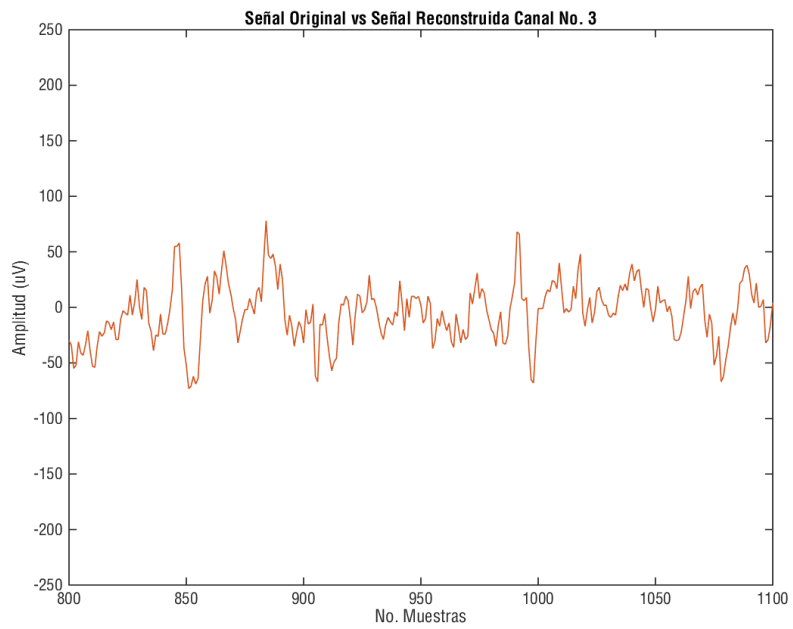
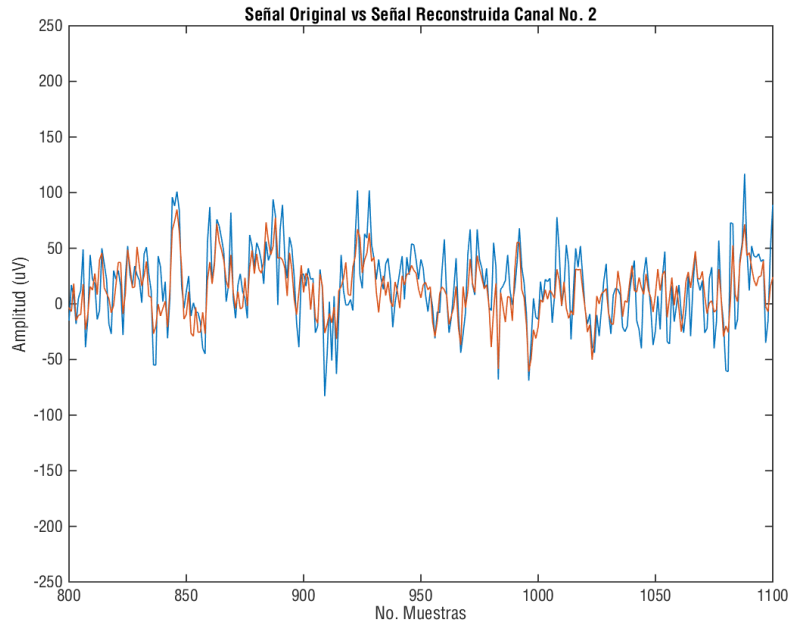


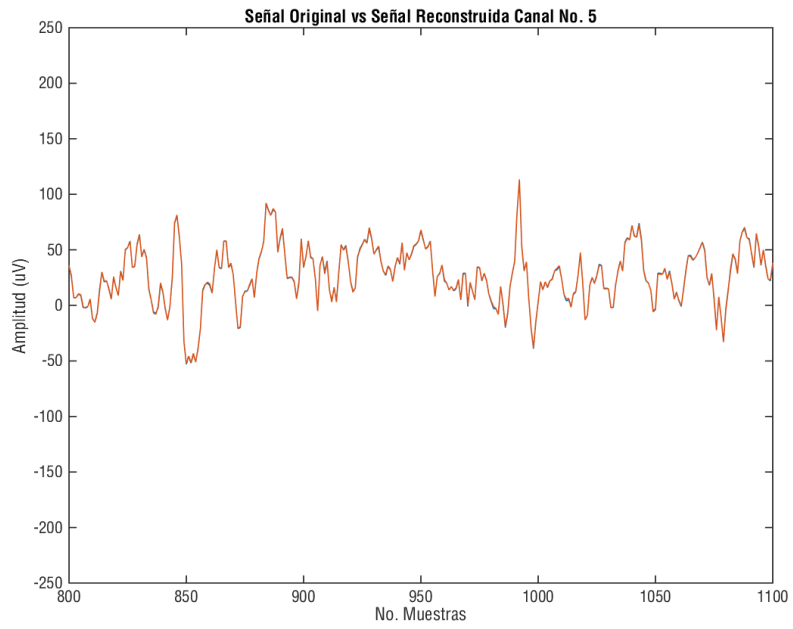
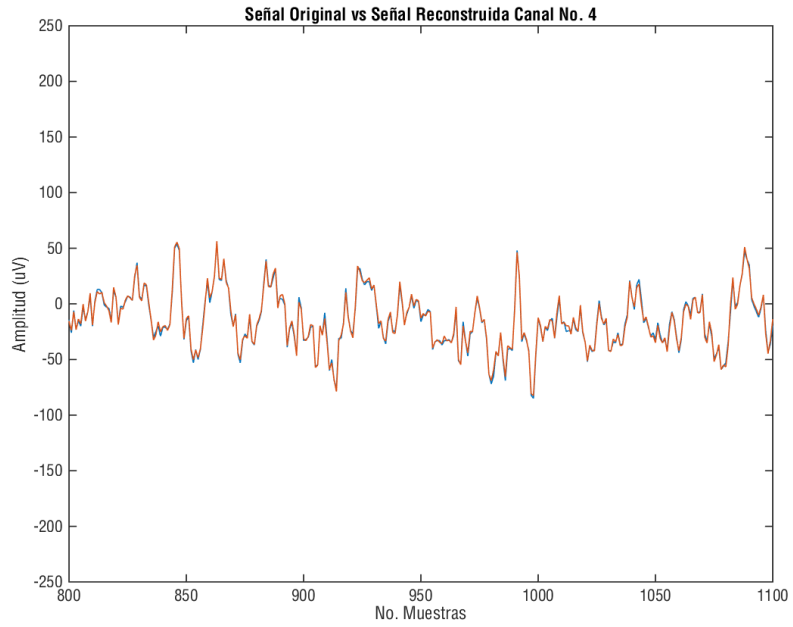
Análisis 6 Canales:

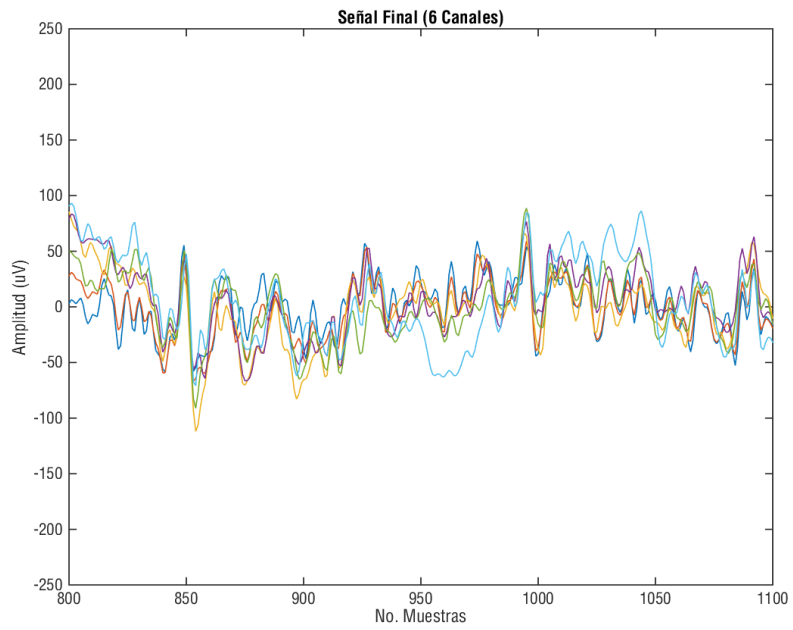
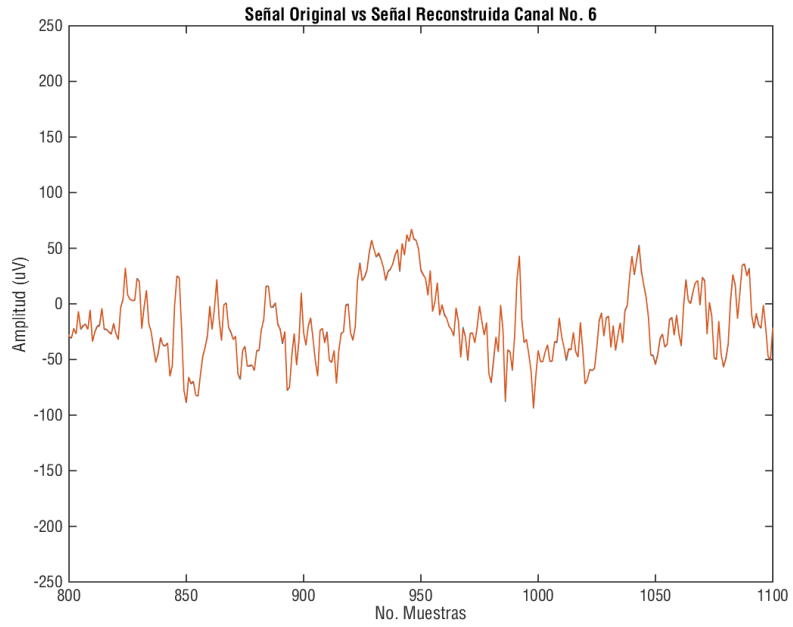






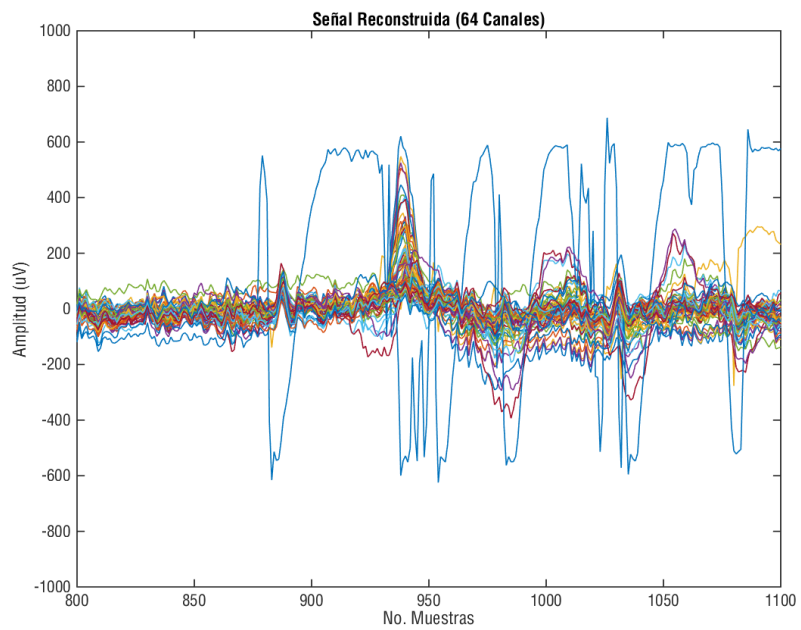
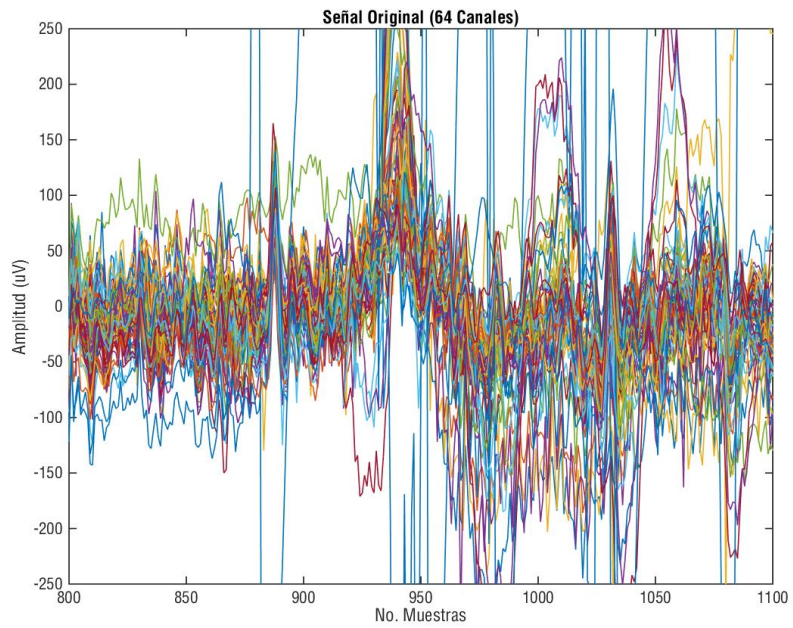


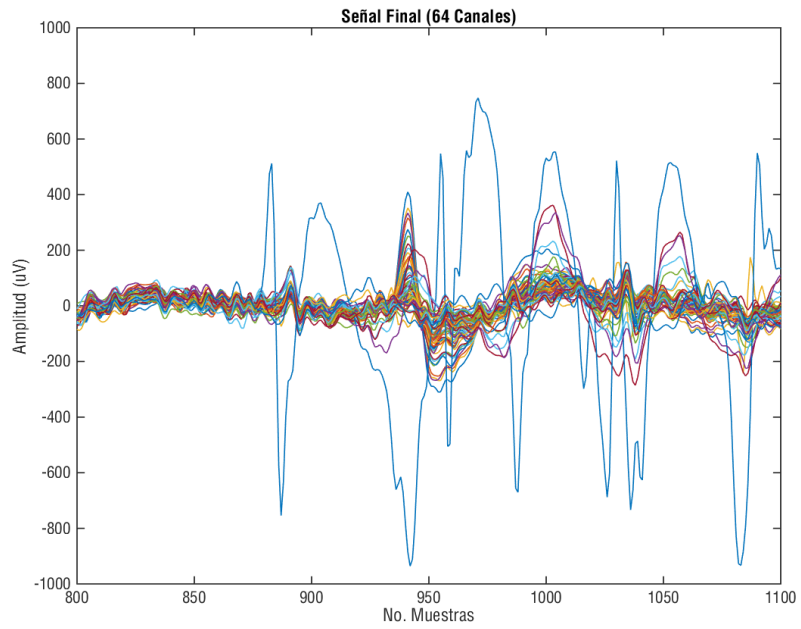




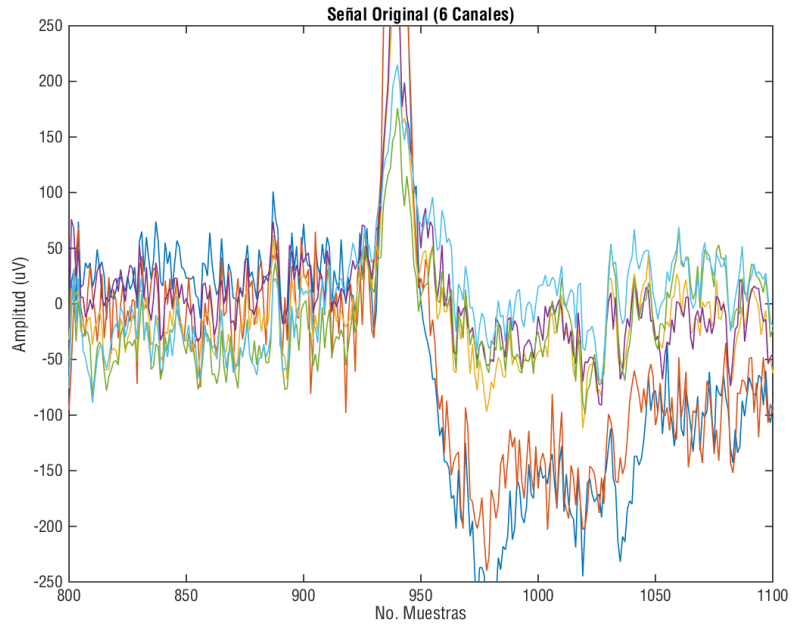
Registro Individuo número 65 tarea 9:

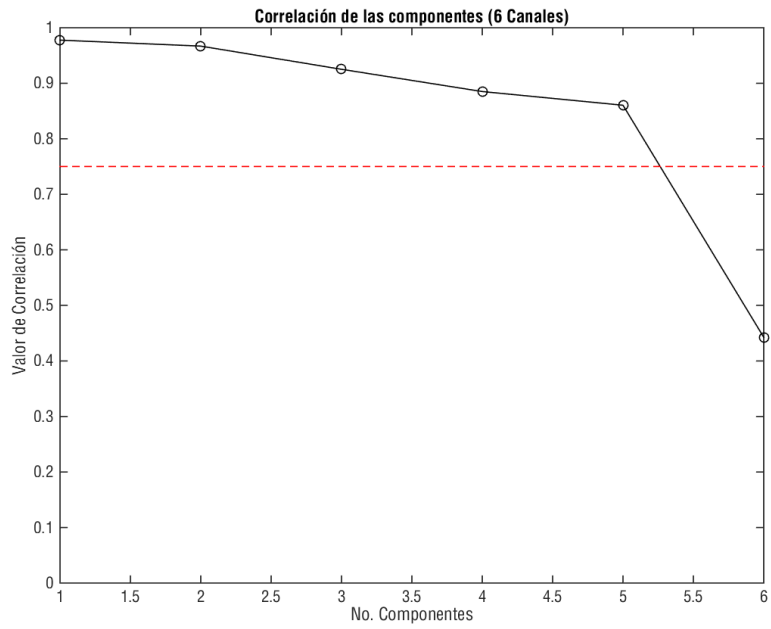
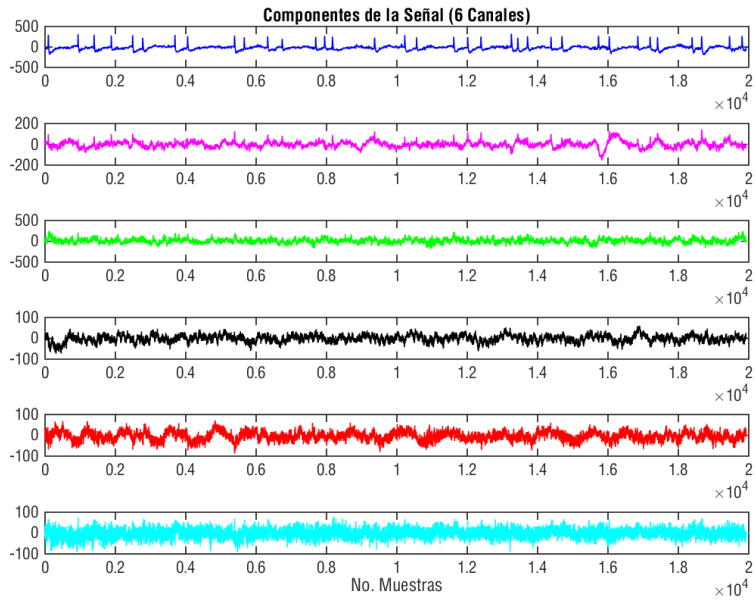
Análisis 64 Canales:

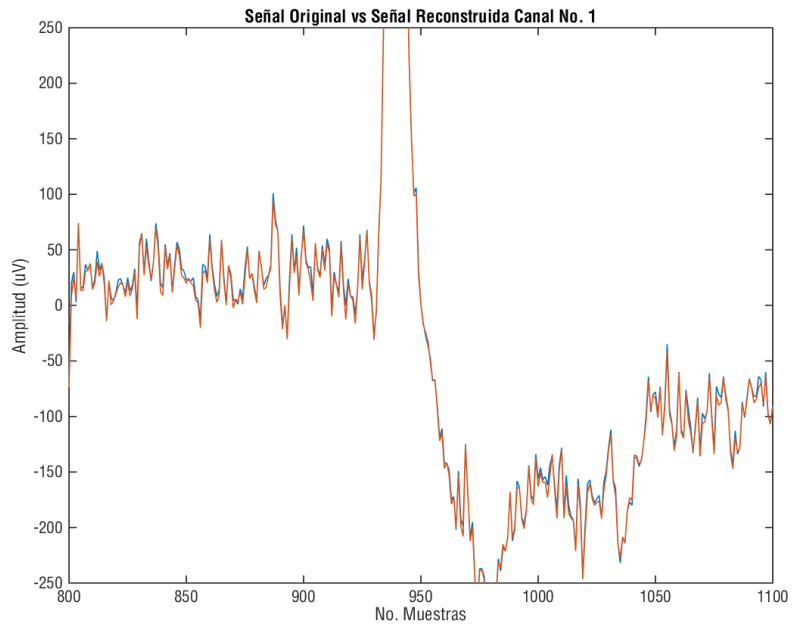
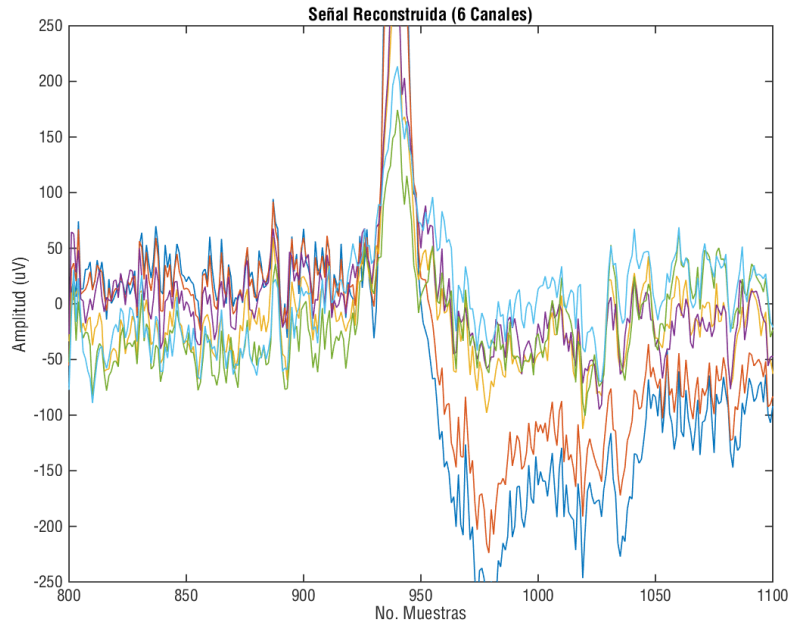


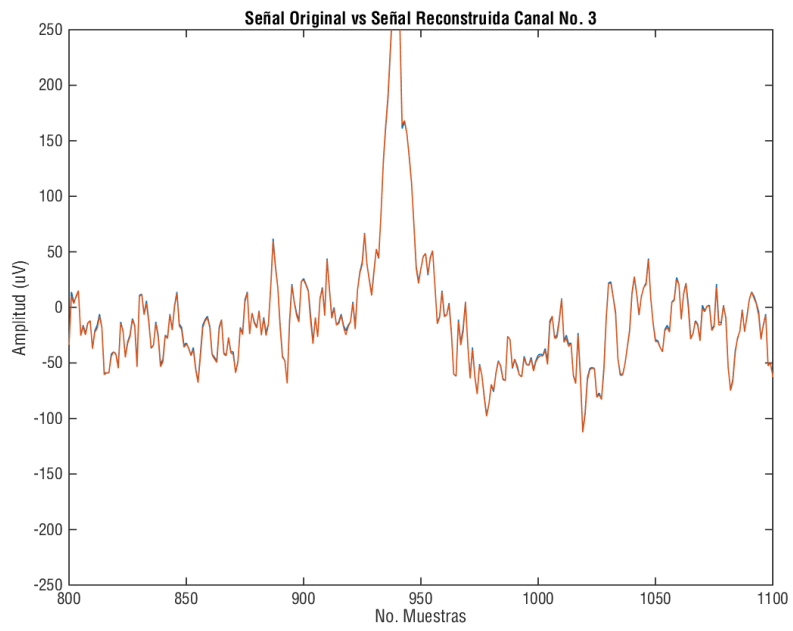
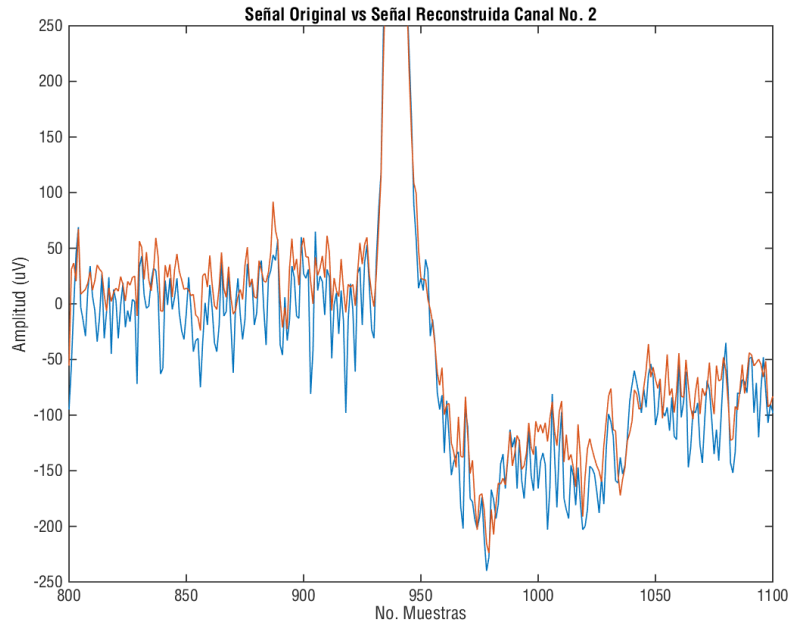


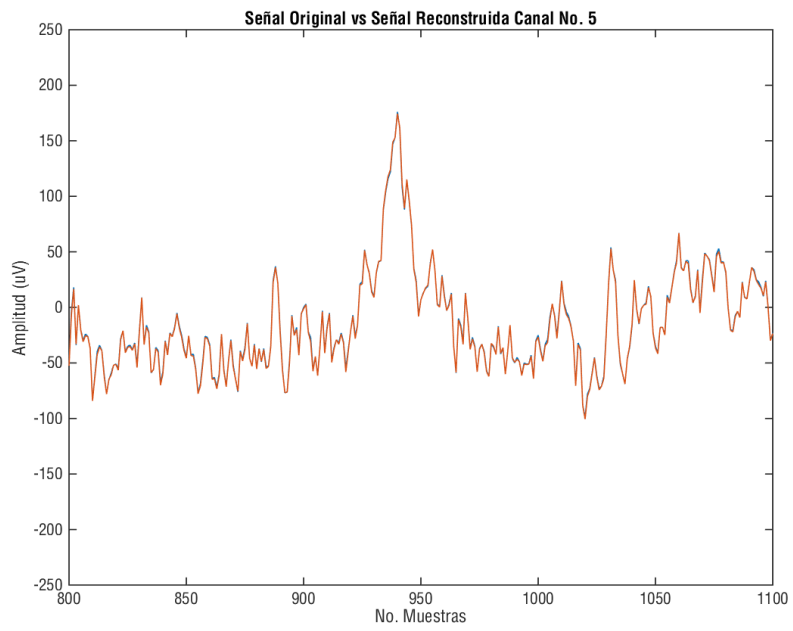
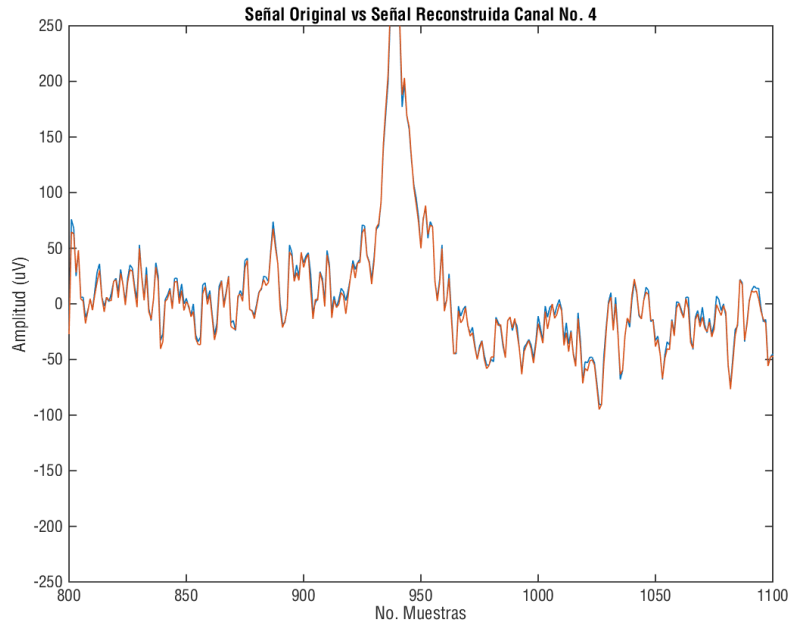
Análisis 6 Canales:

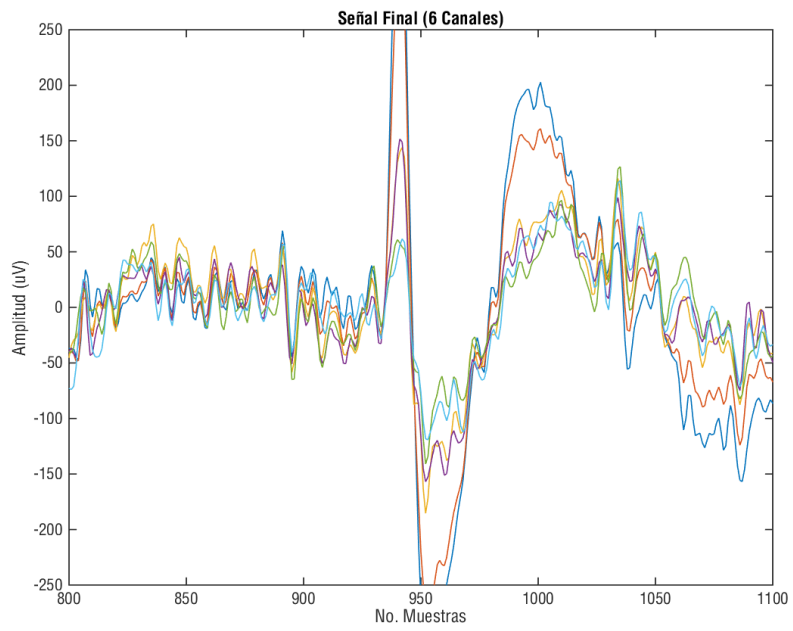
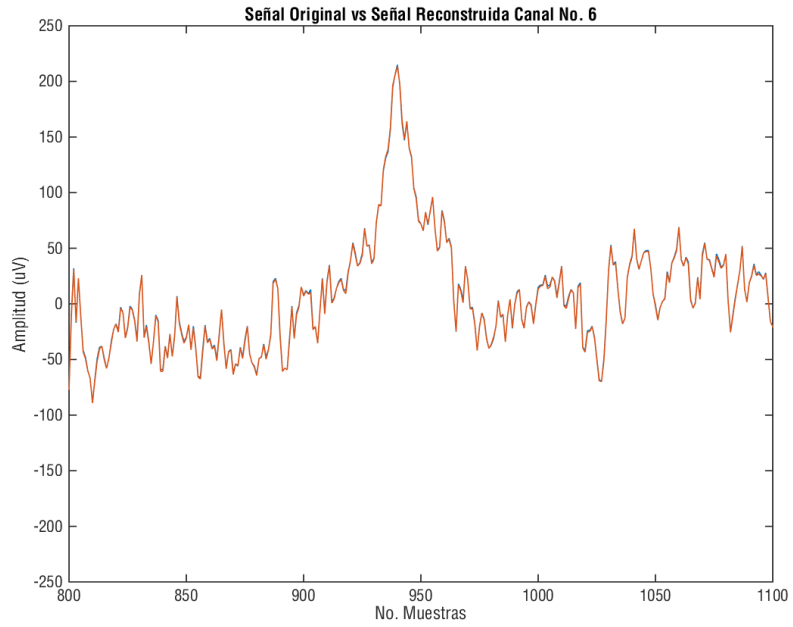






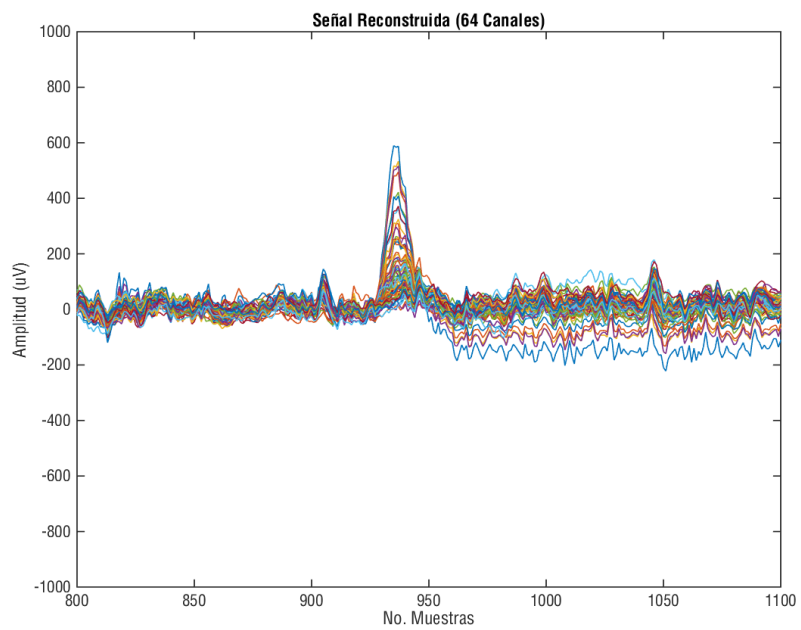
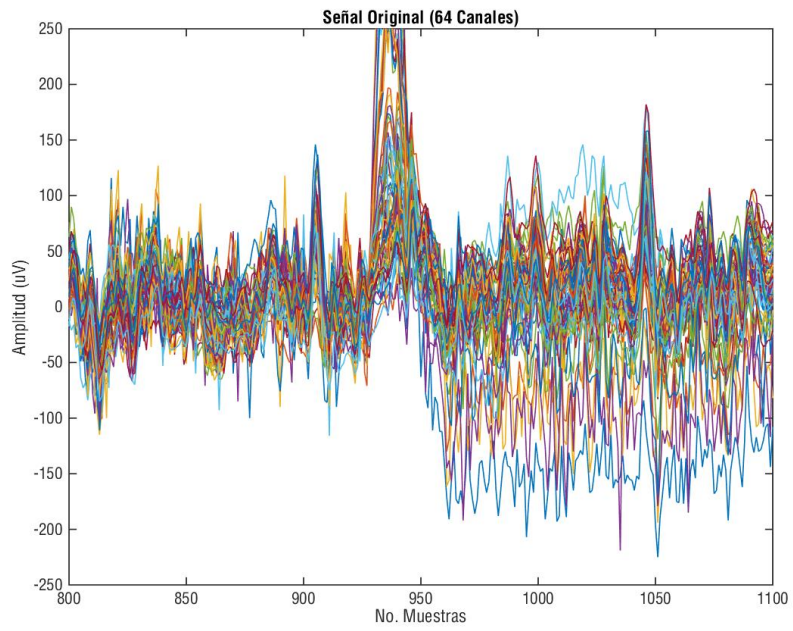


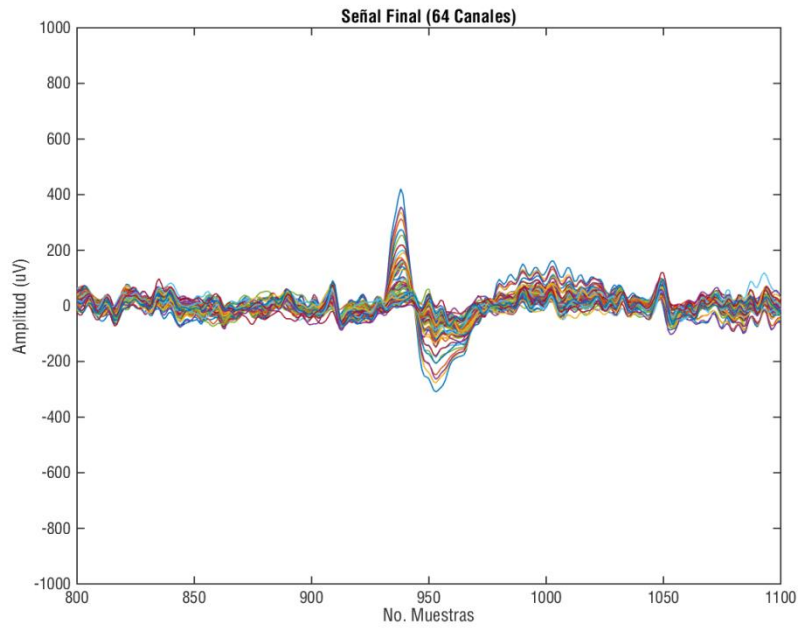




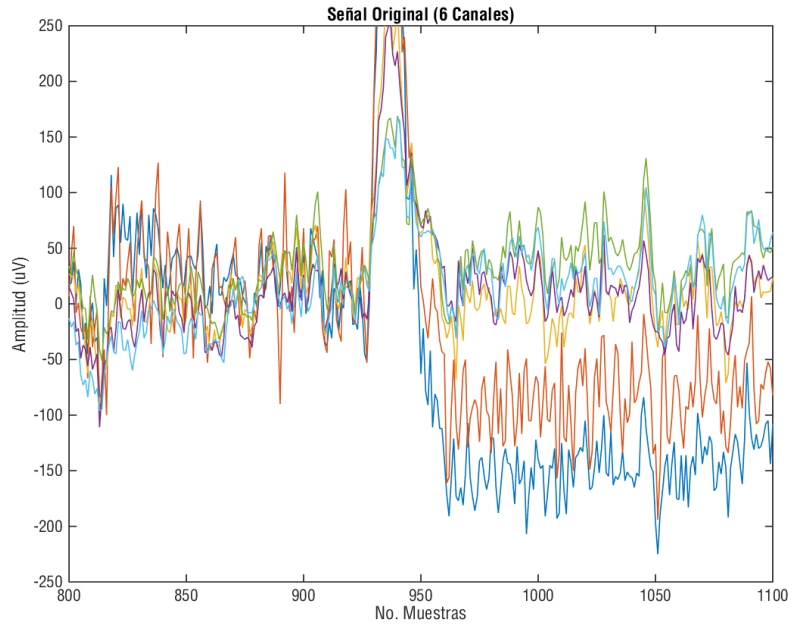
Registro Individuo número 65 tarea 13:

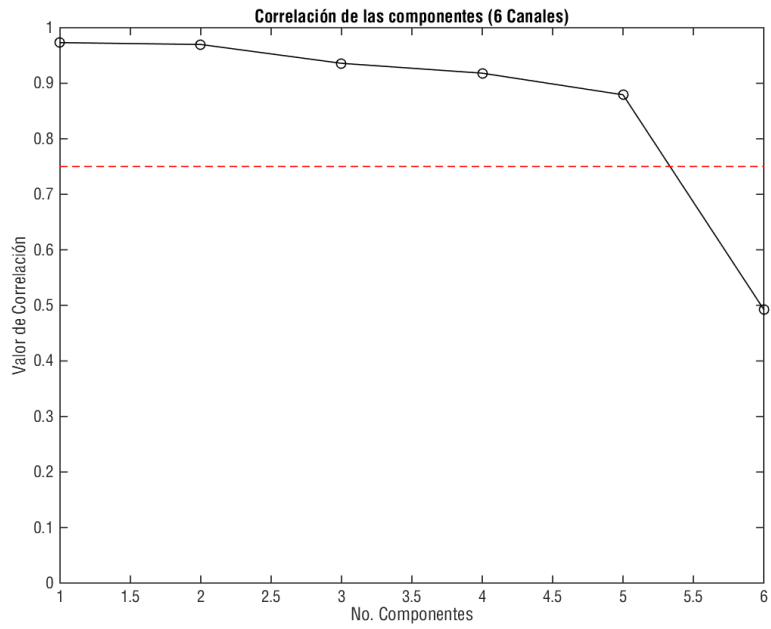
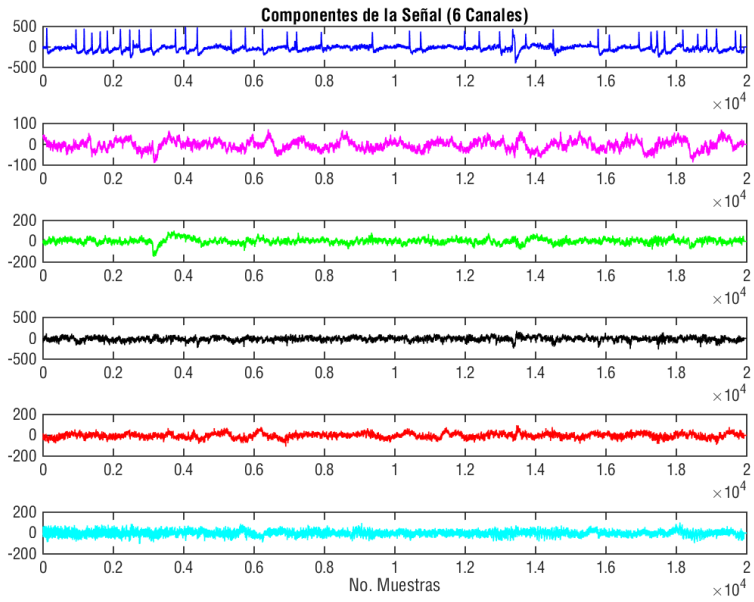
Análisis 64 Canales:

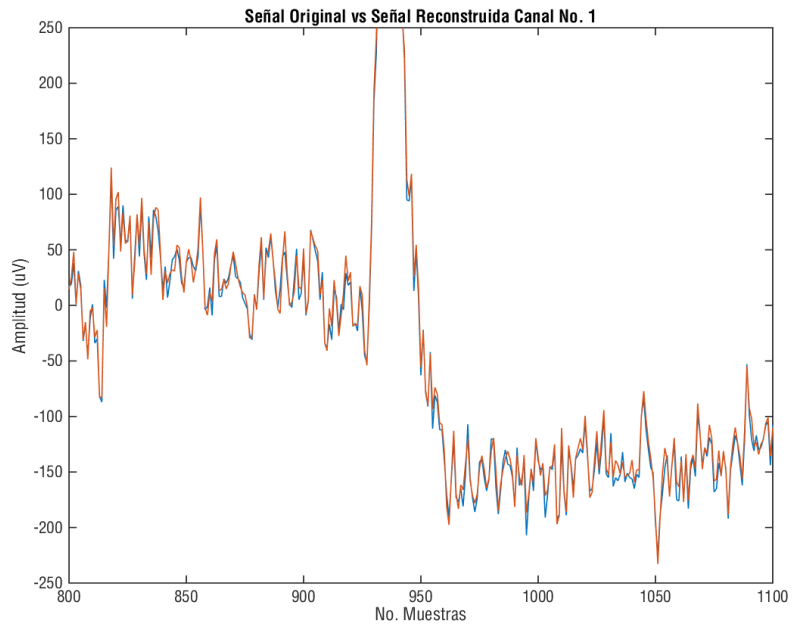
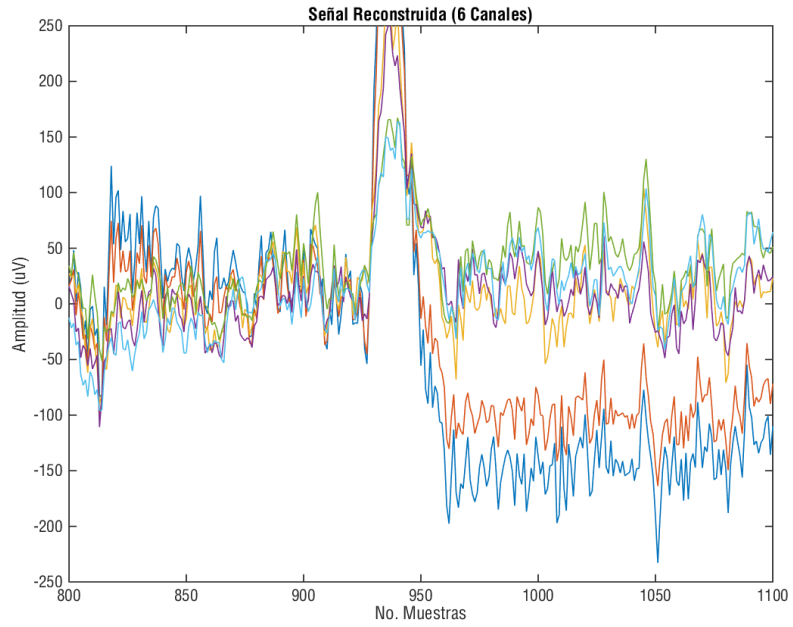


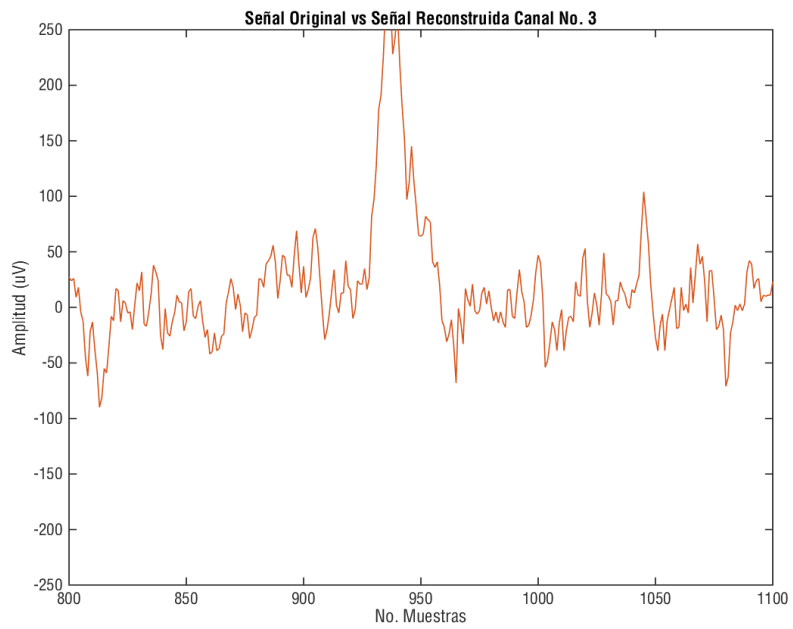
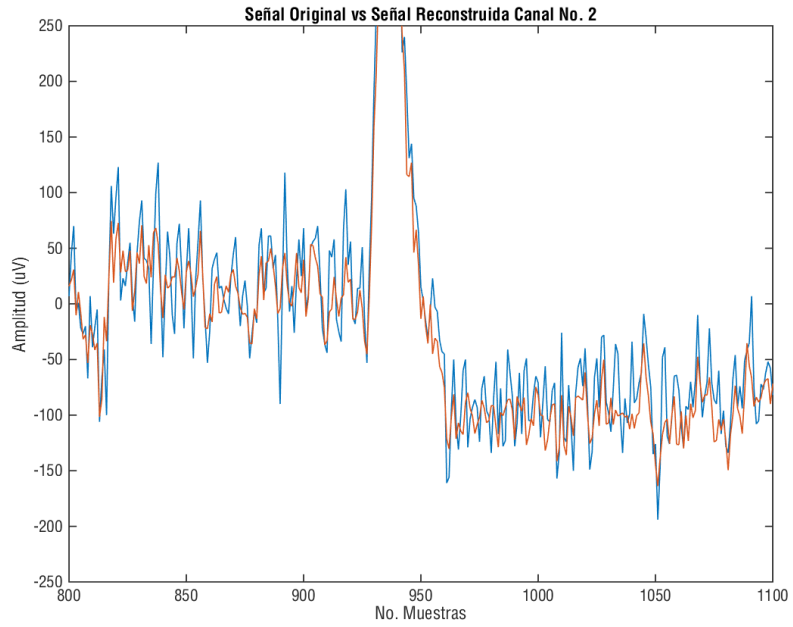


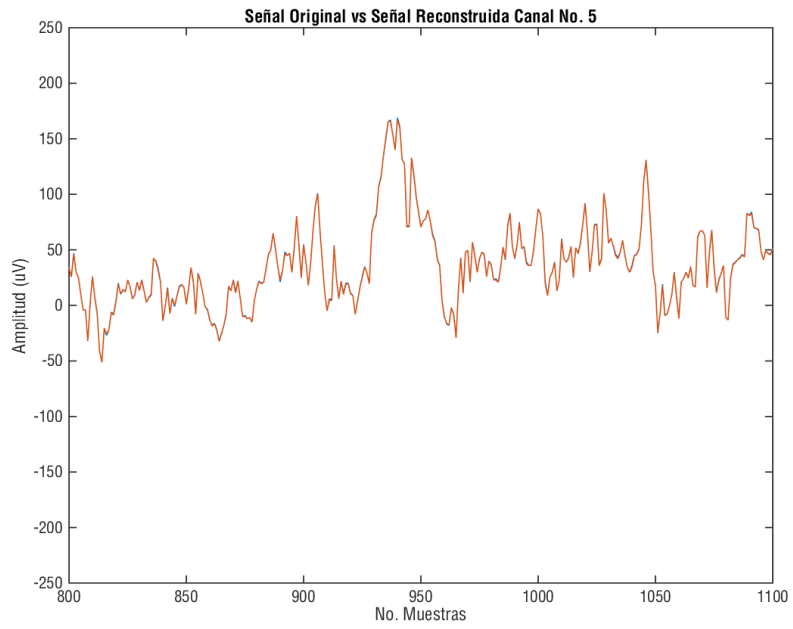
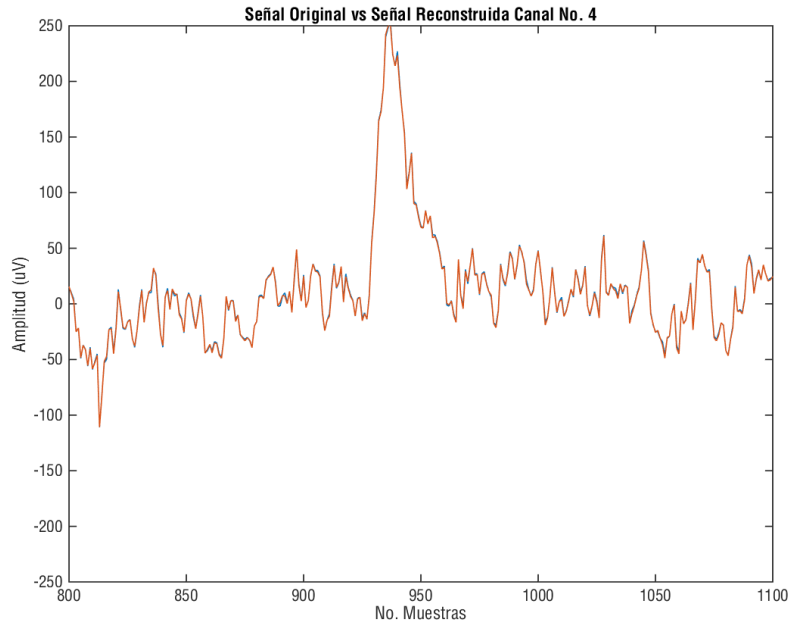
Análisis 6 Canales:

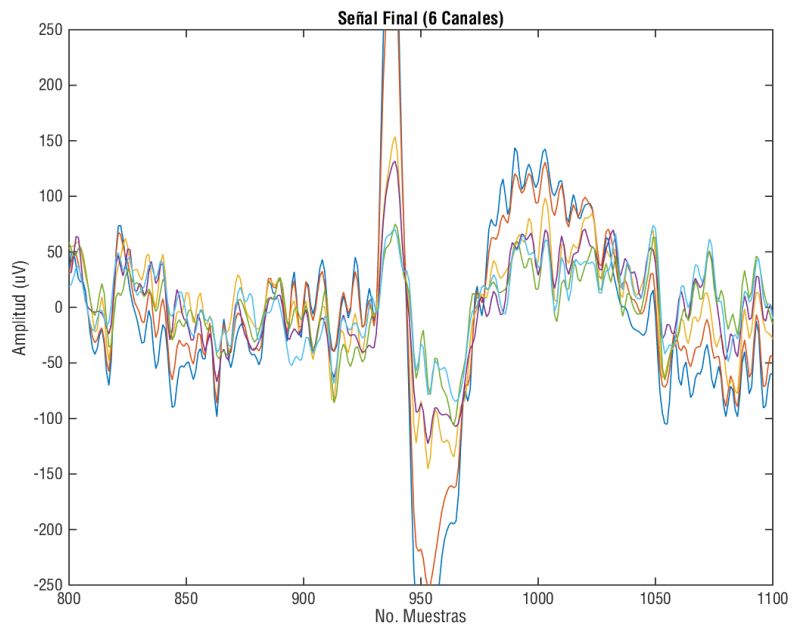
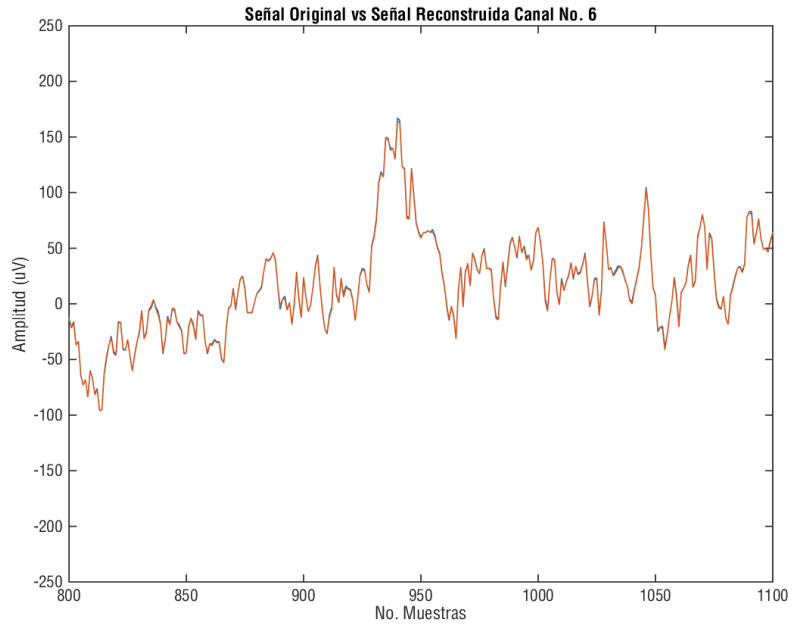






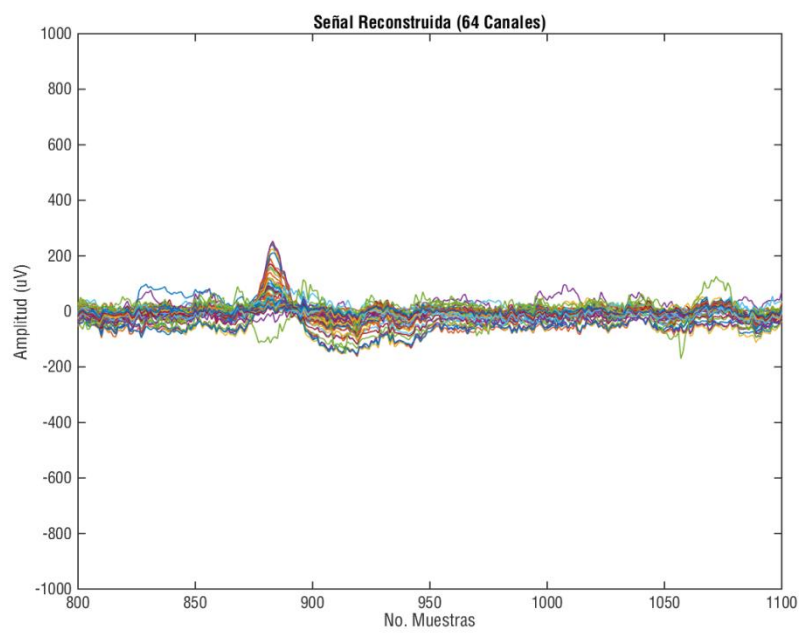
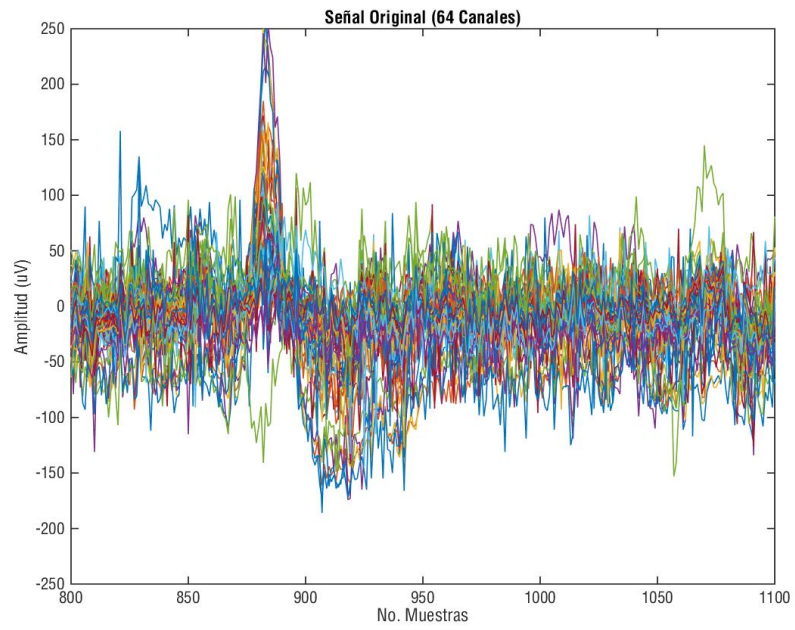


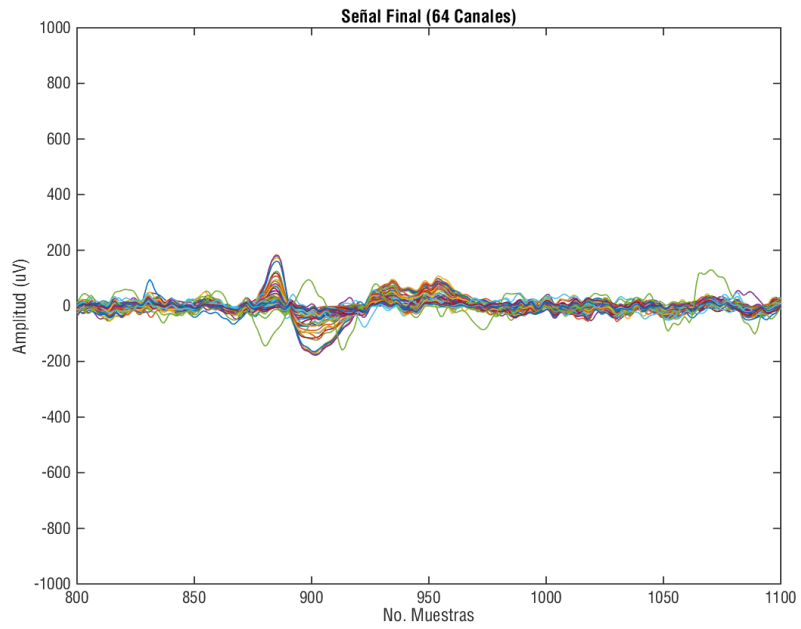




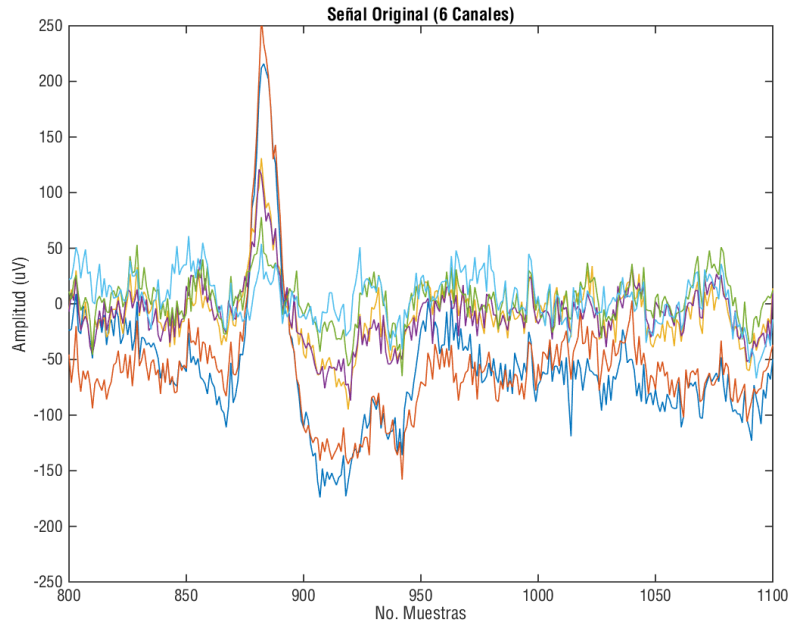
Registro Individuo número 85 tarea 5:

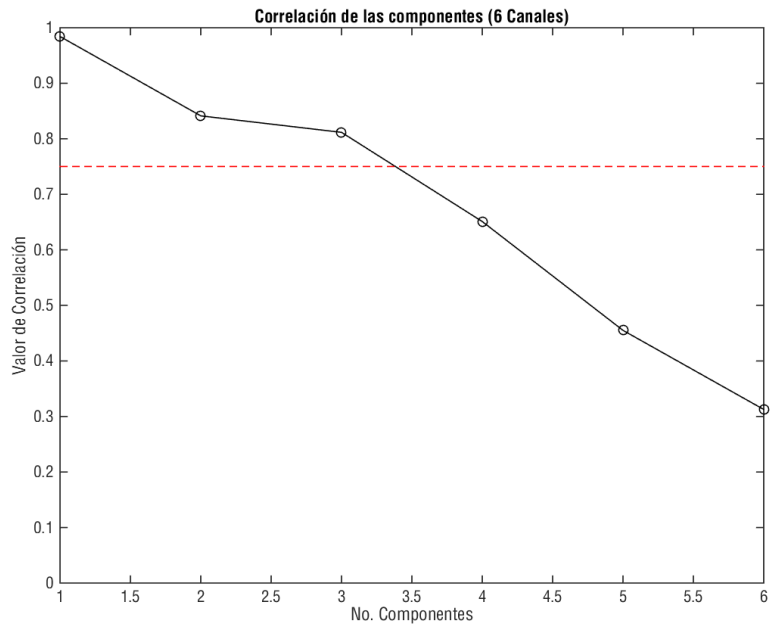
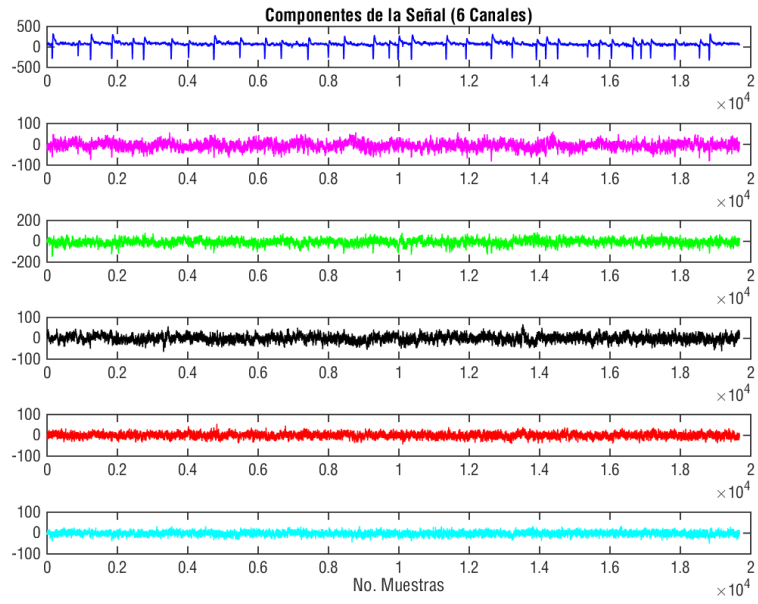
Análisis 64 Canales:

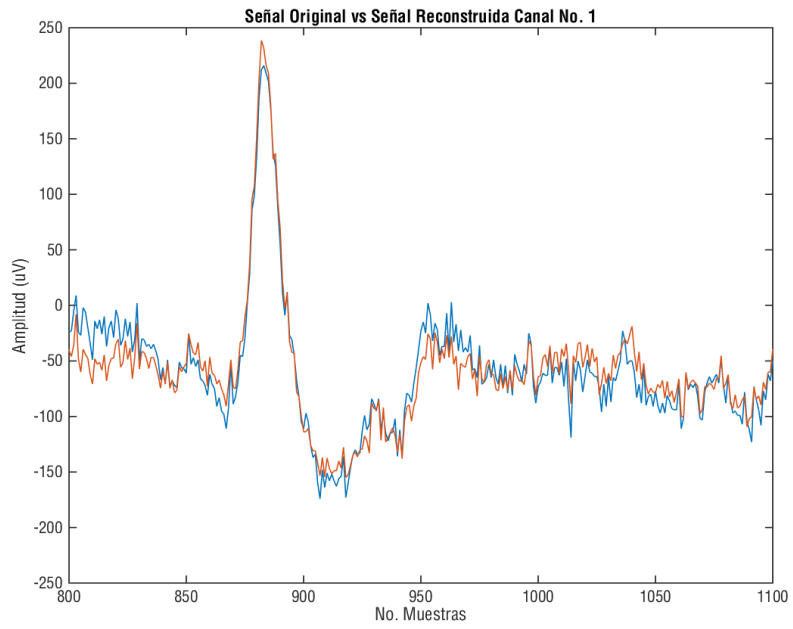
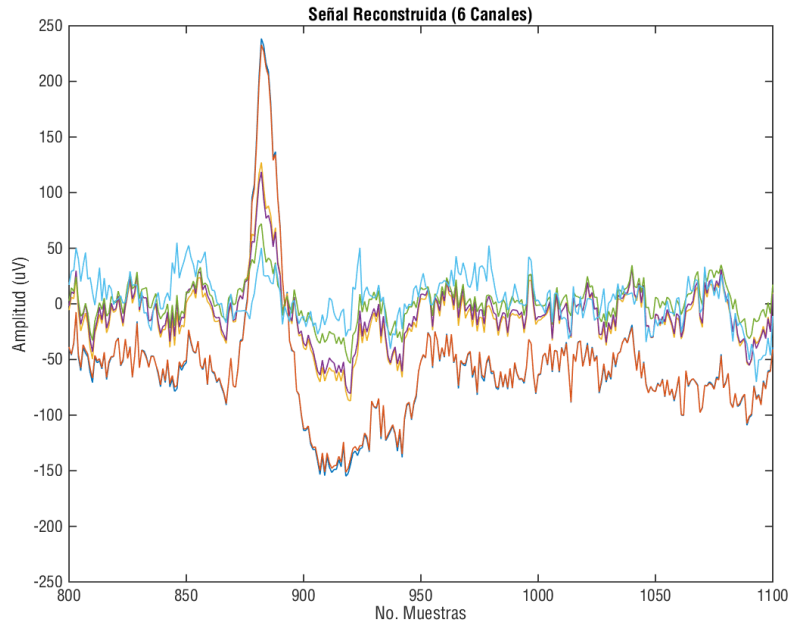


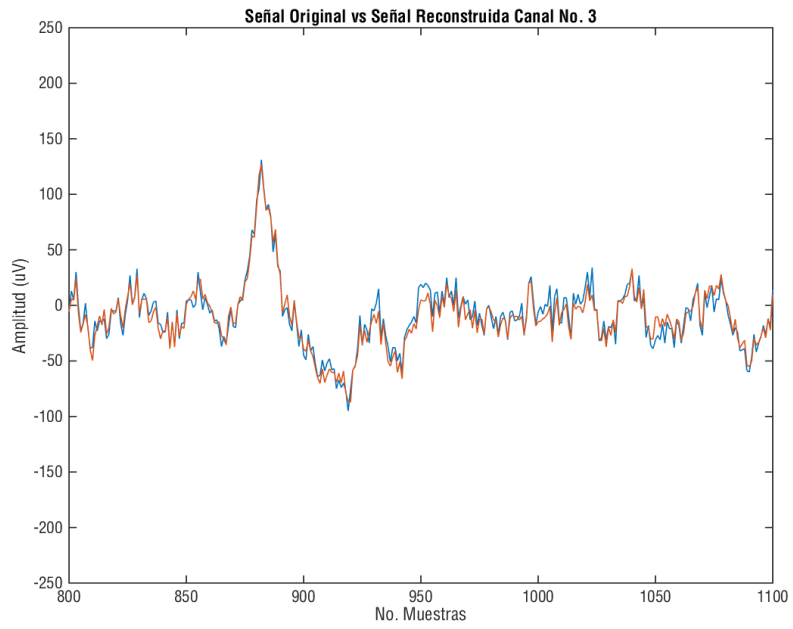
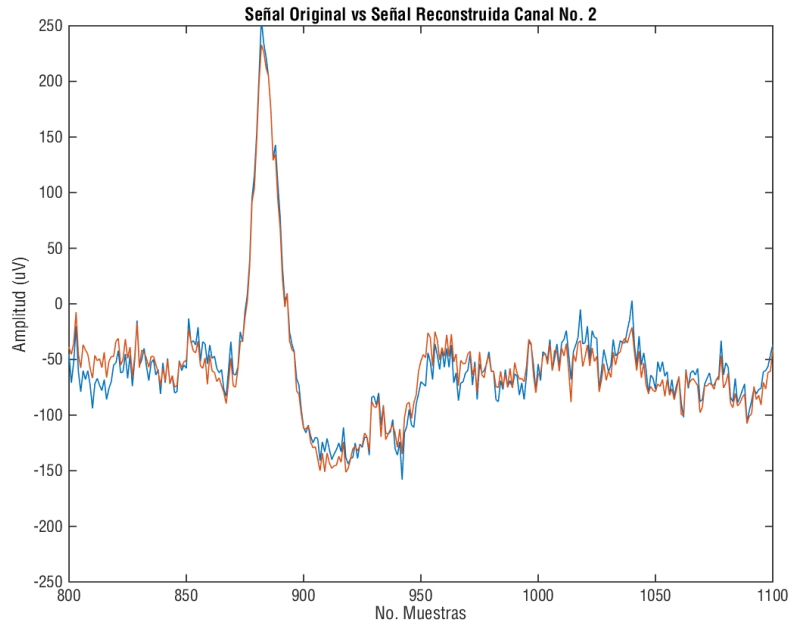


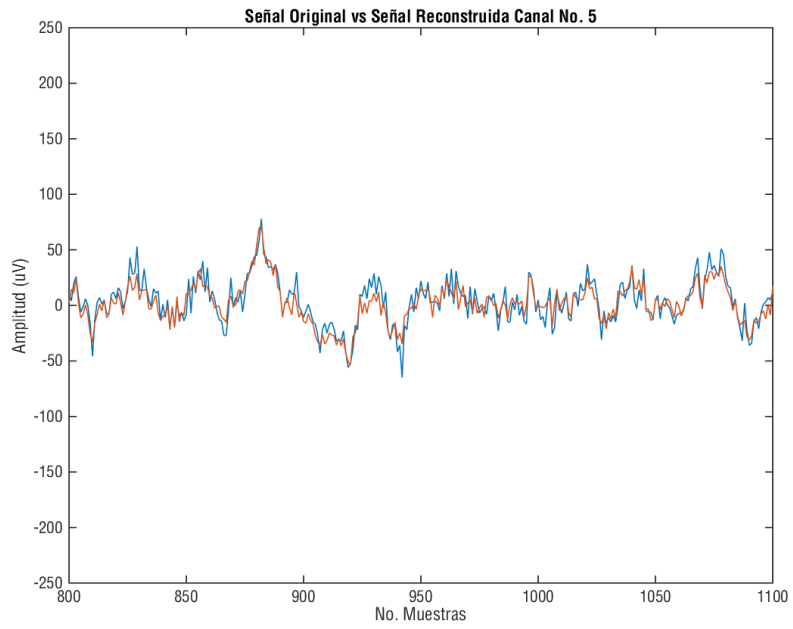
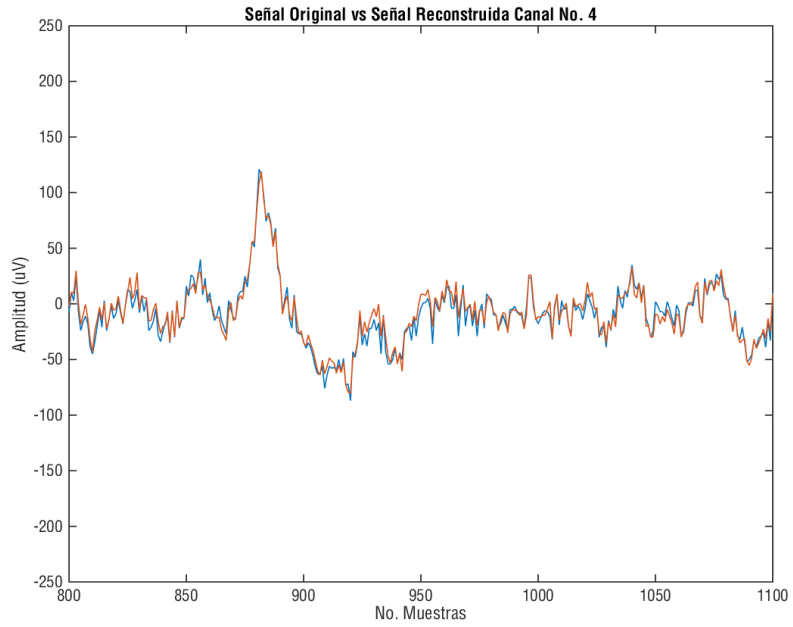
Análisis 6 Canales:

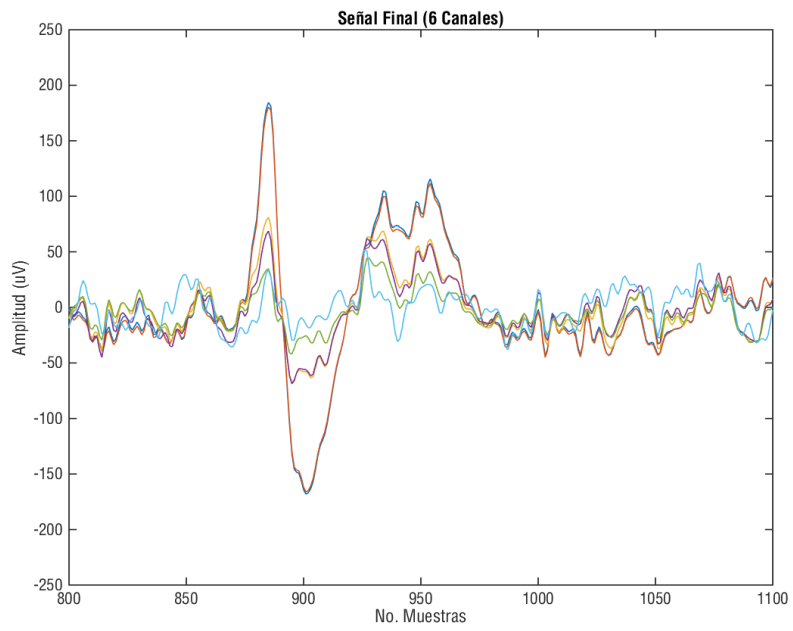
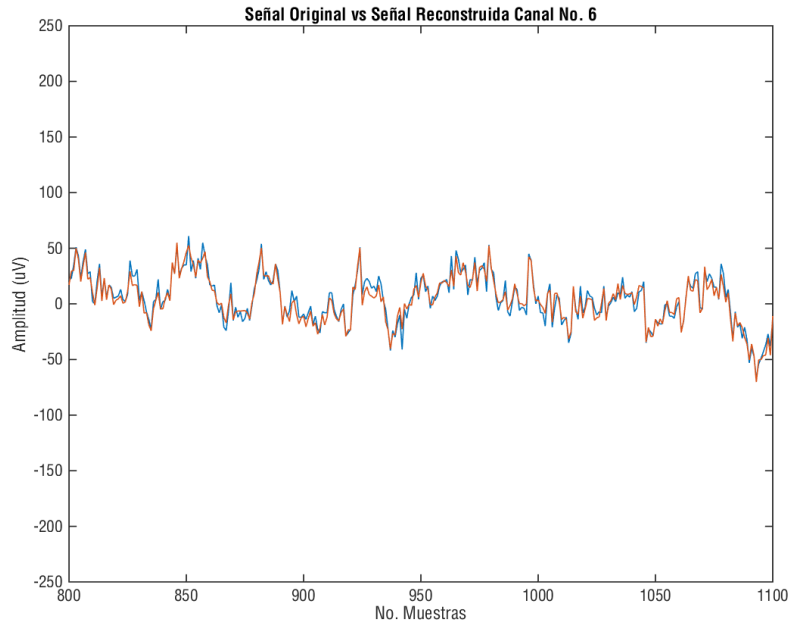






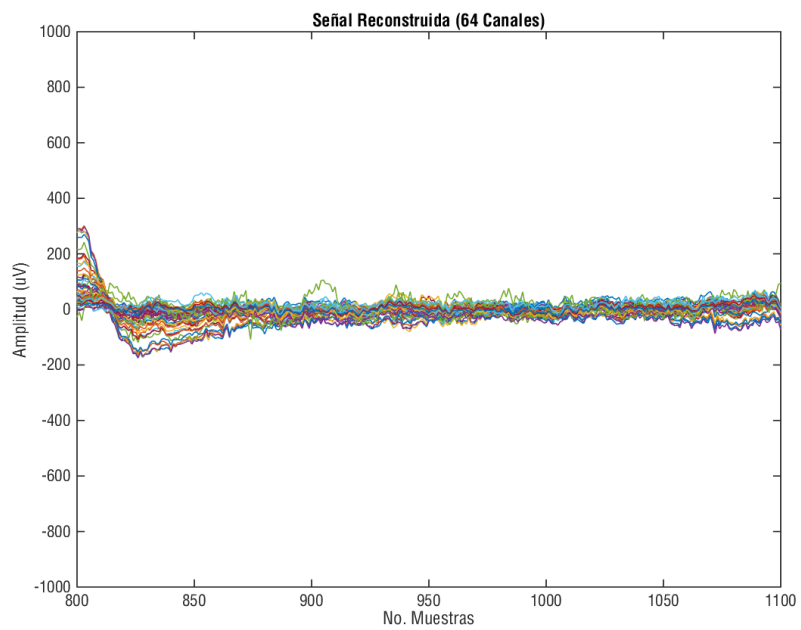
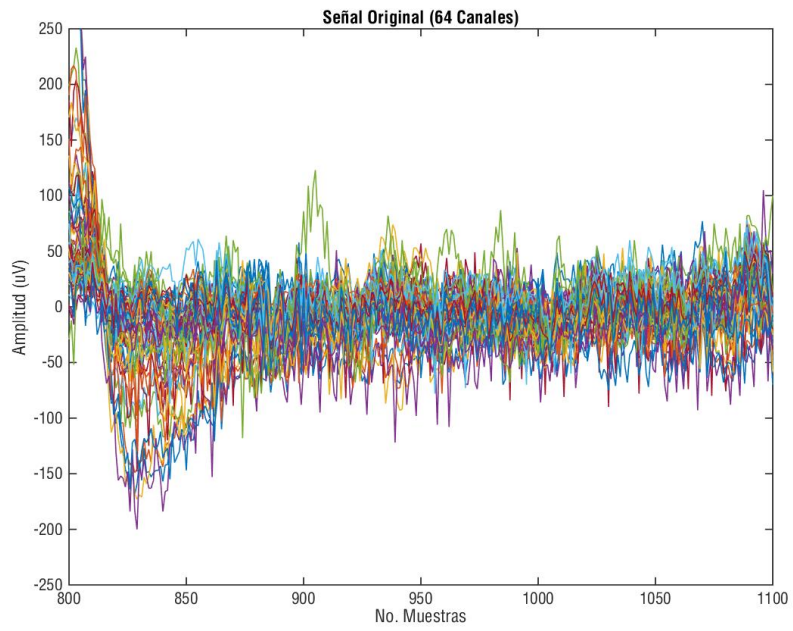


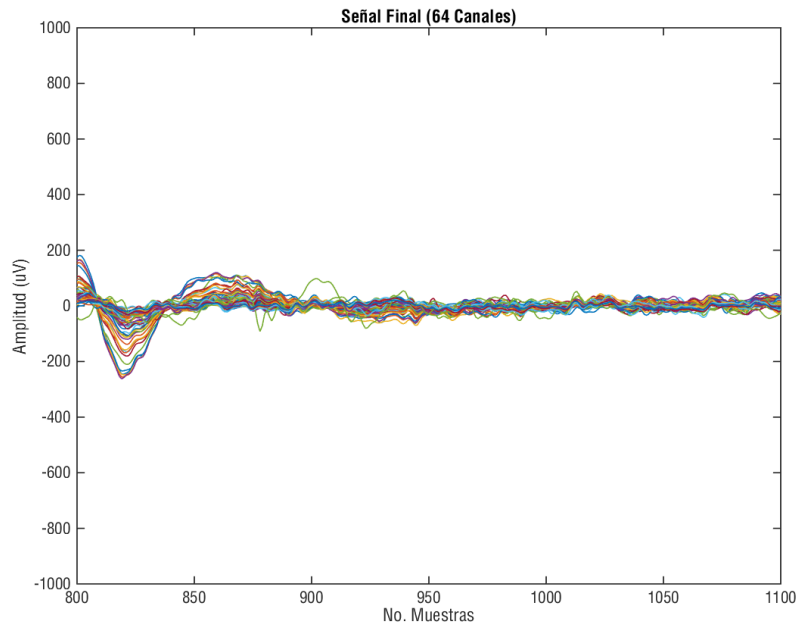




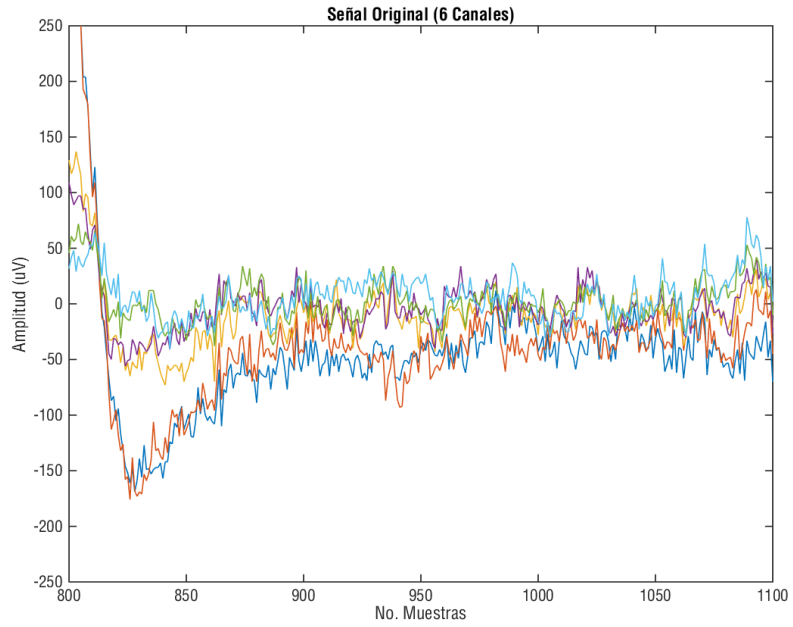
Registro Individuo número 85 tarea 9:

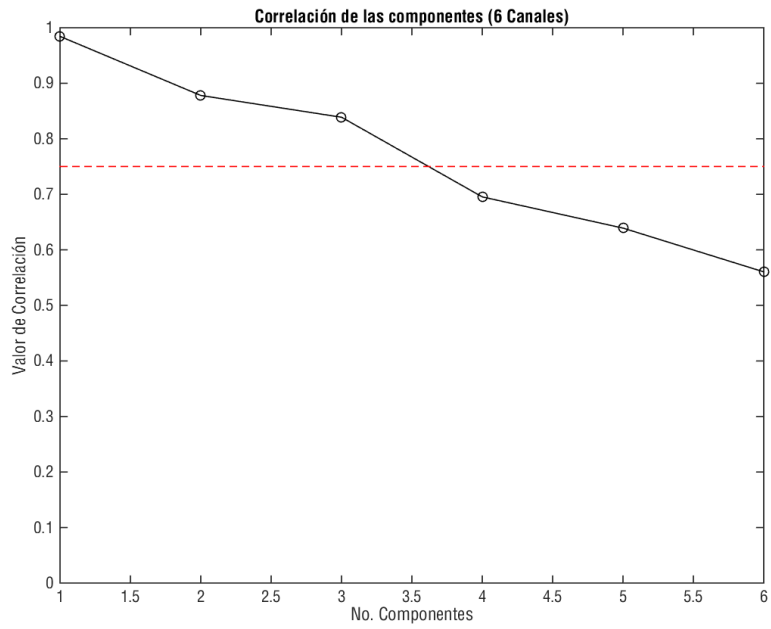
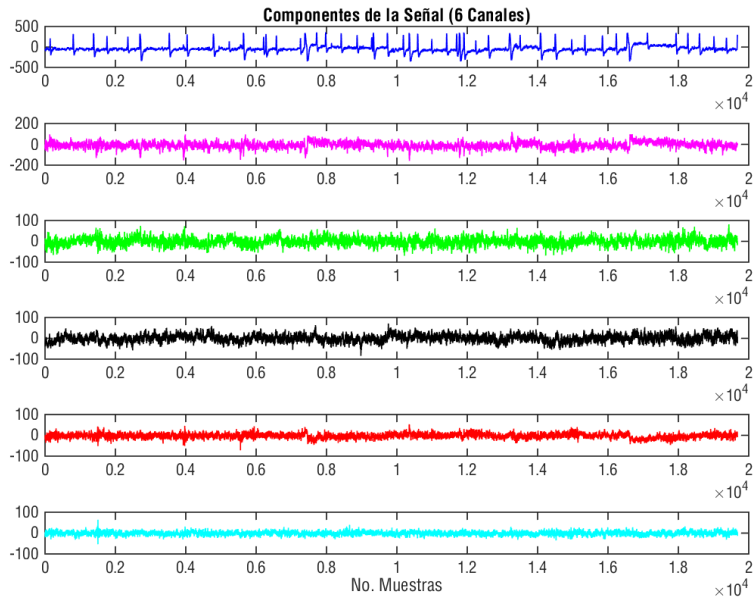
Análisis 64 Canales:

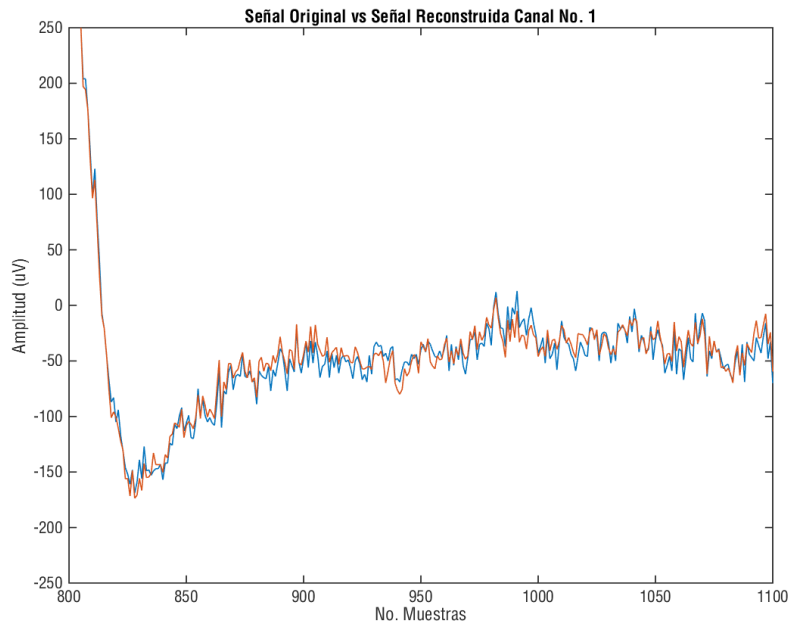
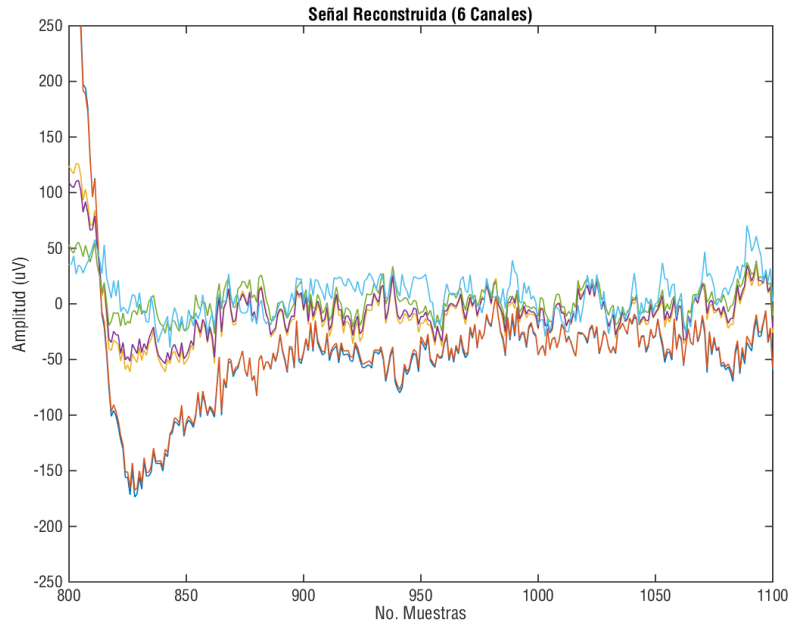


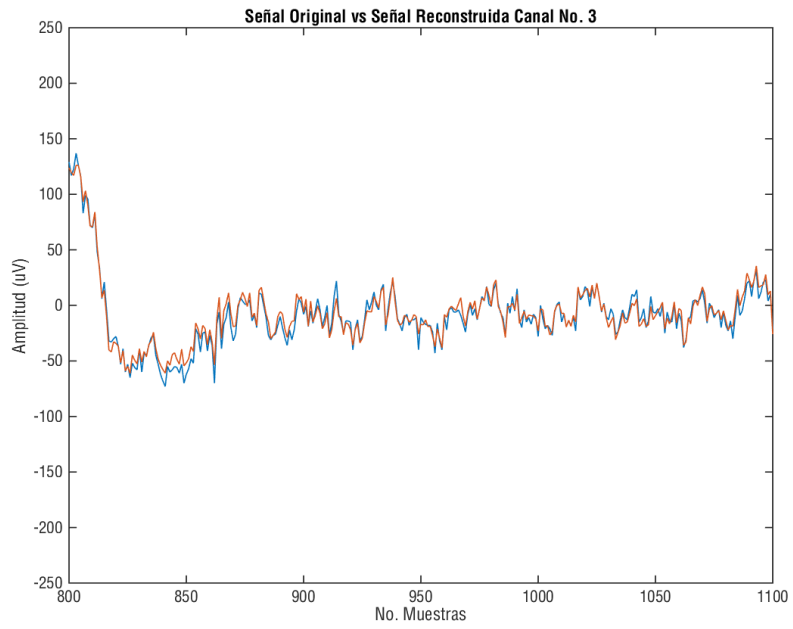
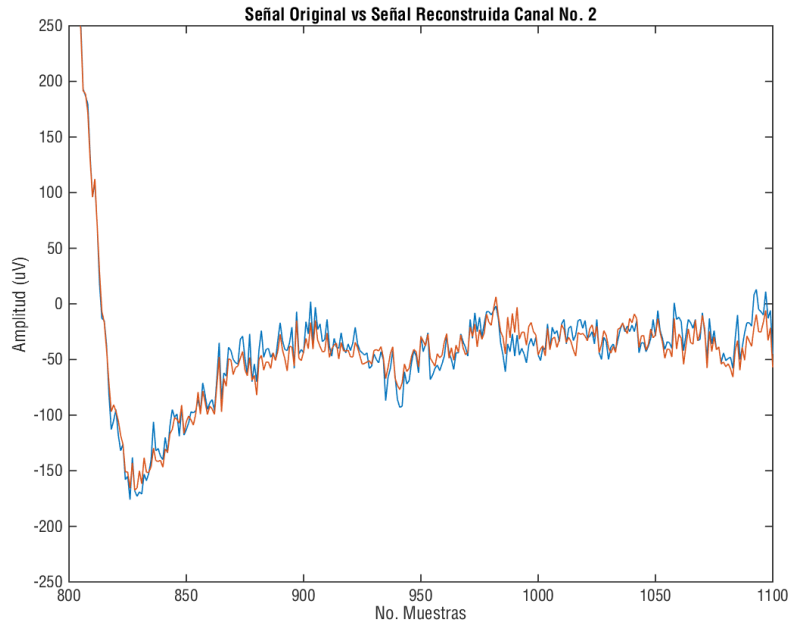


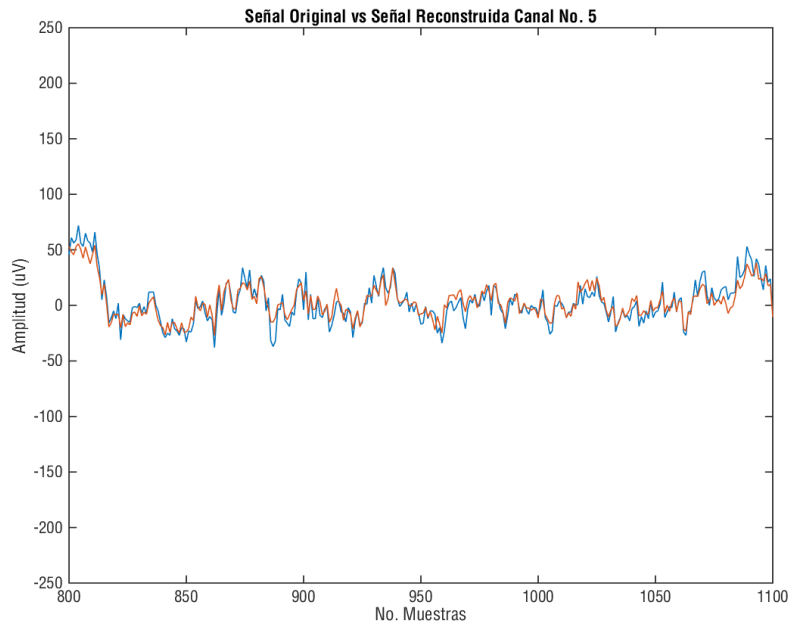
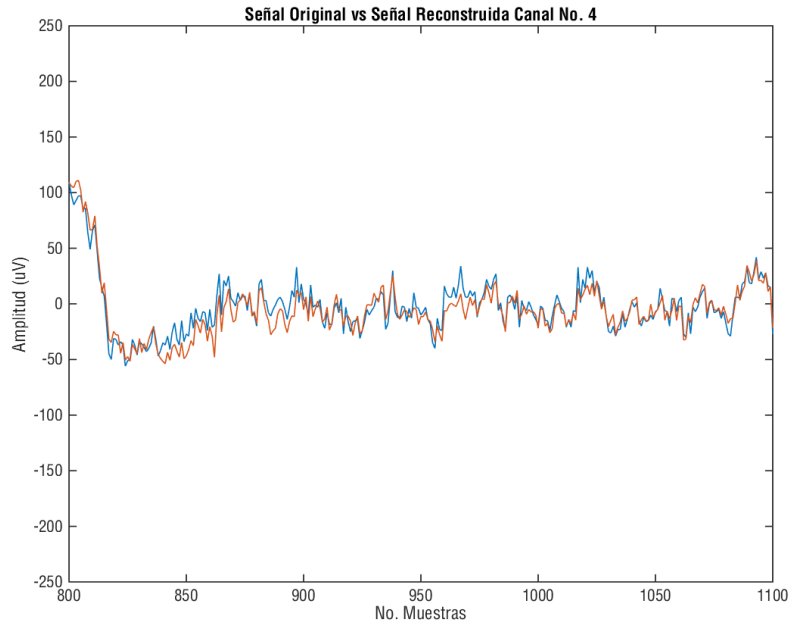
Análisis 6 Canales:

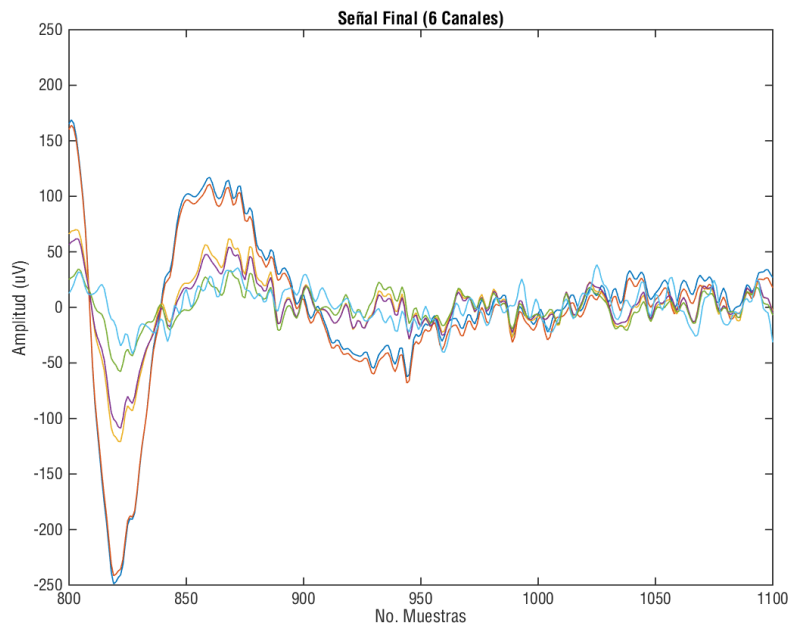
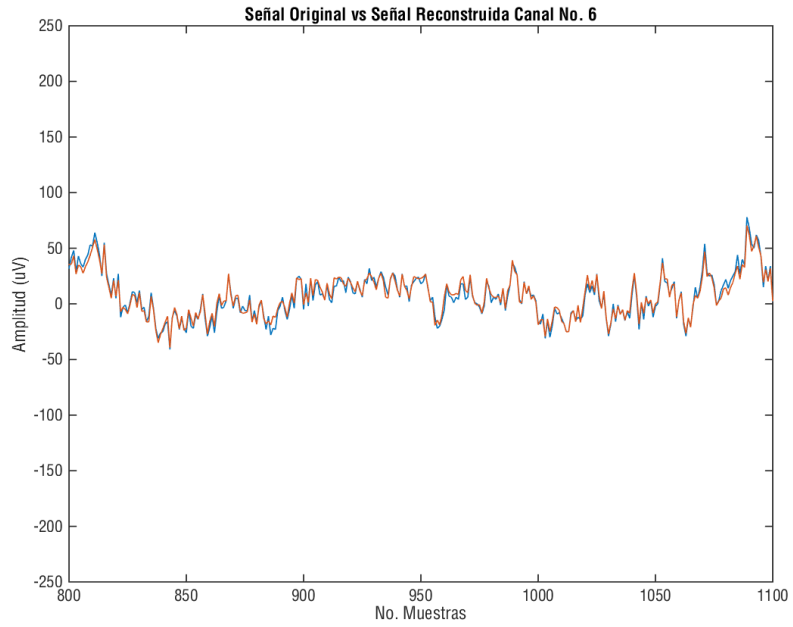






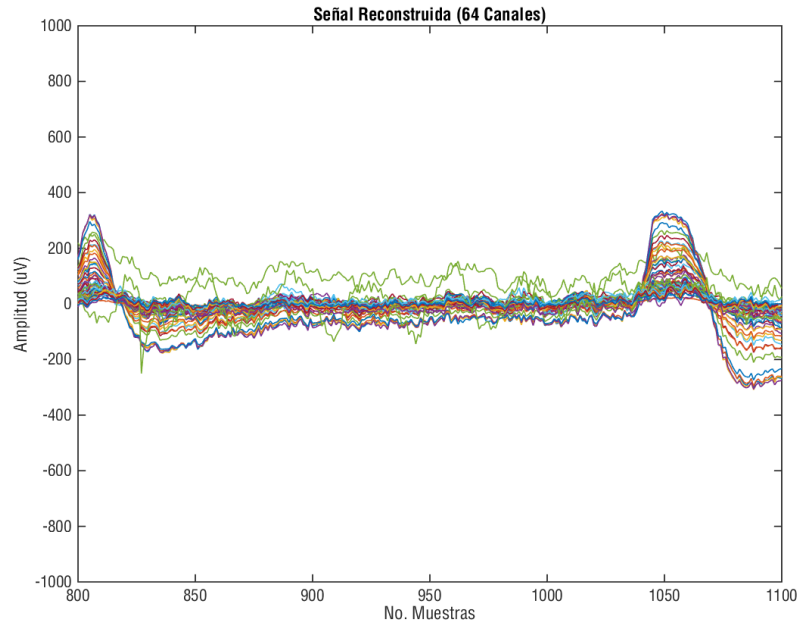
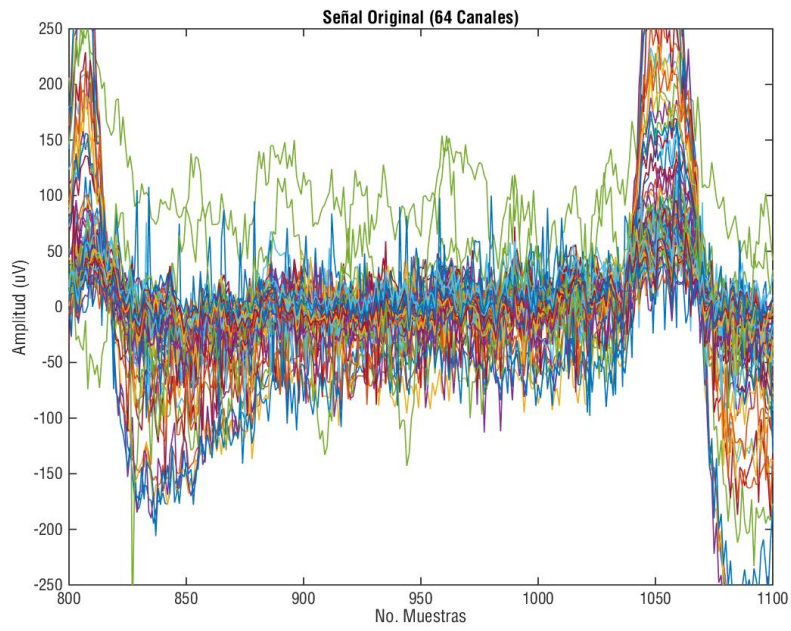


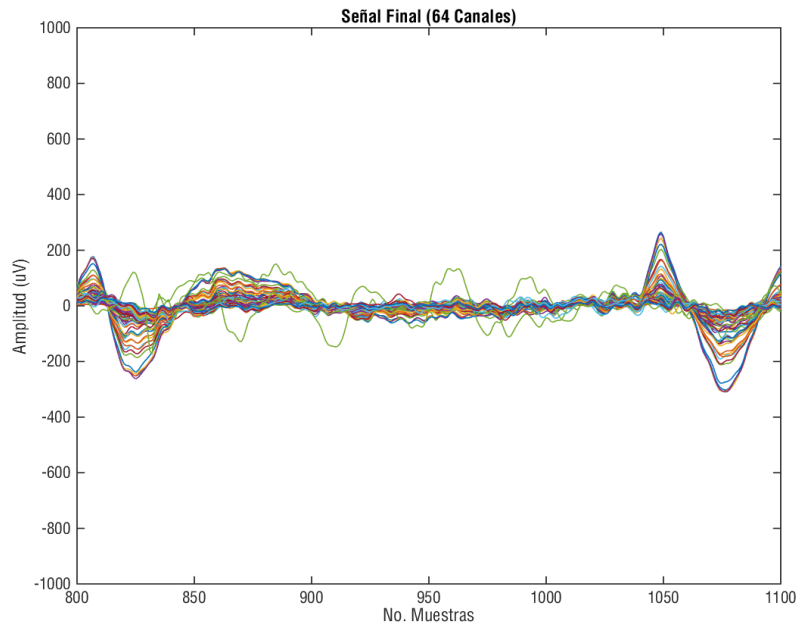




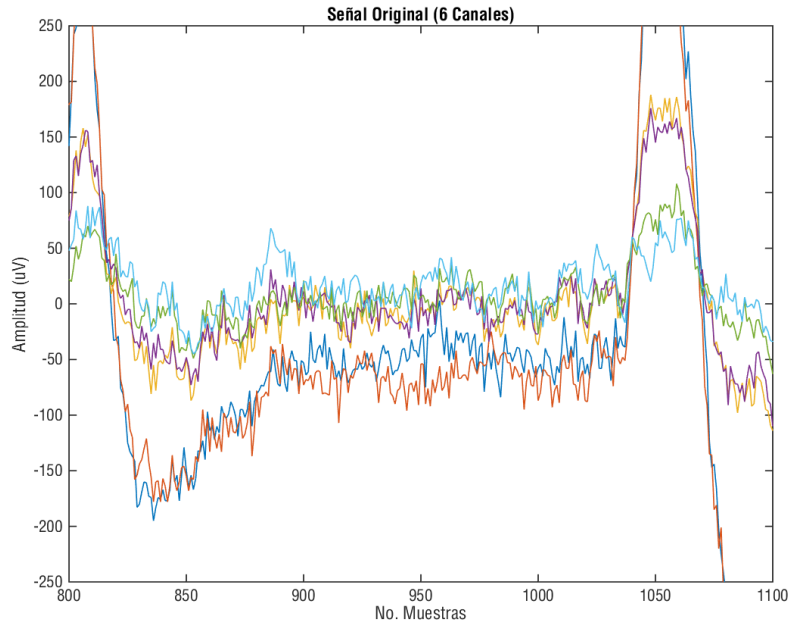
Registro Individuo número 85 tarea 13:

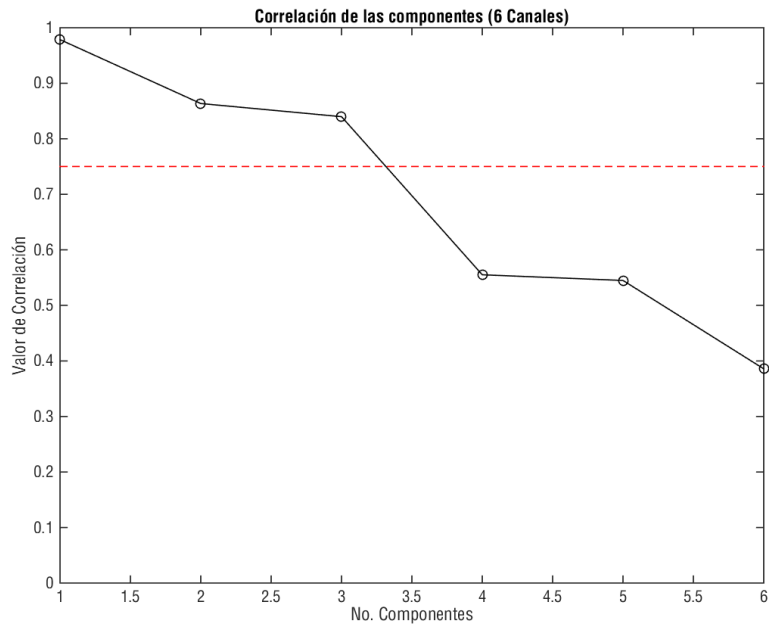
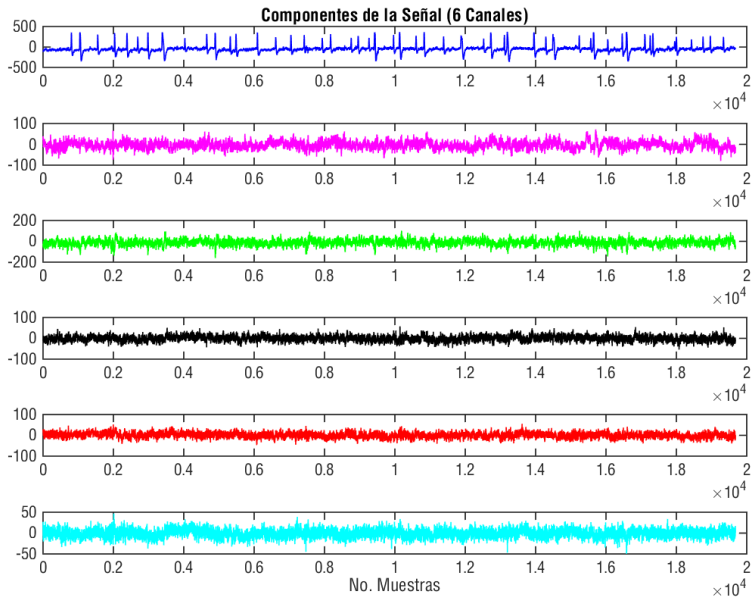
Análisis 64 Canales:

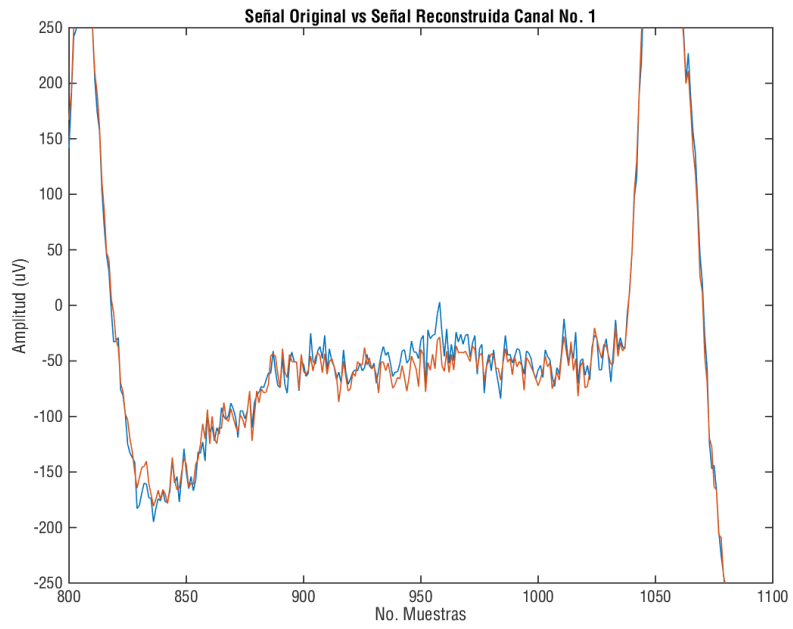
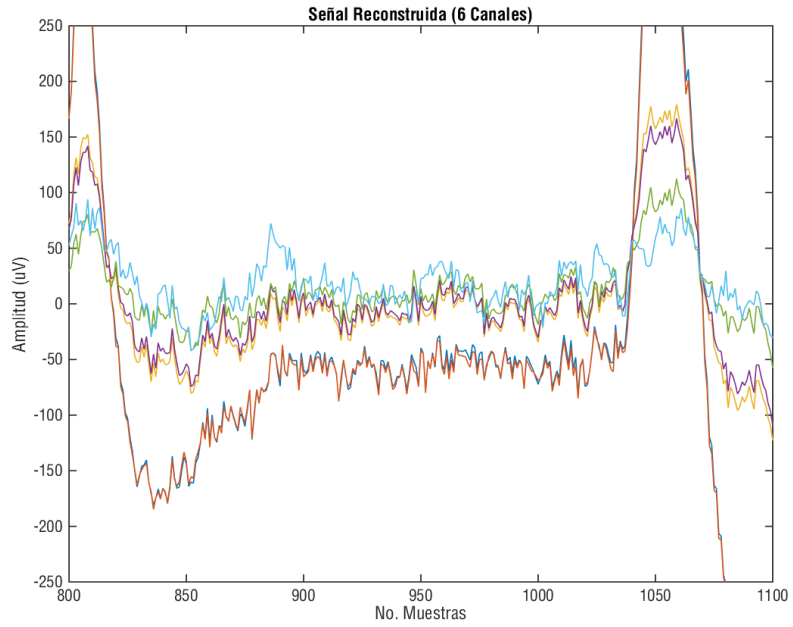


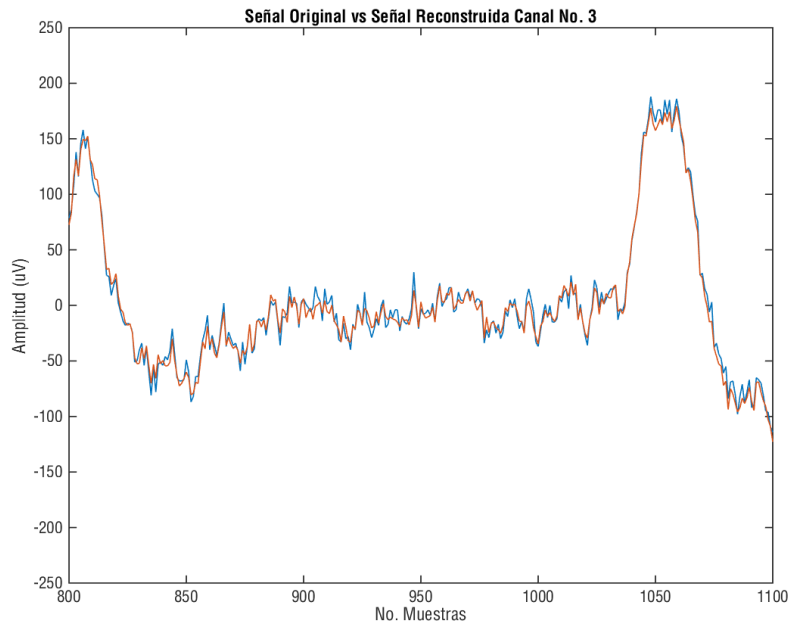
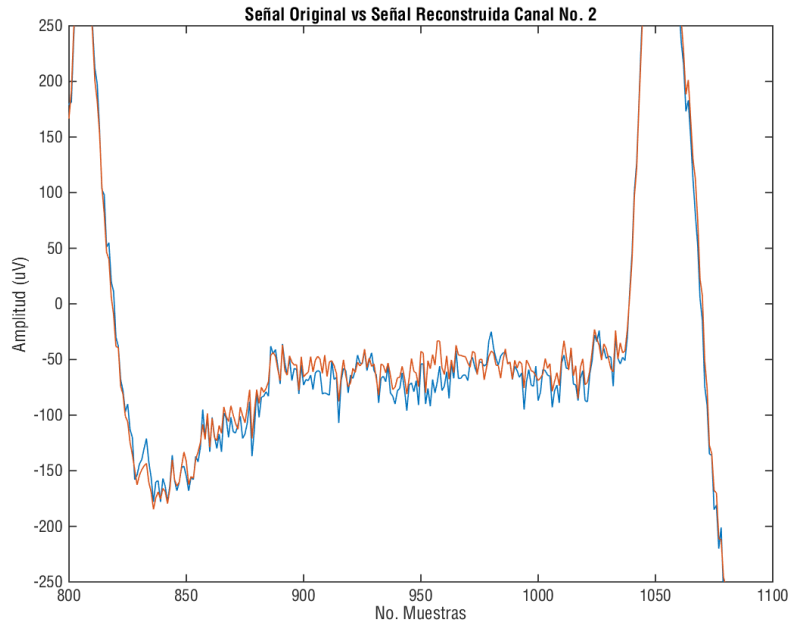


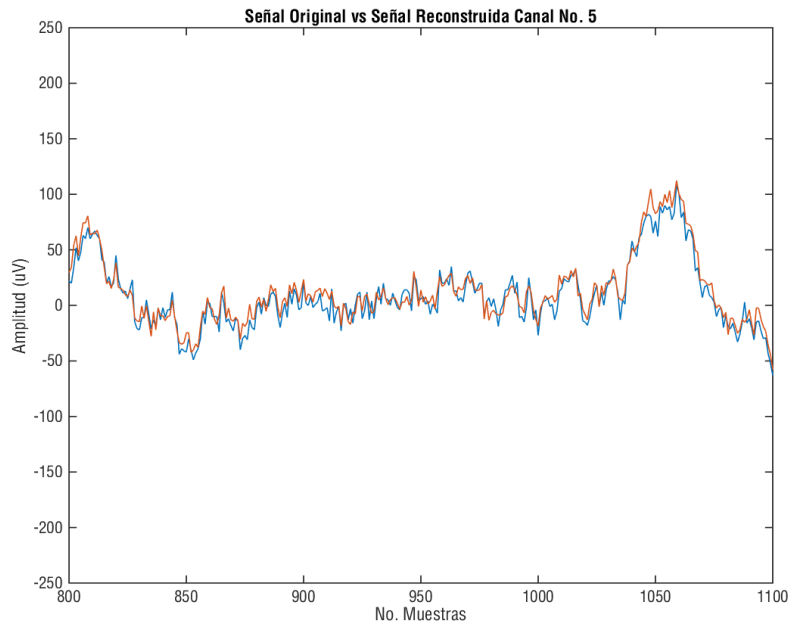
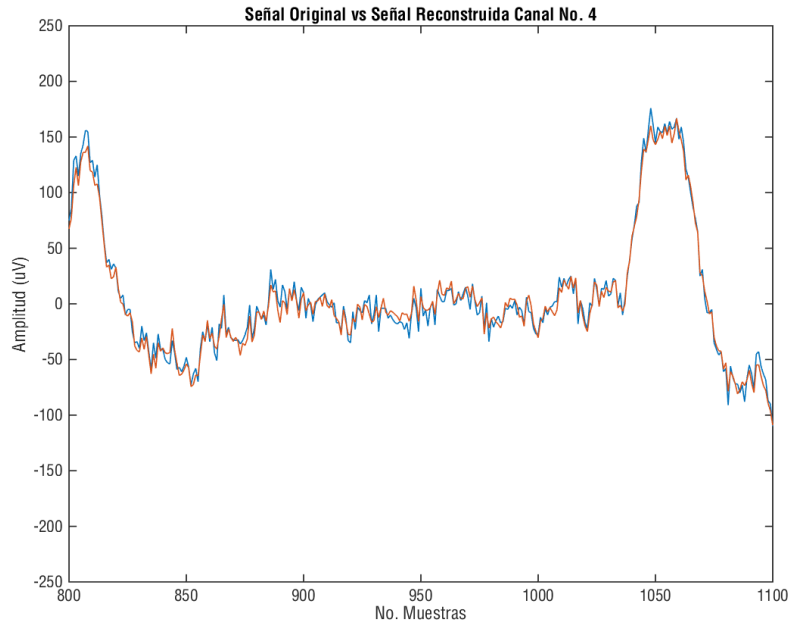
Análisis 6 Canales:

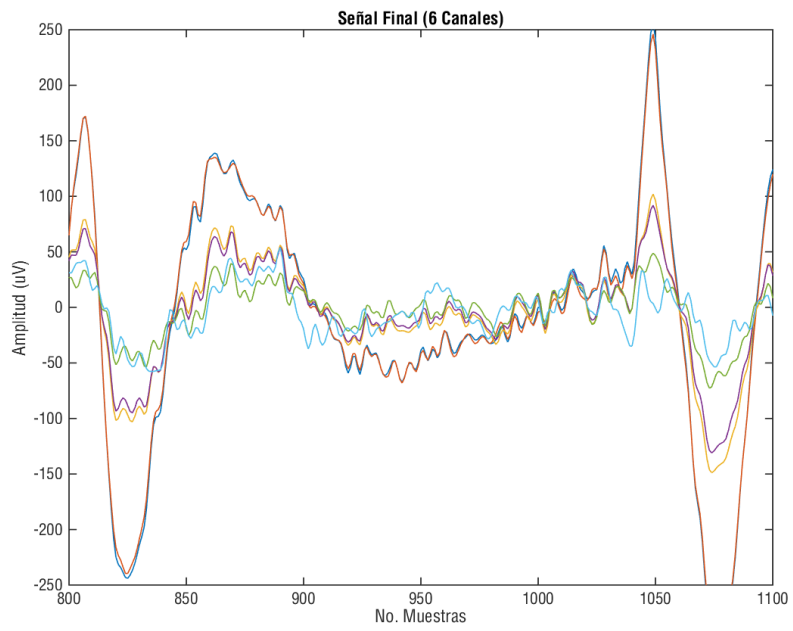
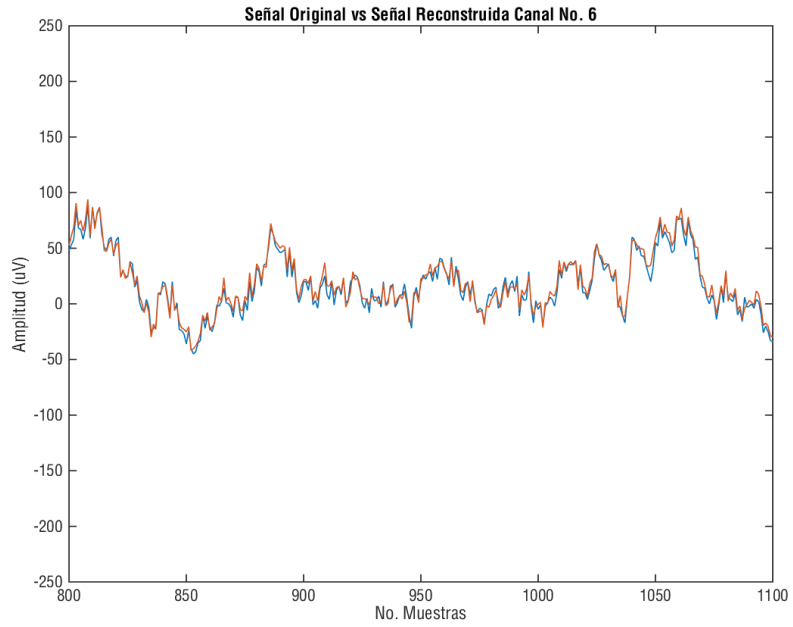












ANEXO 5

PROCESAMIENTO DE SEÑALES ADQUIRIDAS POR LA TARJETA DE EVALUACIÓN EN MATLAB®

```
clear all, close all, clc
%% PROCESAMIENTO DE LAS SEÑALES DE LAS SEÑALES ADQUIRIDAS CON LA TARJETA
% DE EVALUACIÓN ADS1299

x=xlsread('Resultados.xlsx');
X=x;
figure, plot(X)
%axis([1000 3000 -0.1 0.1])
title('Señal Original (6 Canales)')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F1')

%% DESCOMPOSICIÓN DE LA SEÑAL MÉTODO BSS - CCA:

% Descomposición de la señal por el método BSS-CCA a través del algoritmo
% que se encuentra en el Anexo 2.

[Wc,rc] = bsscca(X',1) % El retraso del método se
                        % sugiere en 1. La matriz Wc es
                        % la matriz de desmezcla y el
                        % vector rc es el vector de
                        % correlaciones.

Yc=Wc*X';

% Gráfica de las componentes junto con el vector de correlaciones.
figure
subplot 611, plot(Yc(1,:), 'b')
title('Componentes de la Señal (6 Canales)')
subplot 612, plot(Yc(2,:), 'm')
subplot 613, plot(Yc(3,:), 'g')
subplot 614, plot(Yc(4,:), 'k')
subplot 615, plot(Yc(5,:), 'r')
subplot 616, plot(Yc(6,:), 'c')
xlabel('No. Muestras');
print(gcf, '-dpng', 'F2')

figure, plot(rc, '-ok')
line=[1 2 3 4 5 6];
liney=[0.75 0.75 0.75 0.75 0.75 0.75];
hold on
plot(line, liney, '--r')
axis([1 6 0 1])
title('Correlación de las componentes (6 Canales)')
xlabel('No. Componentes');
ylabel('Valor de Correlación');
print(gcf, '-dpng', 'F3')

% Se realiza una búsqueda dentro del vector de correlaciones y de acuerdo
% con lo relacionado al método BSS-CCA se eliminan las componentes que
% tengan una correlación menor a 0.75.

for i=1:length(rc)
    if rc(i)<= 0.75
        Yc(i,:)=0;
    end
end

% Proceso de reconstrucción de la señal después de eliminar componentes

Ac=inv(Wc);
Xc=Ac*Yc; % Al realizar esta operación
```

```

% se obtiene una señal Xc que
% es una estimación de la señal
% original con los artefactos
% eliminados.

Xc=Xc';
figure, plot(Xc)
% axis([4000 5000 -0.0005 0.001])
title('Señal Reconstruida (6 Canales)')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F4')

%% COMPARACIÓN ENTRE LA SEÑAL ORIGINAL Y LA RECONSTRUIDA
figure, plot (X(:,1))
% axis([800 1100 -250 250])
hold on
plot (Xc(:,1))
title('Señal Original vs Señal Reconstruida Canal No. 1')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F5')

figure, plot (X(:,2))
% axis([800 1100 -250 250])
hold on
plot (Xc(:,2))
title('Señal Original vs Señal Reconstruida Canal No. 2')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F6')

figure, plot (X(:,3))
% axis([800 1100 -250 250])
hold on
plot (Xc(:,3))
title('Señal Original vs Señal Reconstruida Canal No. 3')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F7')

figure, plot (X(:,4))
% axis([800 1100 -250 250])
hold on
plot (Xc(:,4))
title('Señal Original vs Señal Reconstruida Canal No. 4')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F8')

figure, plot (X(:,5))
% axis([800 1100 -250 250])
hold on
plot (Xc(:,5))
title('Señal Original vs Señal Reconstruida Canal No. 5')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F9')

figure, plot (X(:,6))
% axis([4000 5000 -0.039 0.039])
hold on
plot (Xc(:,6))
title('Señal Original vs Señal Reconstruida Canal No. 6')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F10')

%% FILTRADO DE LA SEÑAL RECONSTRUIDA

```

```
Fs=160; % Frecuencia de muestreo de la
        % base de datos de Physionet.

% El filtro Butterworth Pasa Banda con frecuencias de corte de 0.5 - 35 Hz
% para hacer una ventana de las frecuencias cerebrales útiles.
Wn=[0.5 35]/(Fs/2);
[b,a]=butter(8,Wn,'bandpass');
sig_pb=filter(b,a,Xc); % Señal procesada sin
                       % artefactos utilizando el
                       % método BSS-CCA.

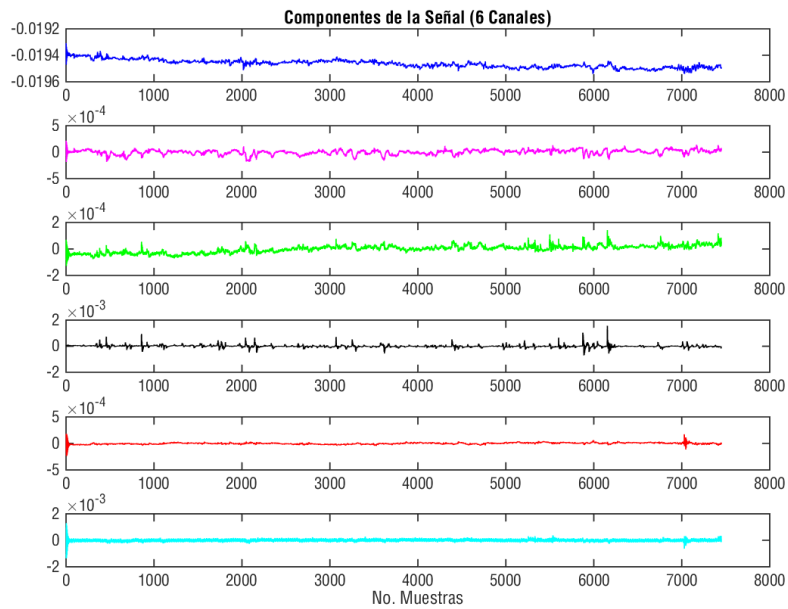
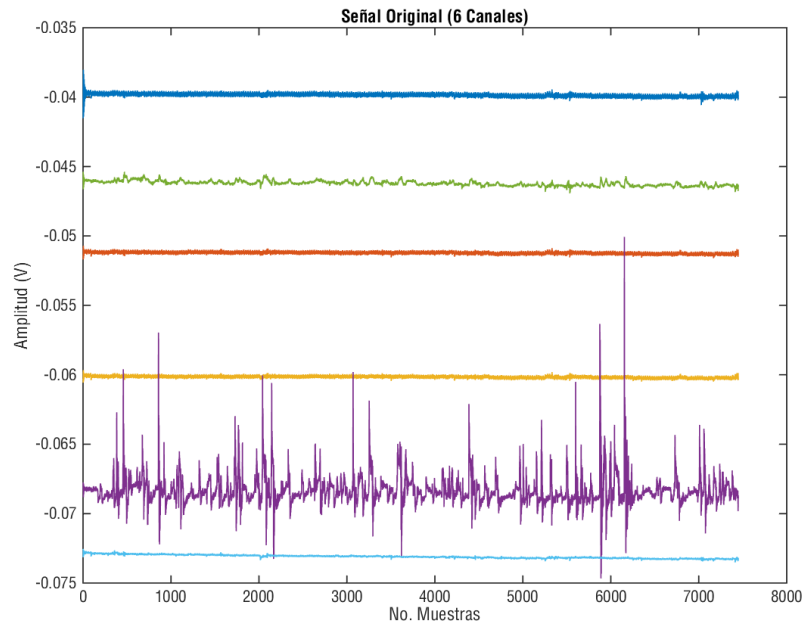
figure, plot(sig_pb)
axis([4000 5000 -0.0005 0.0005])
title('Señal Final (6 Canales)')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F11')

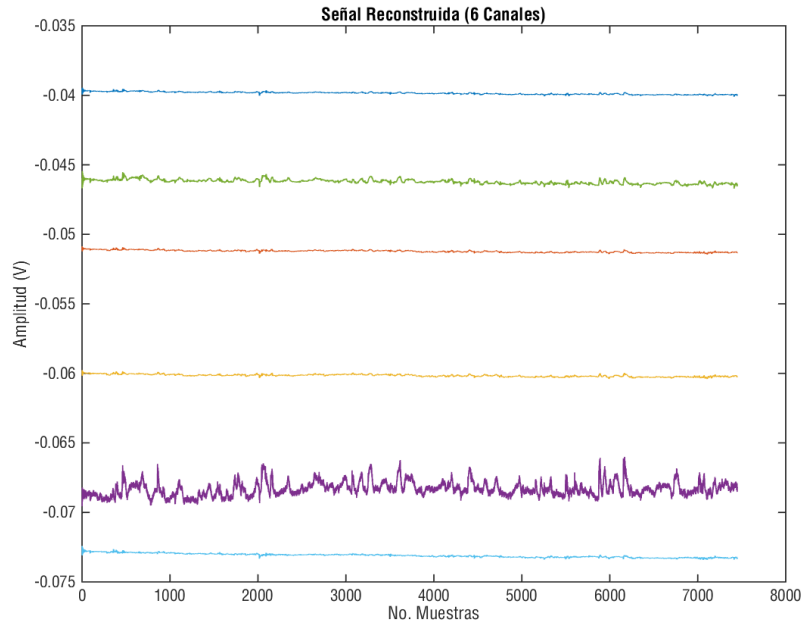
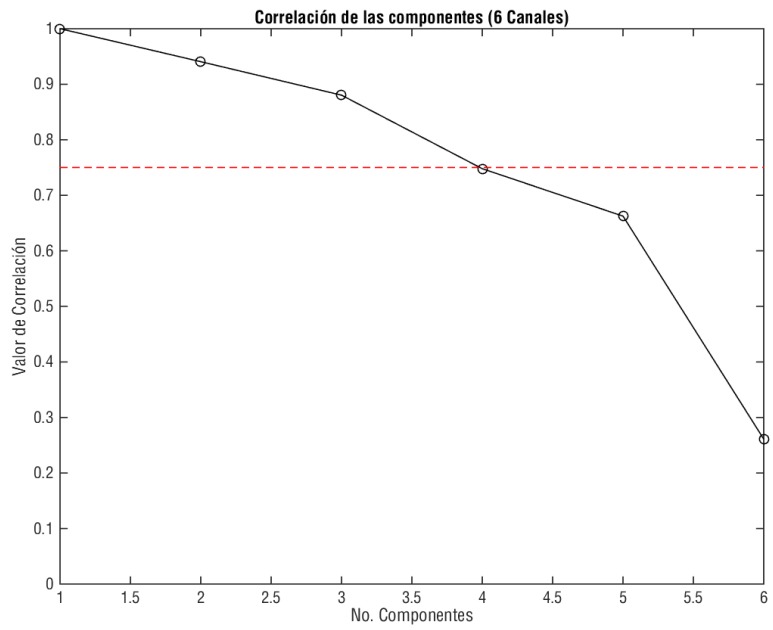
%% FIN DEL CÓDIGO
```

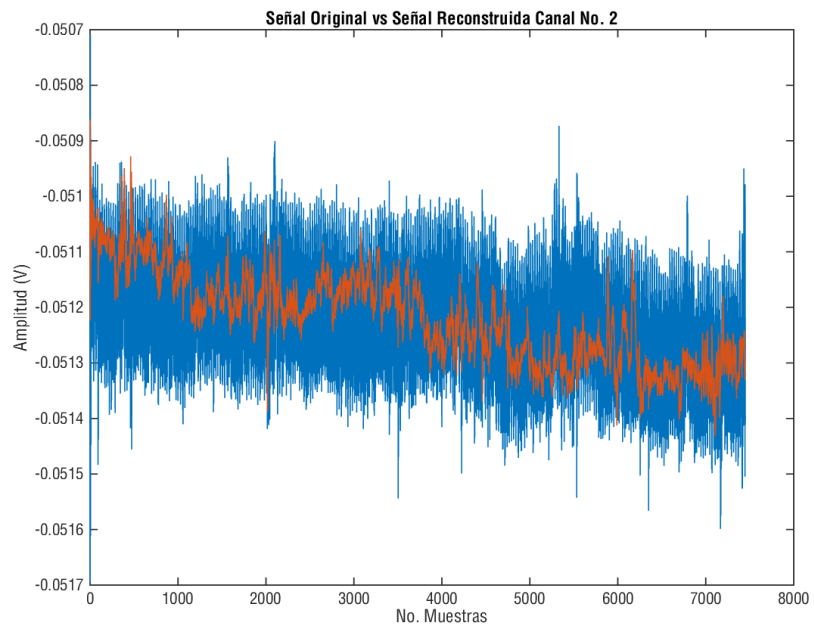
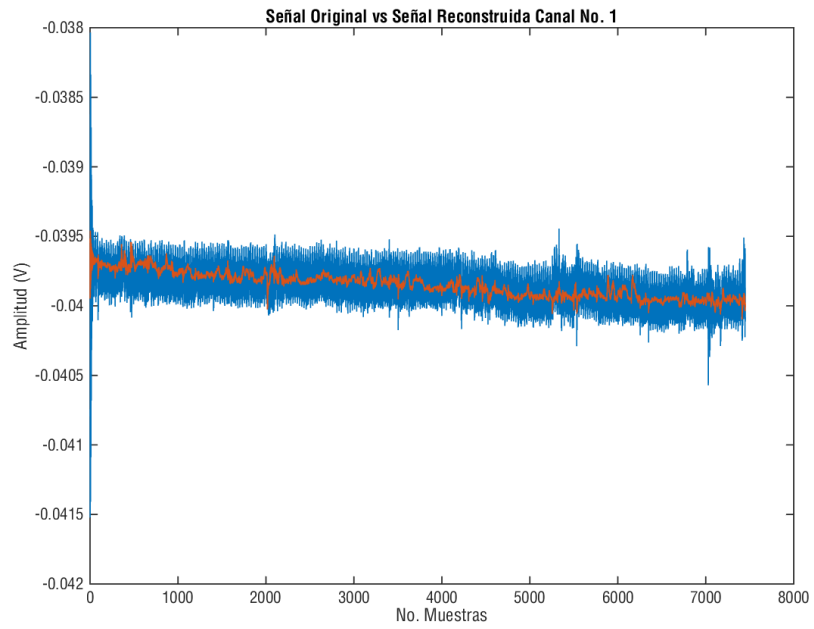
ANEXO 6.

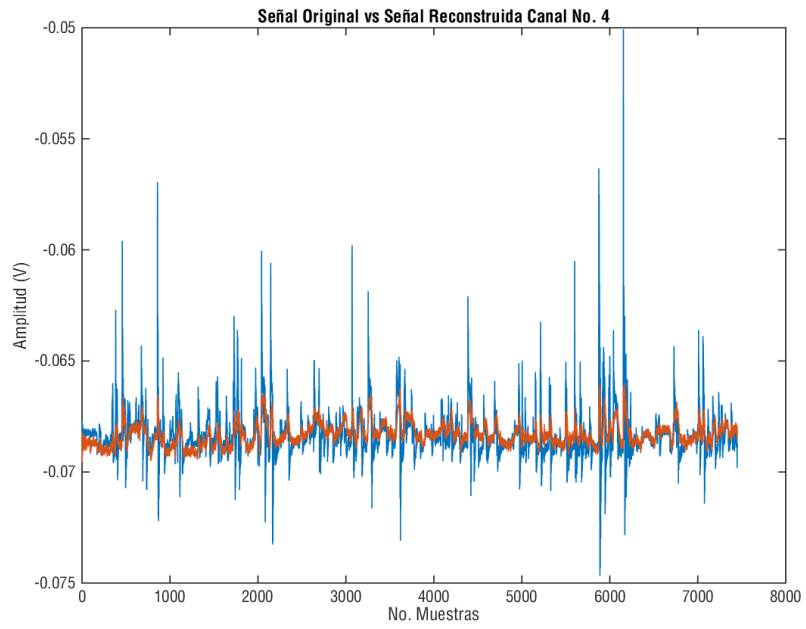
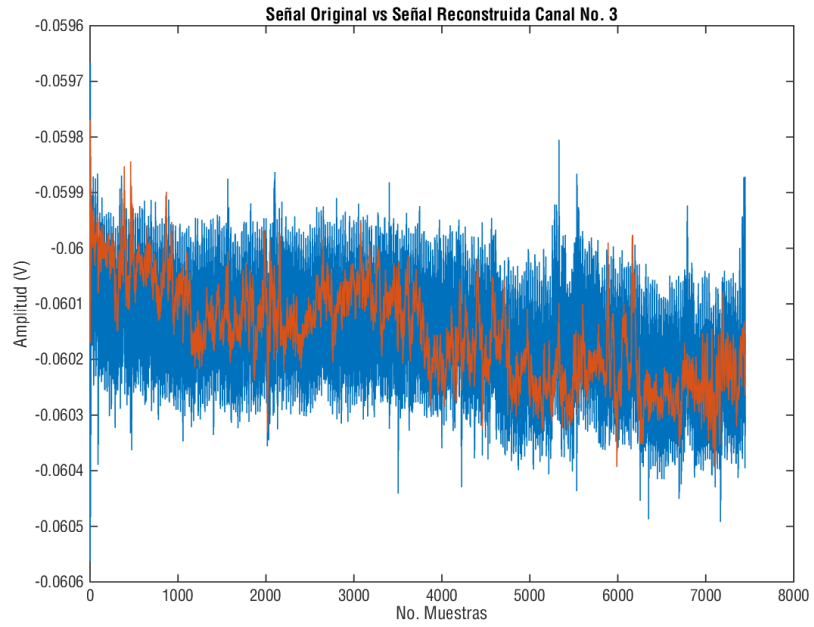
RESULTADOS DE LA IMPLEMENTACIÓN DEL PROCESAMIENTO A LAS SEÑALES ADQUIRIDAS POR LA TARJETA DE EVALUACIÓN.

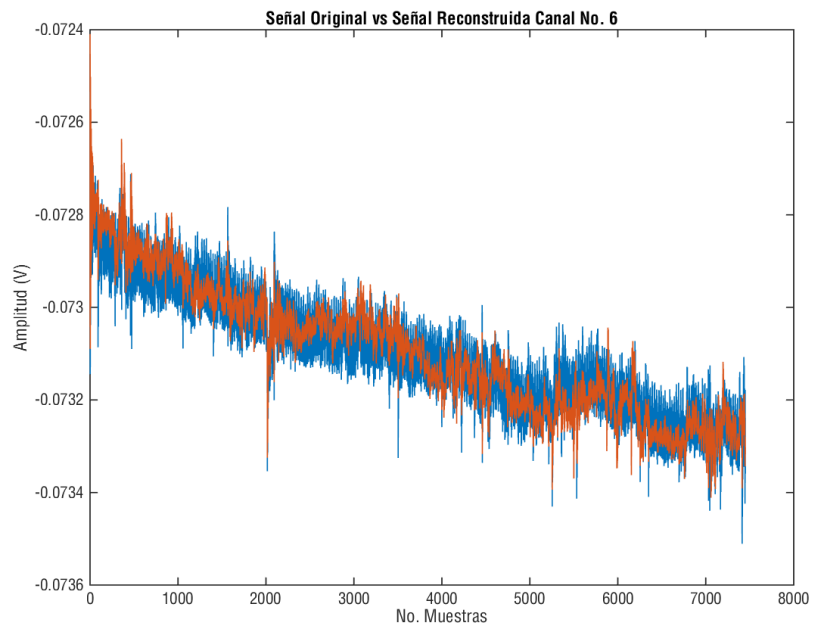
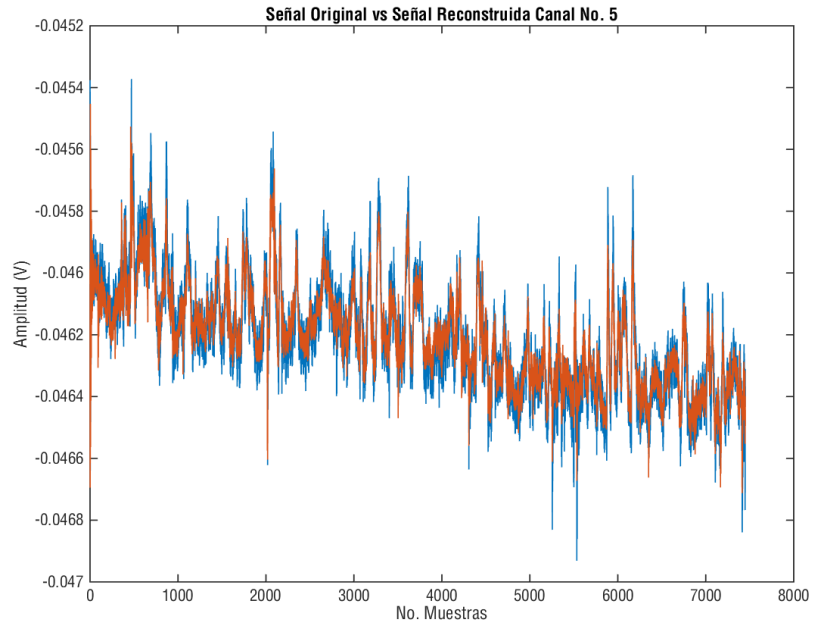
Sujeto 1. Movimiento

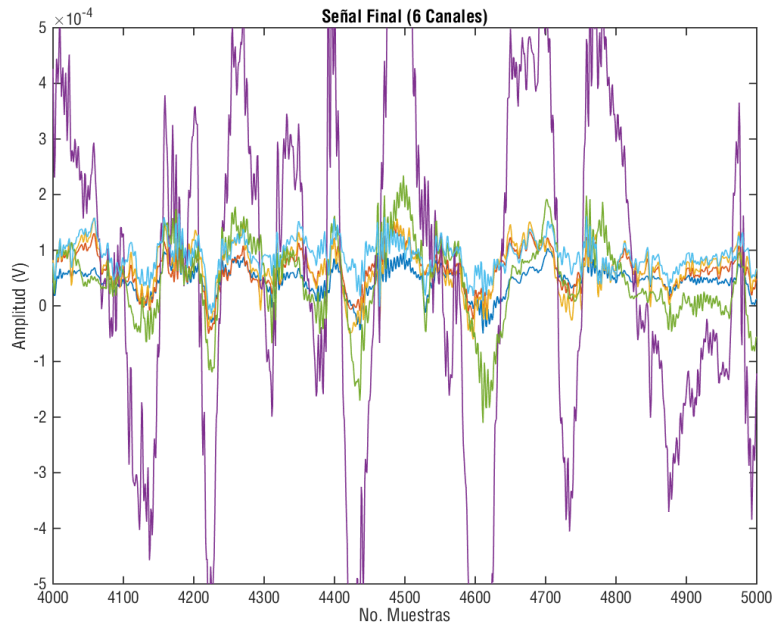




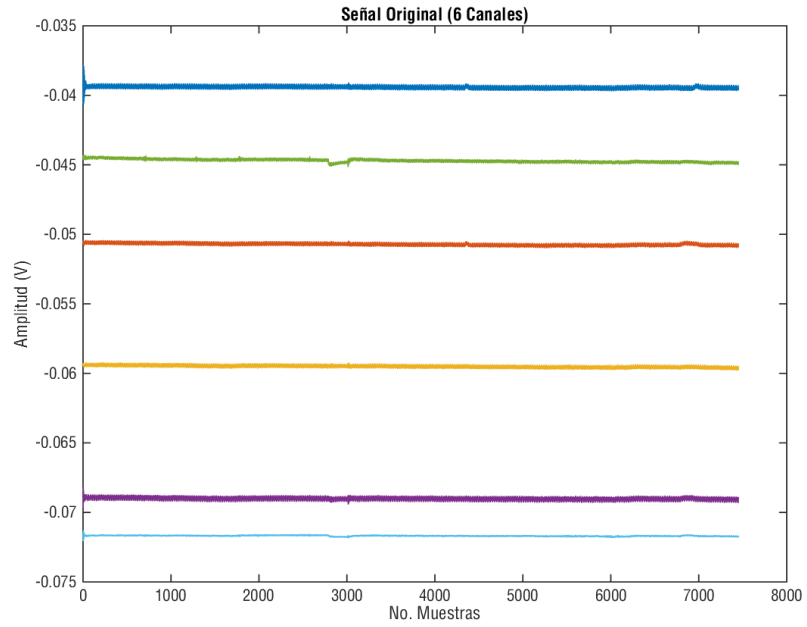


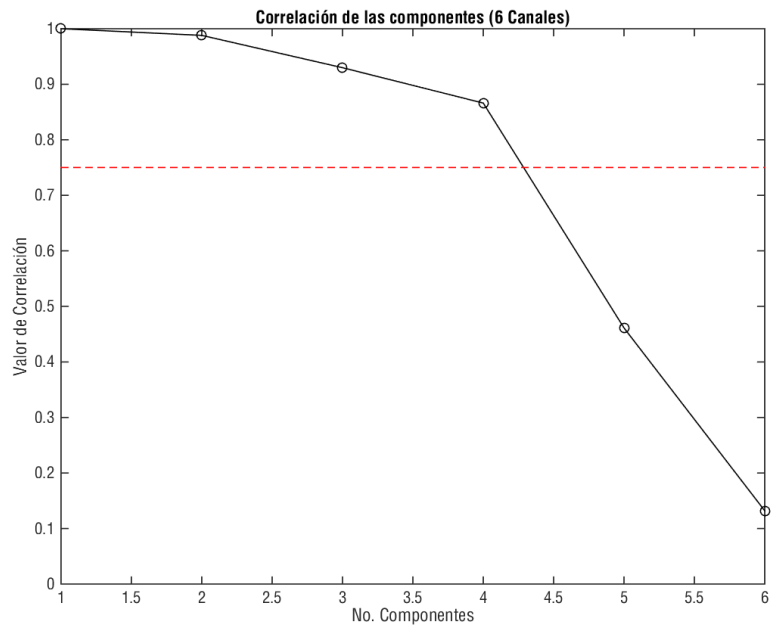
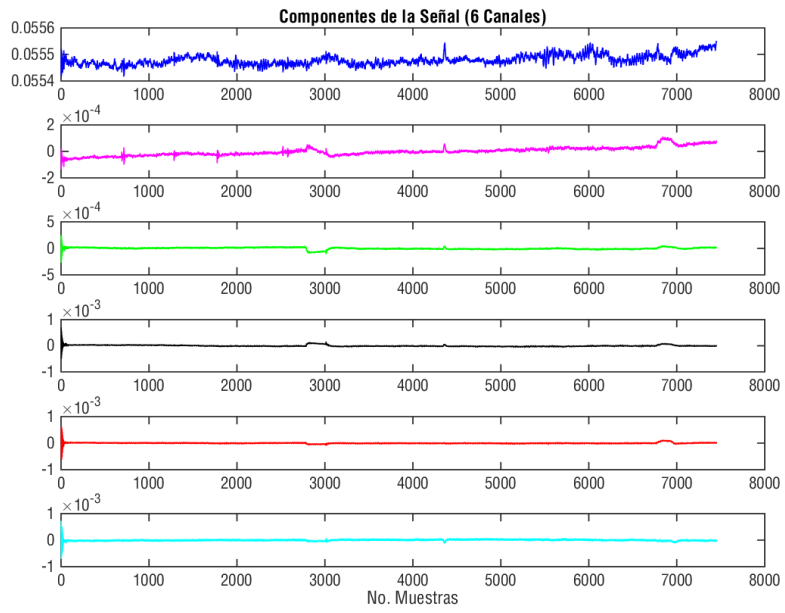


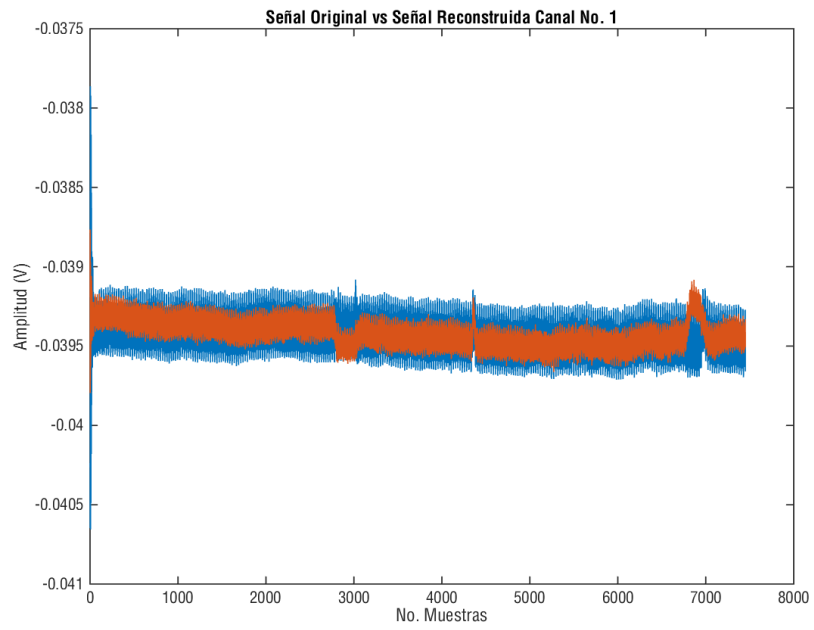
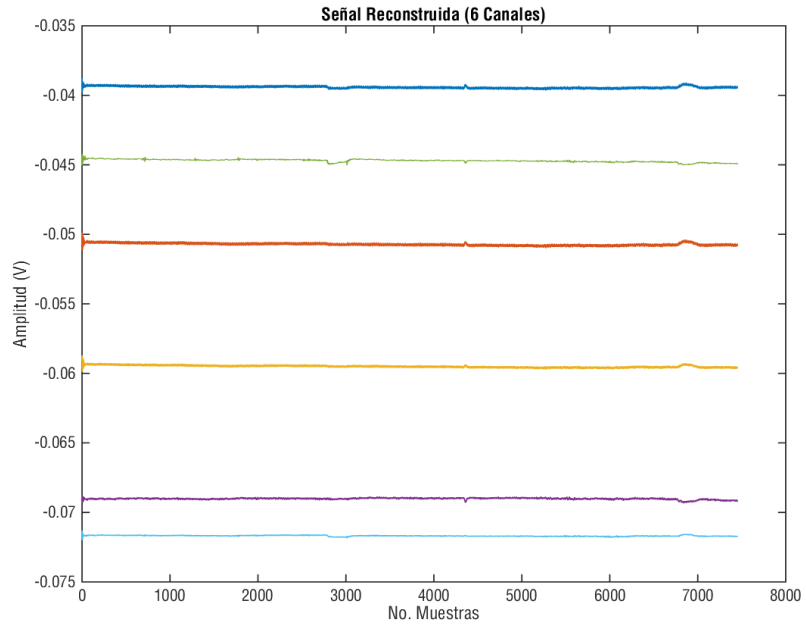


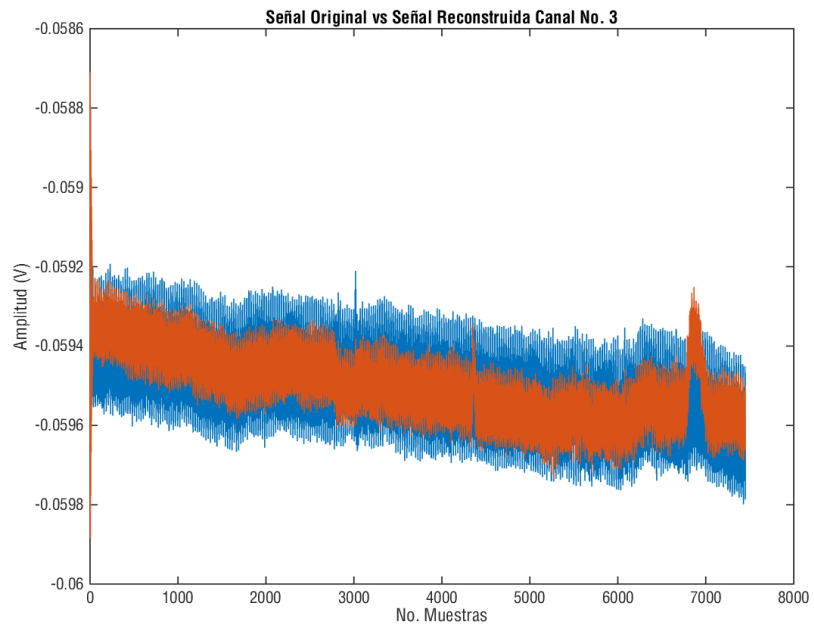
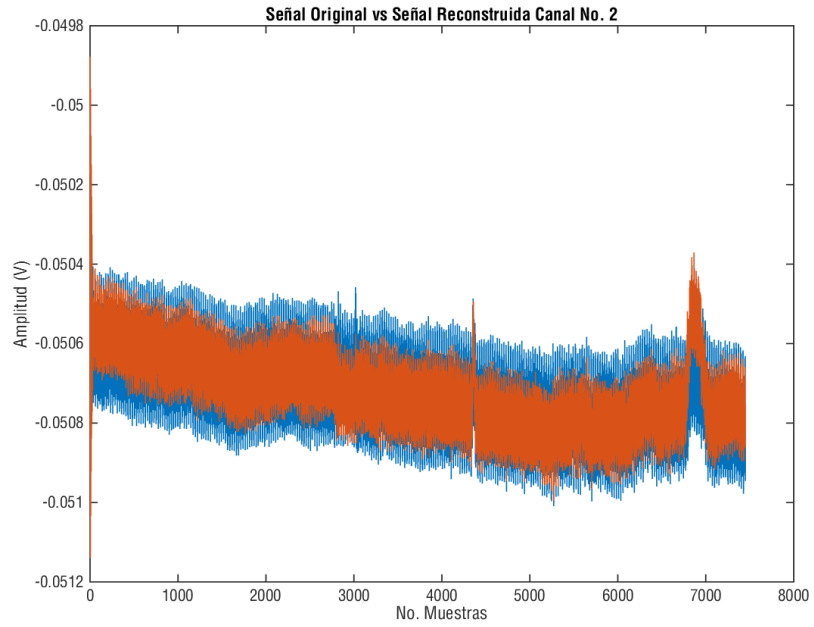


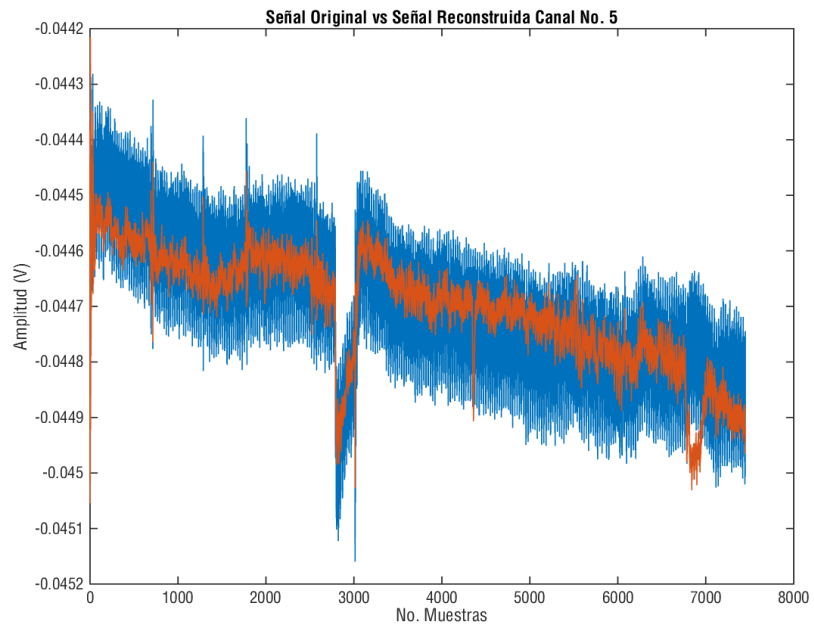
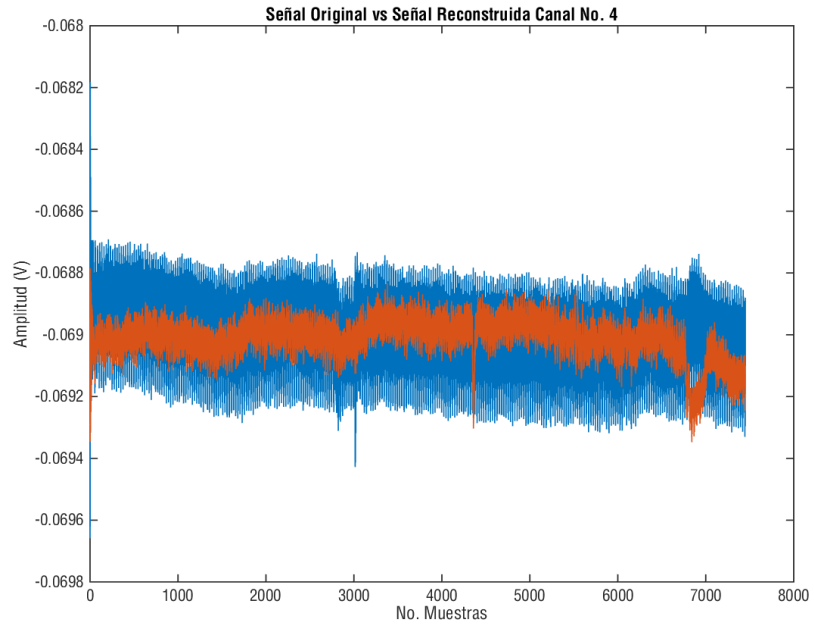
Sujeto 1. Reposo

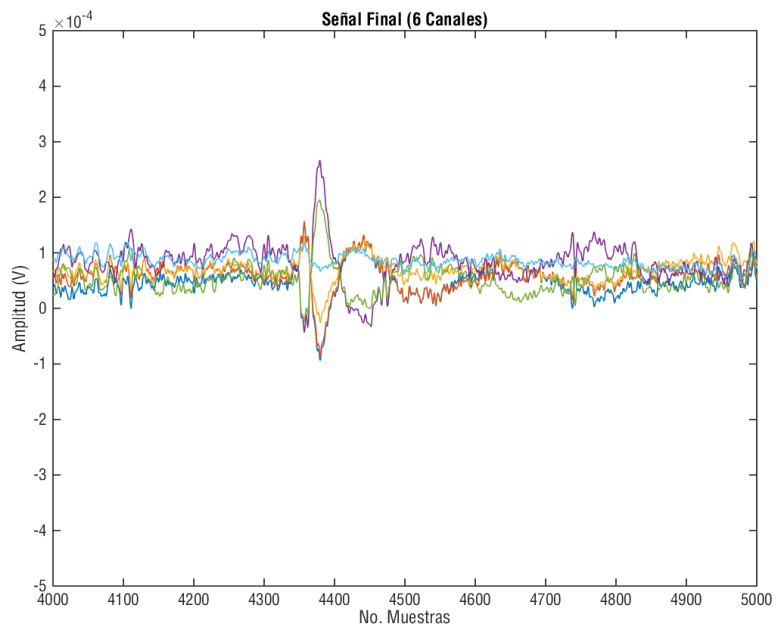
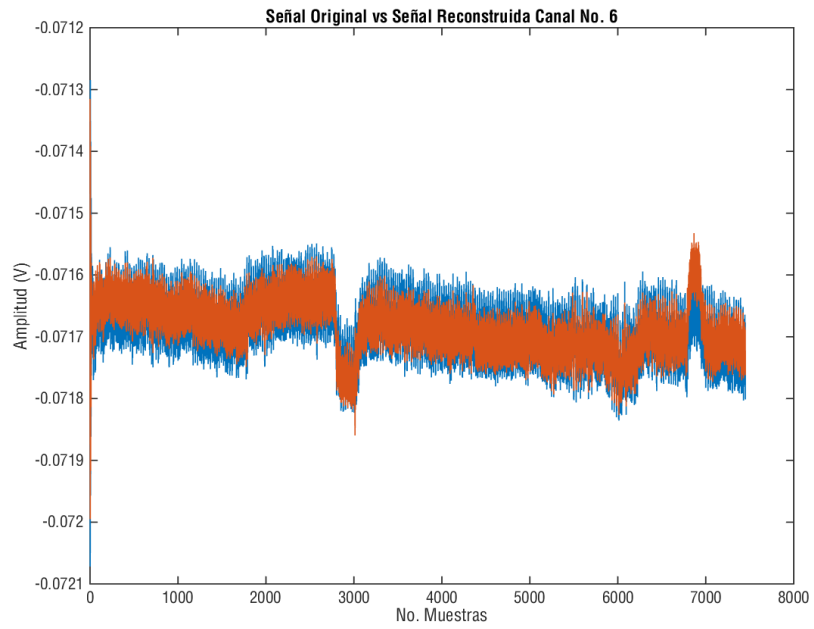




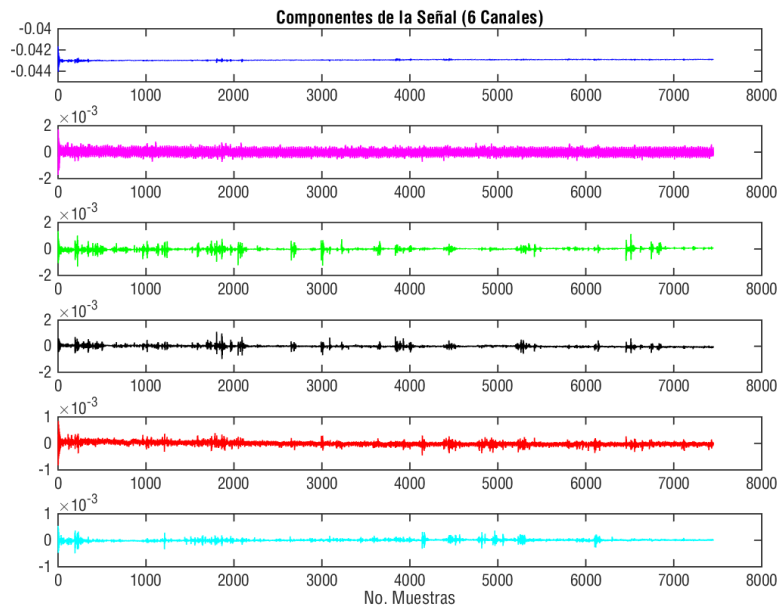
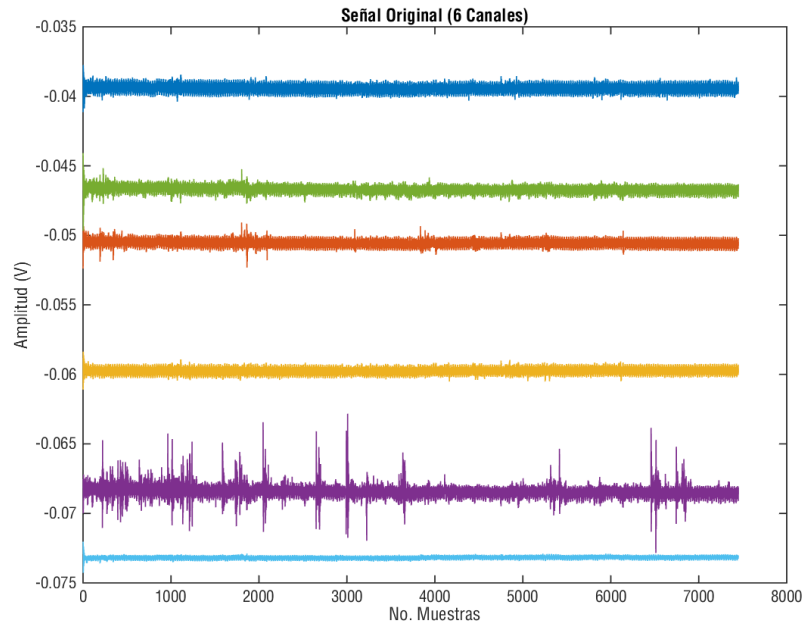


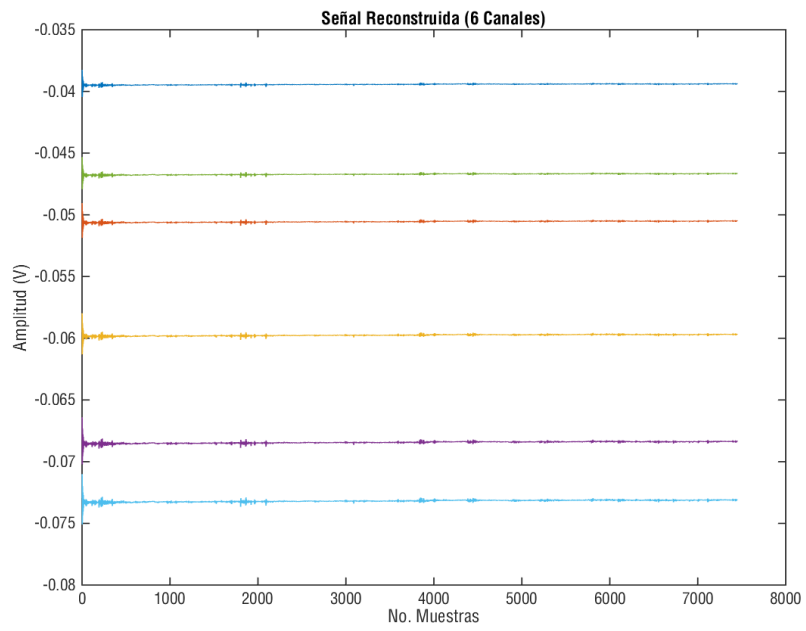
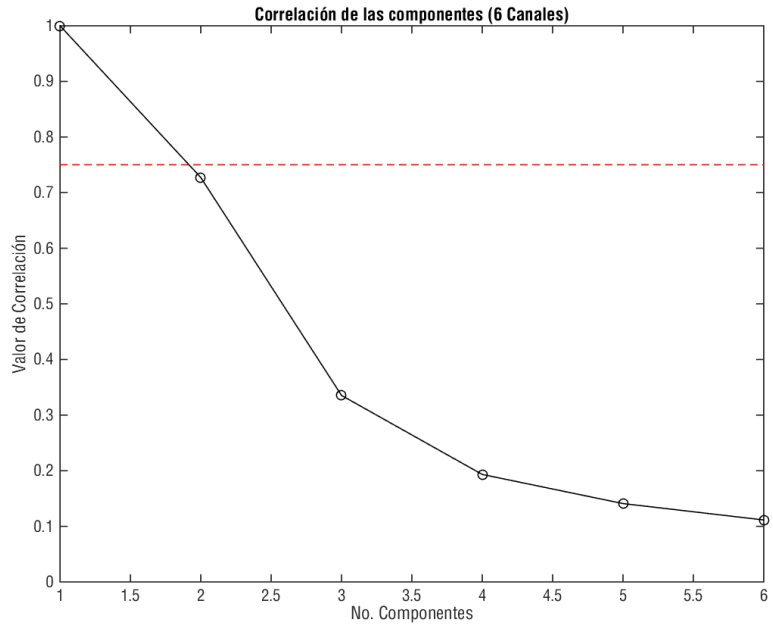


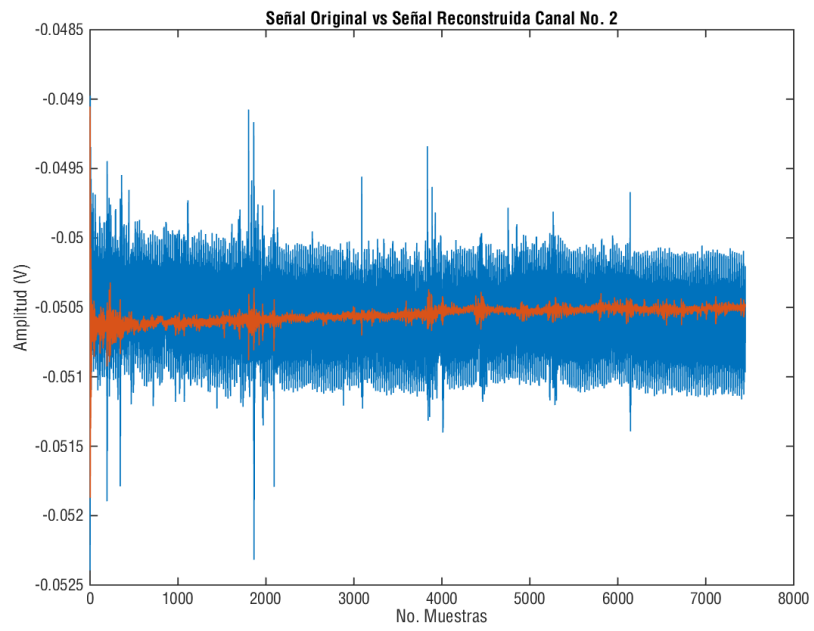
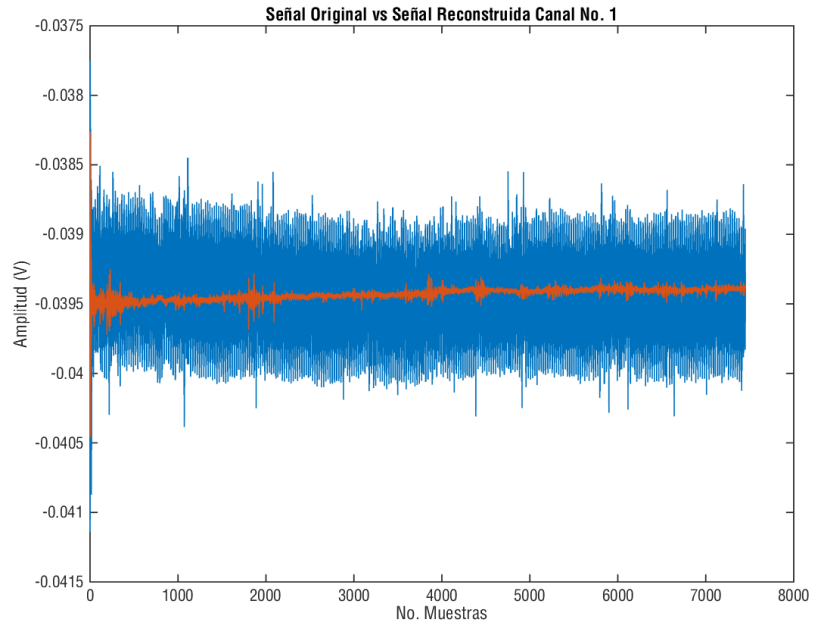


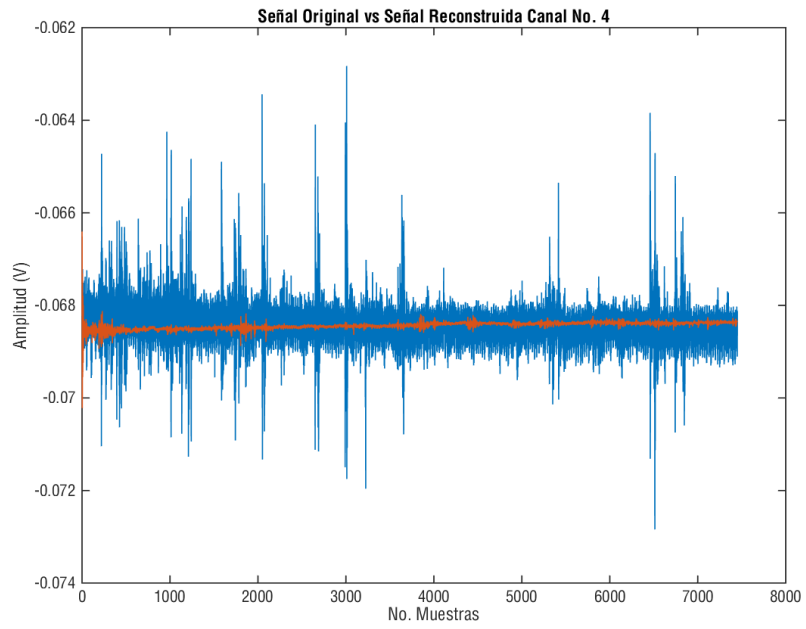
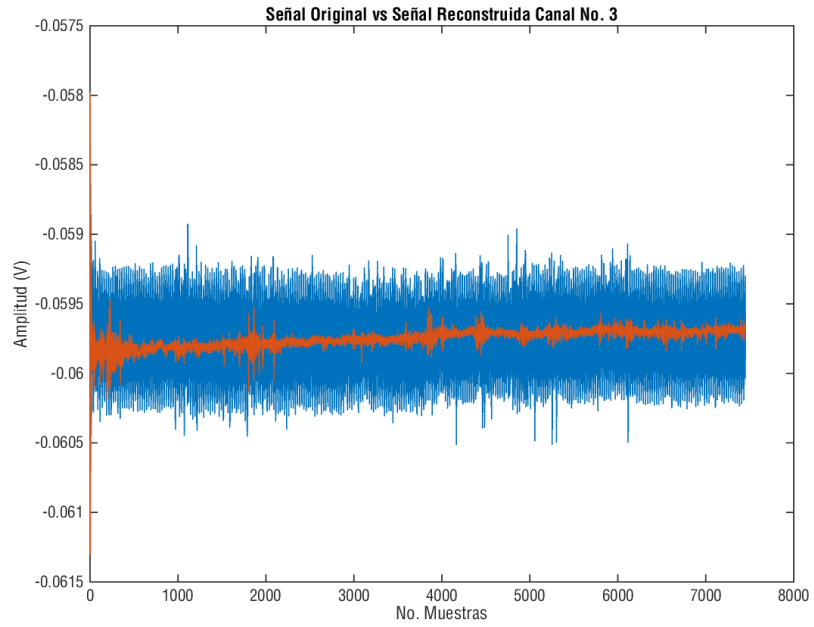


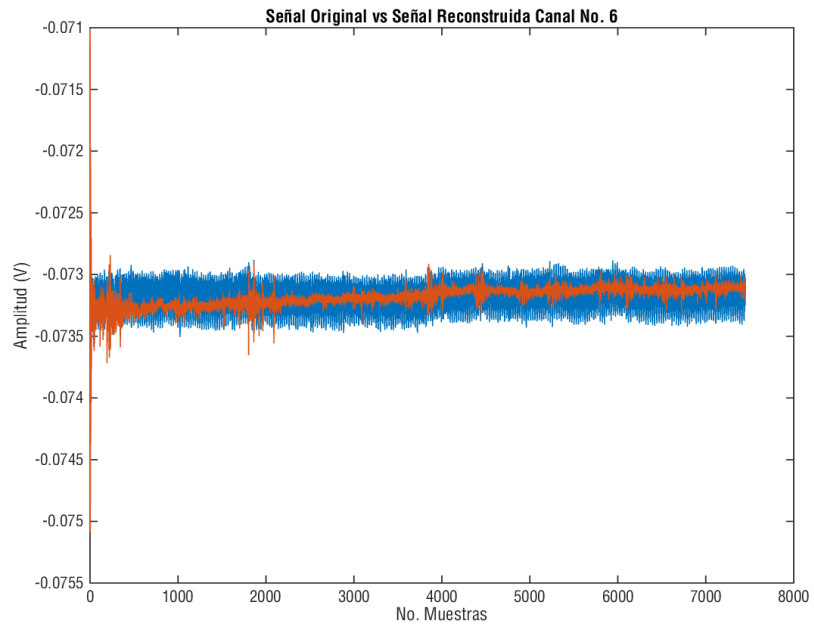
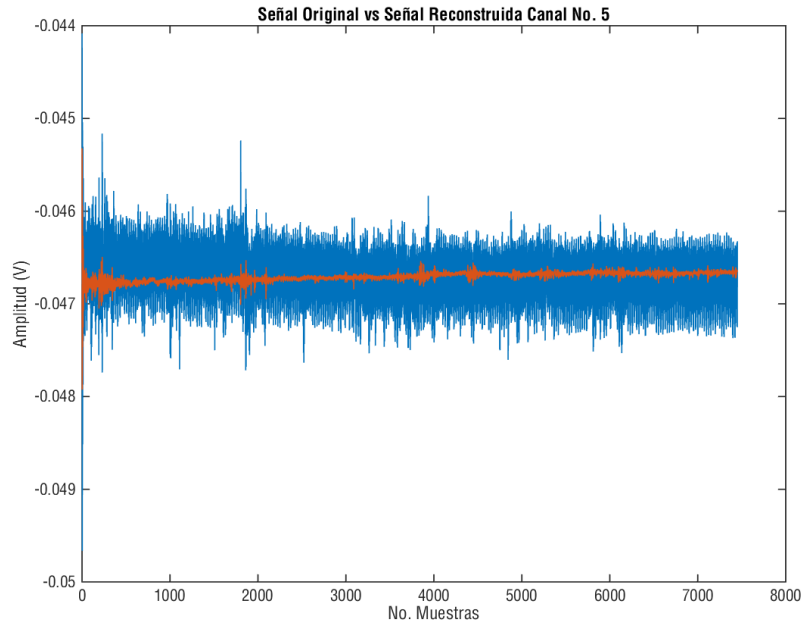
Sujeto 2. Movimiento

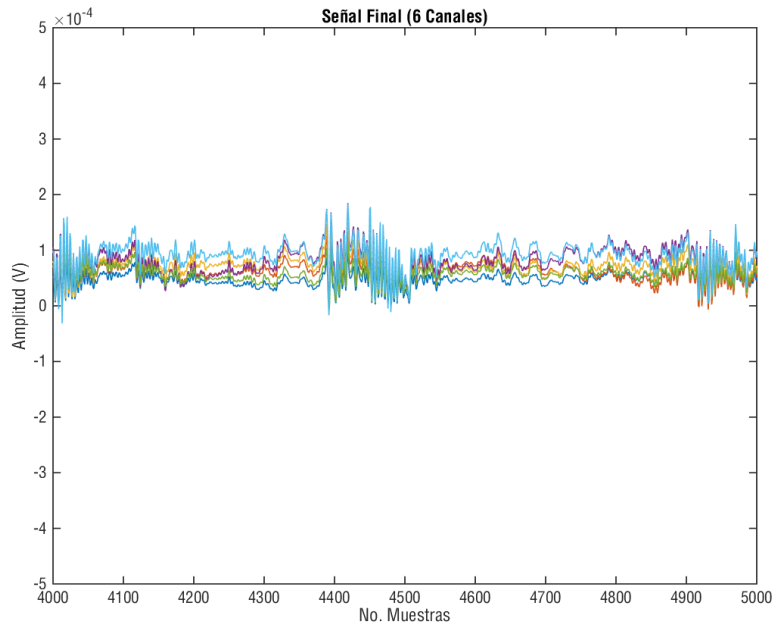




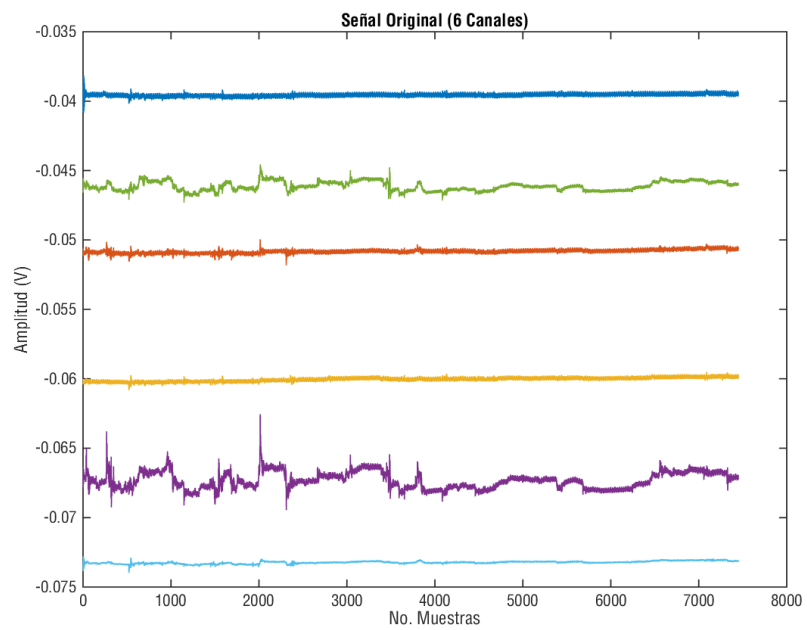


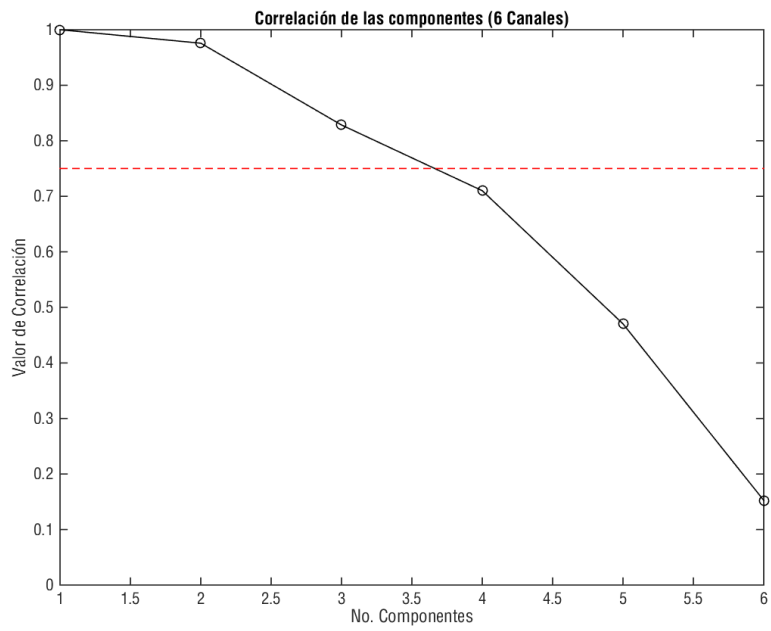
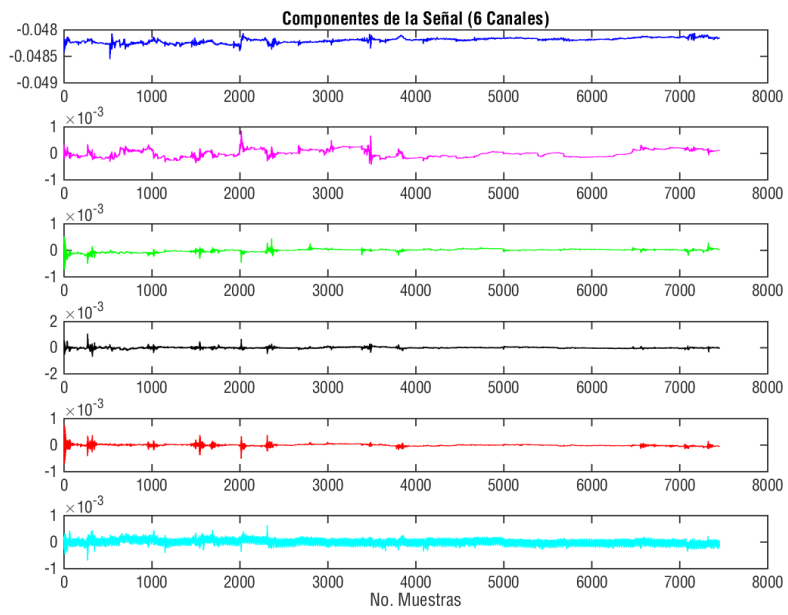


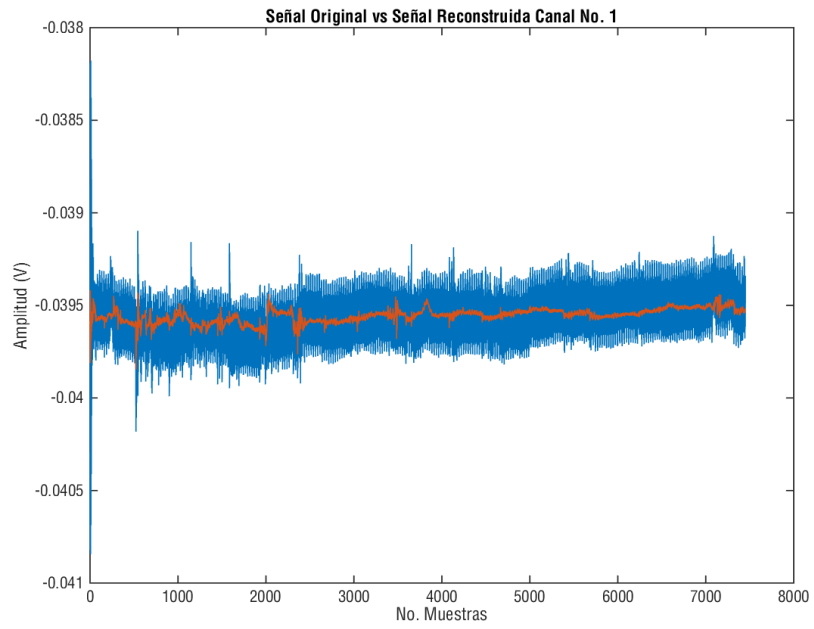
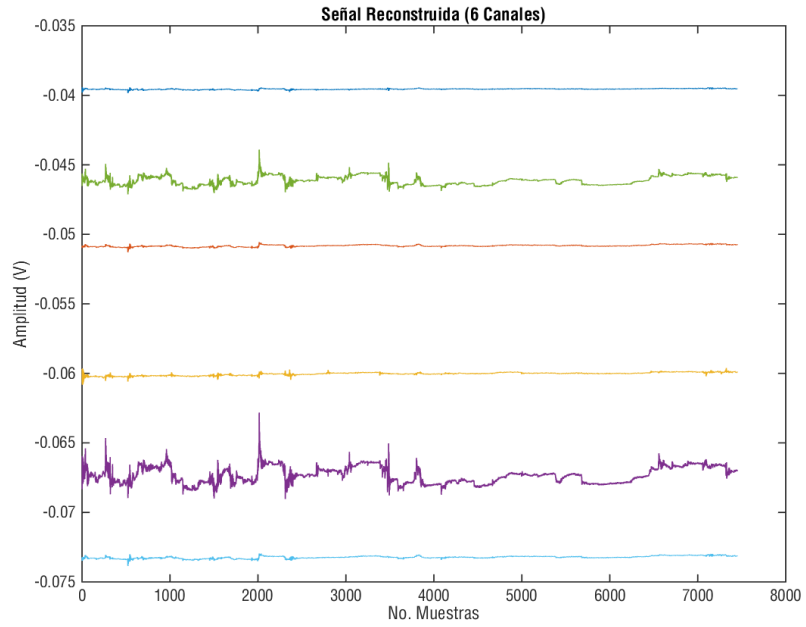


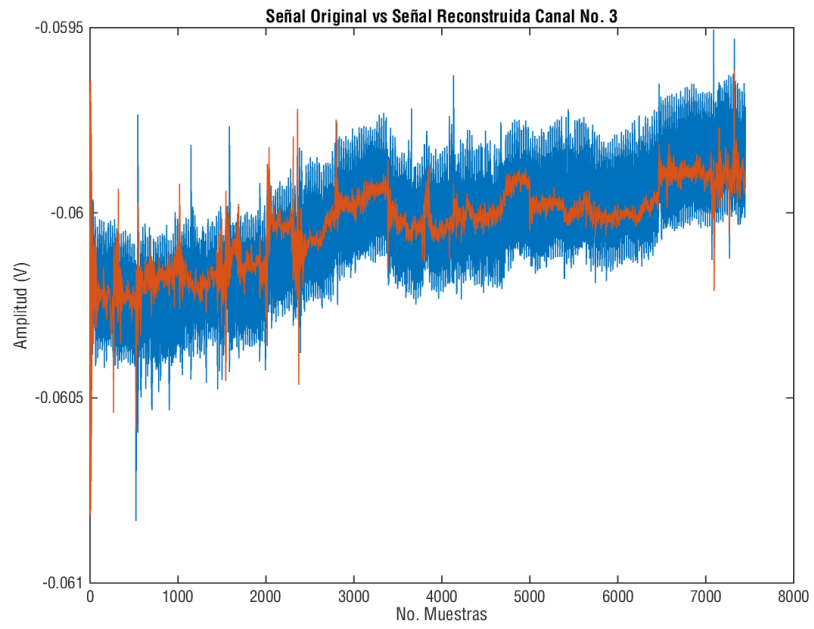
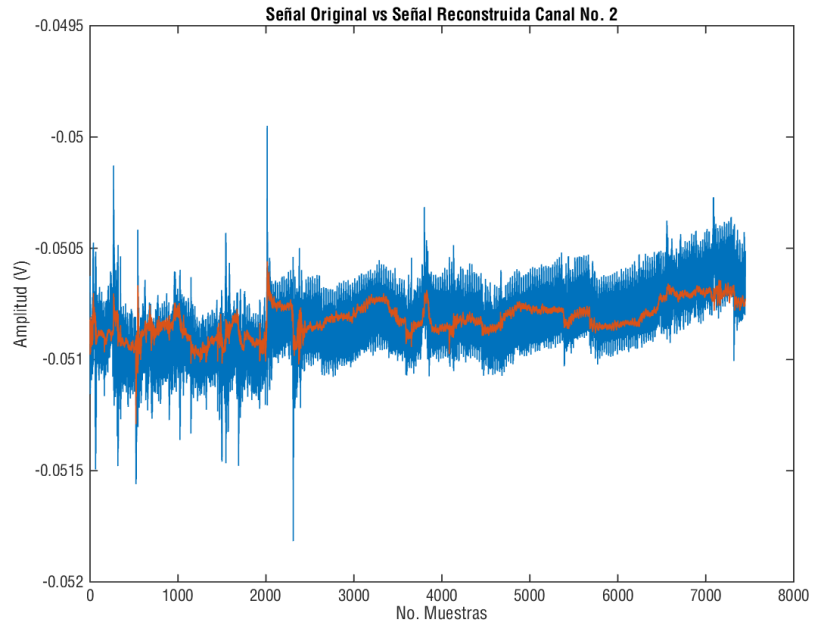


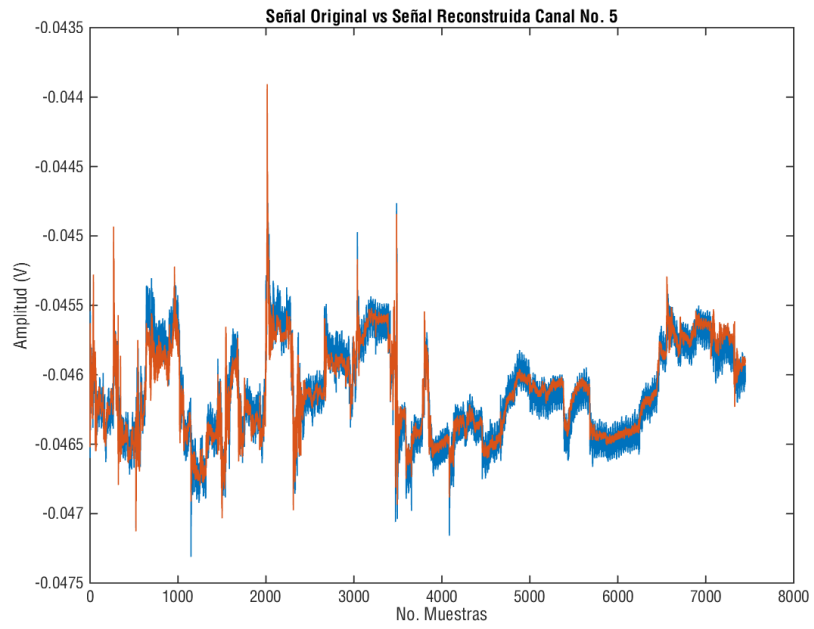
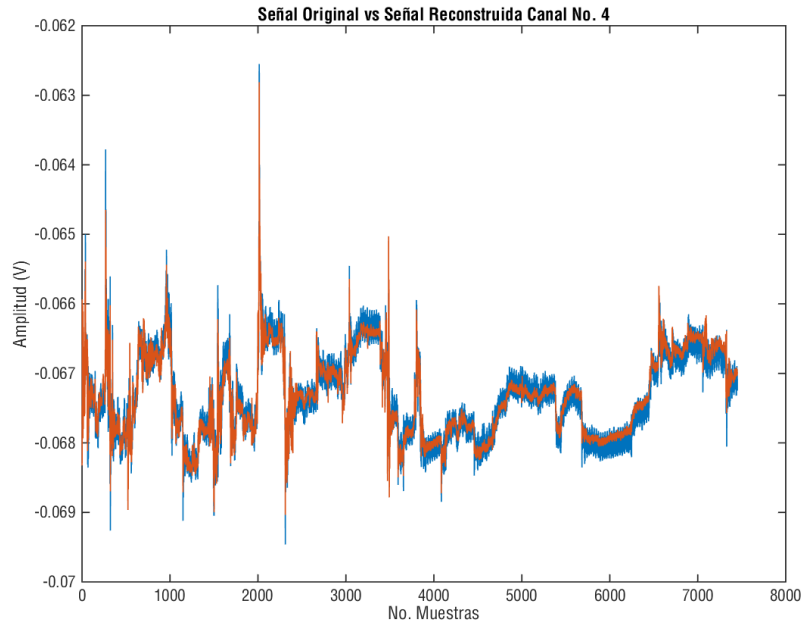
Sujeto 2. Reposo

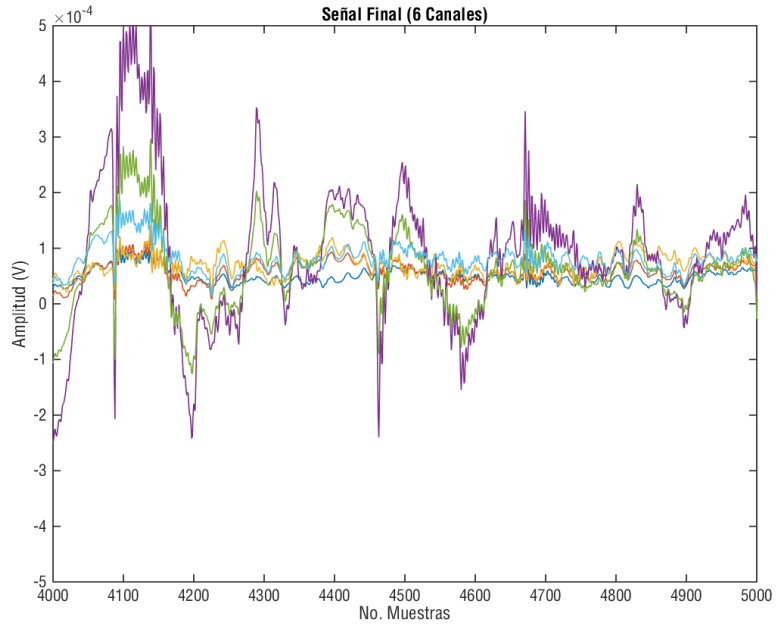
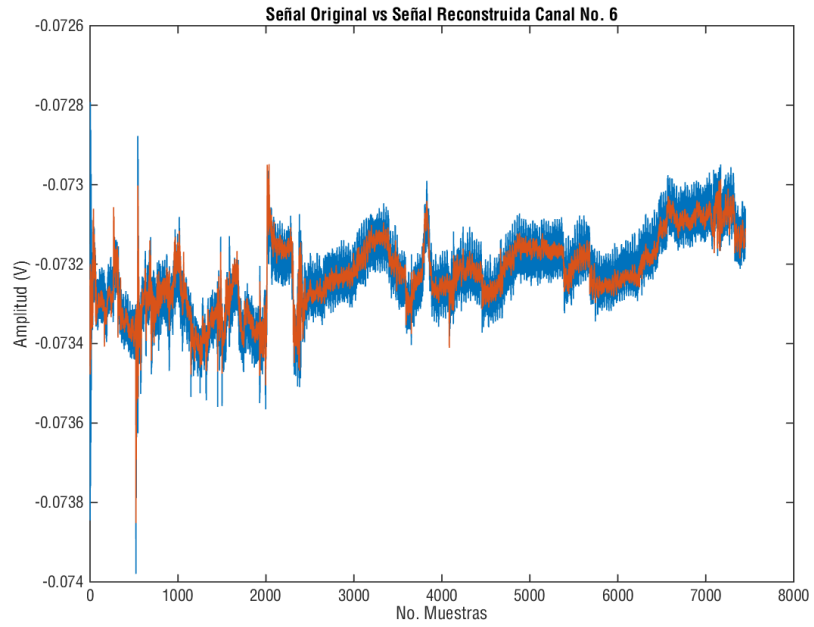




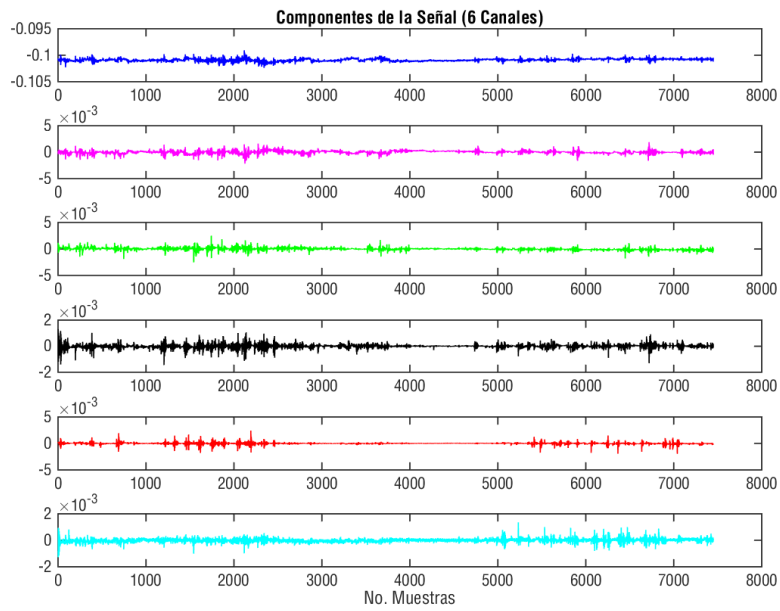
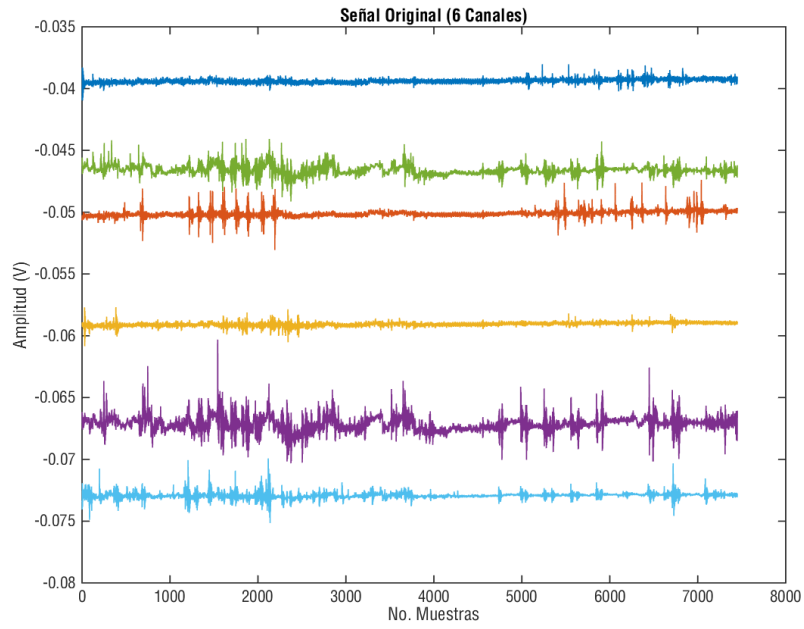


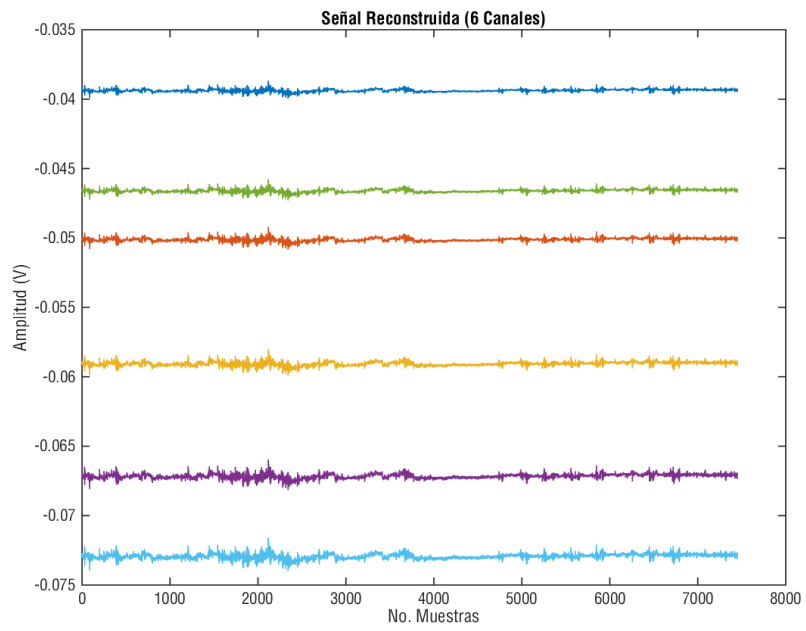
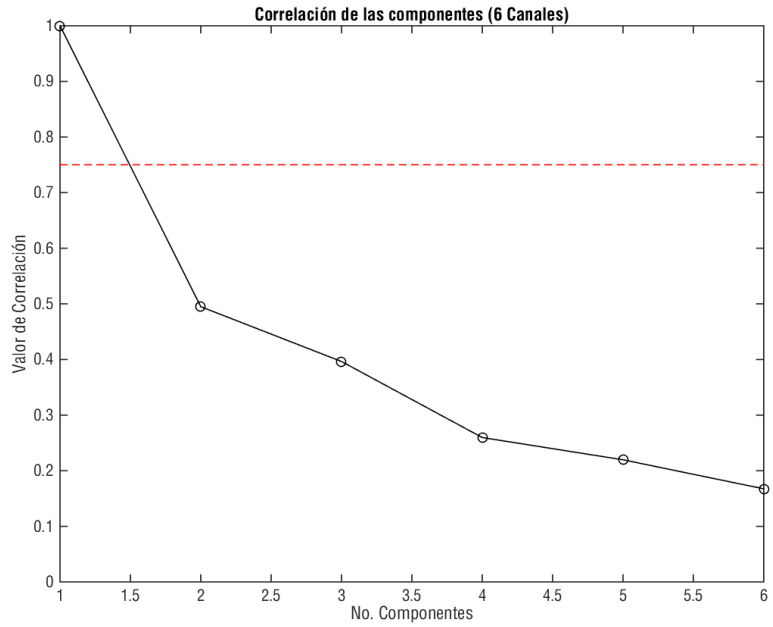


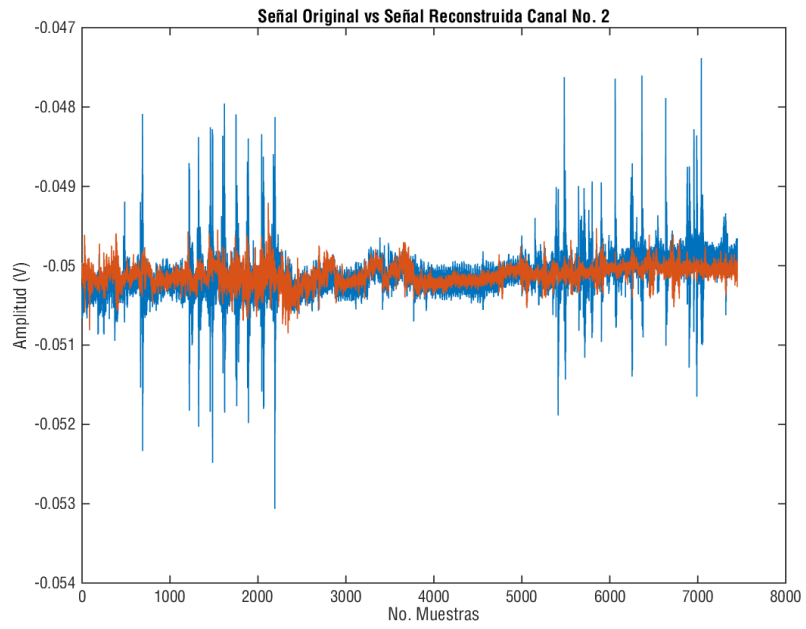
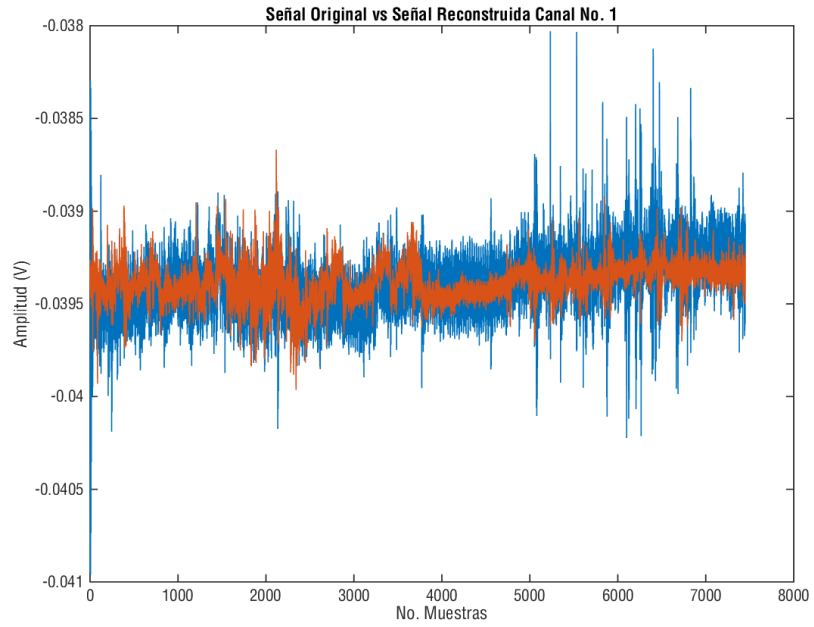


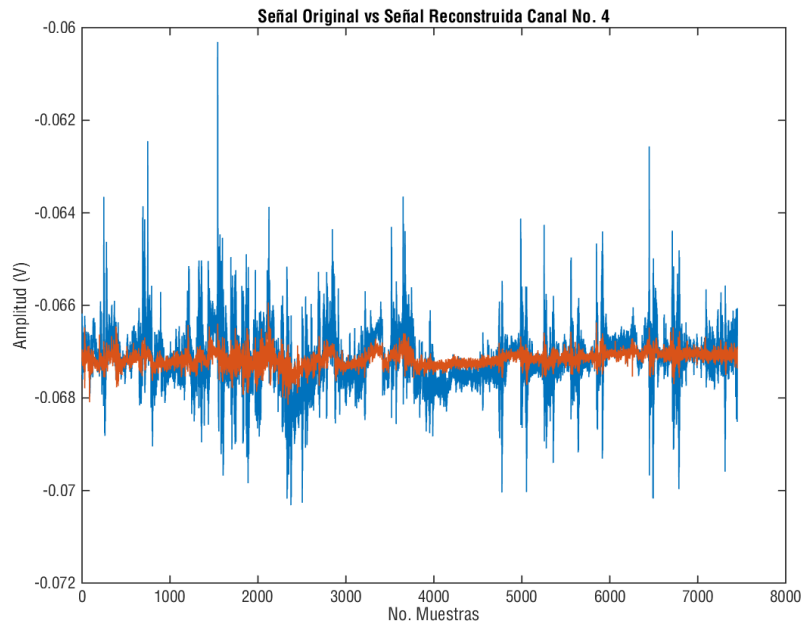
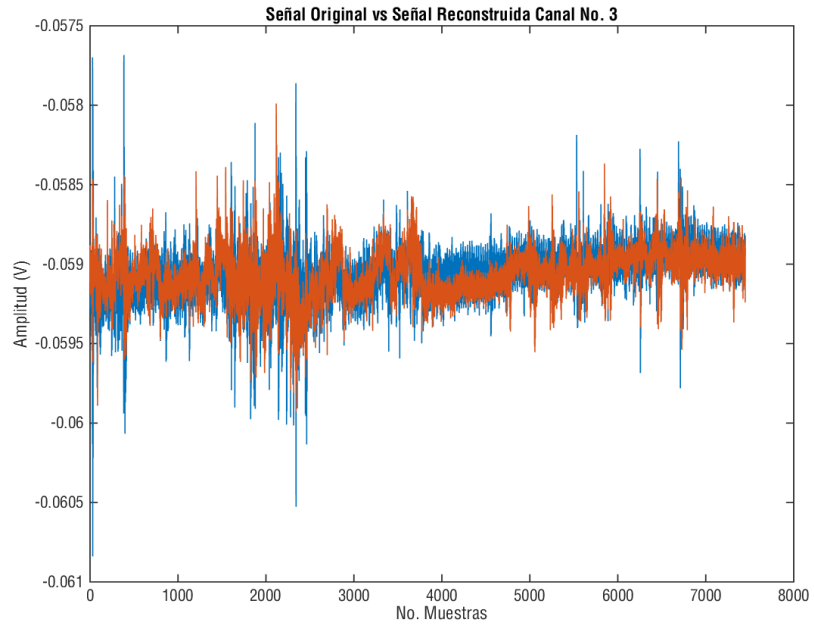


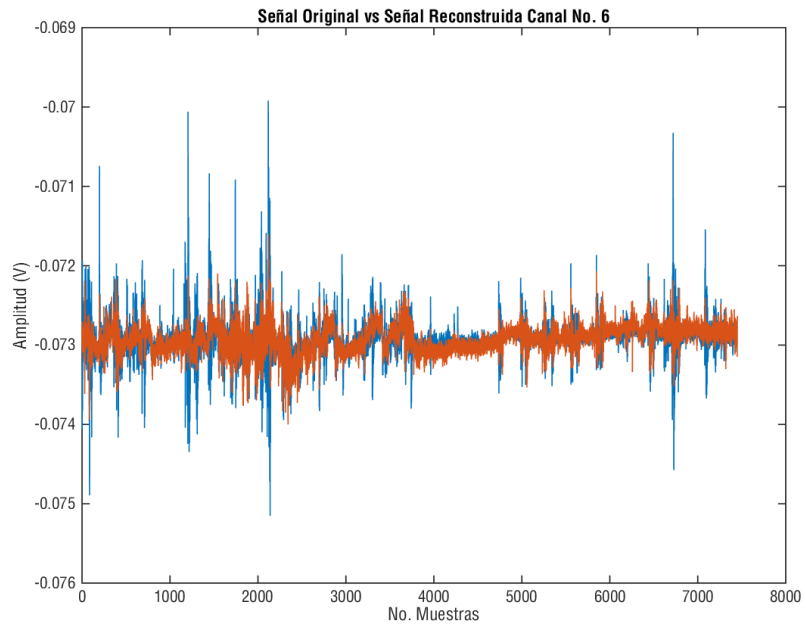
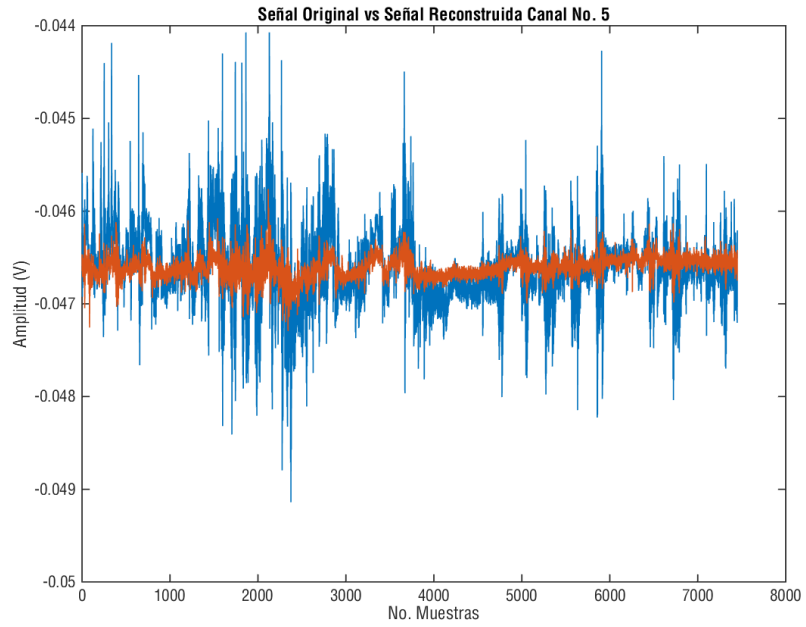
Sujeto 3. Movimiento

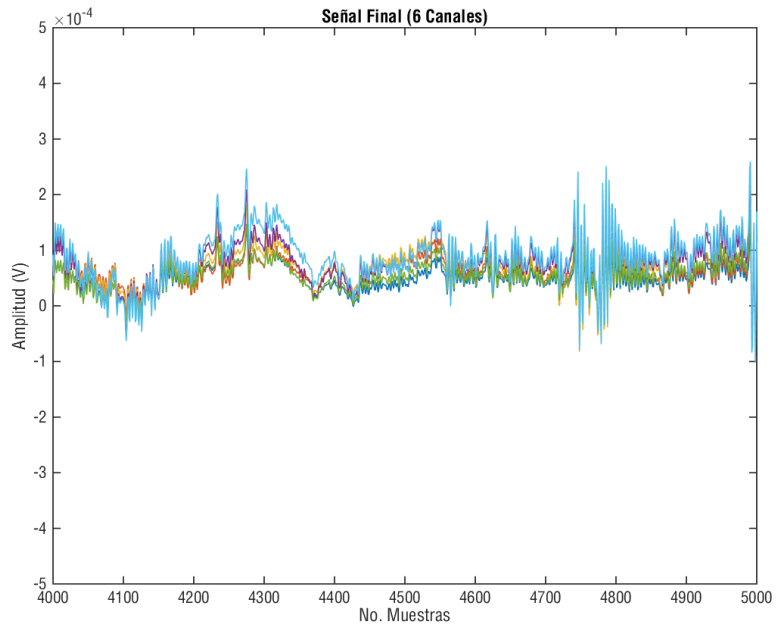




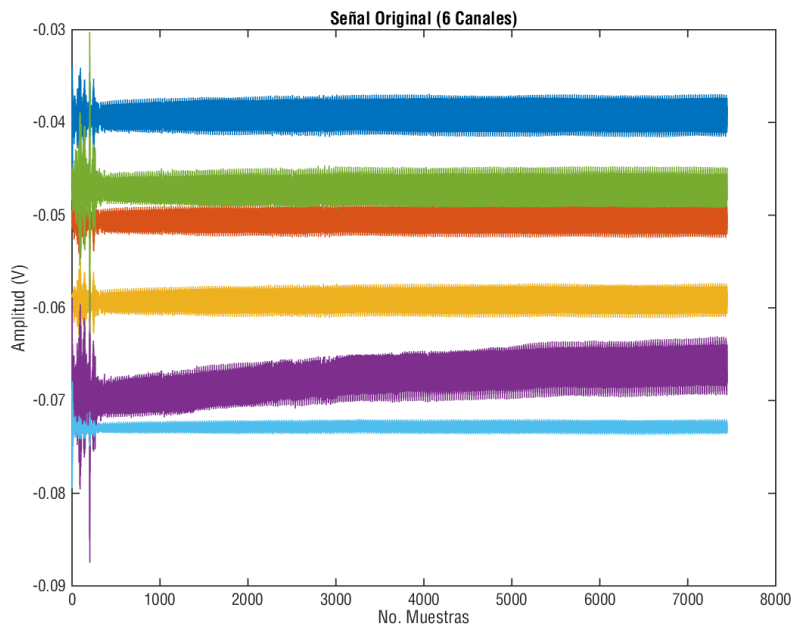


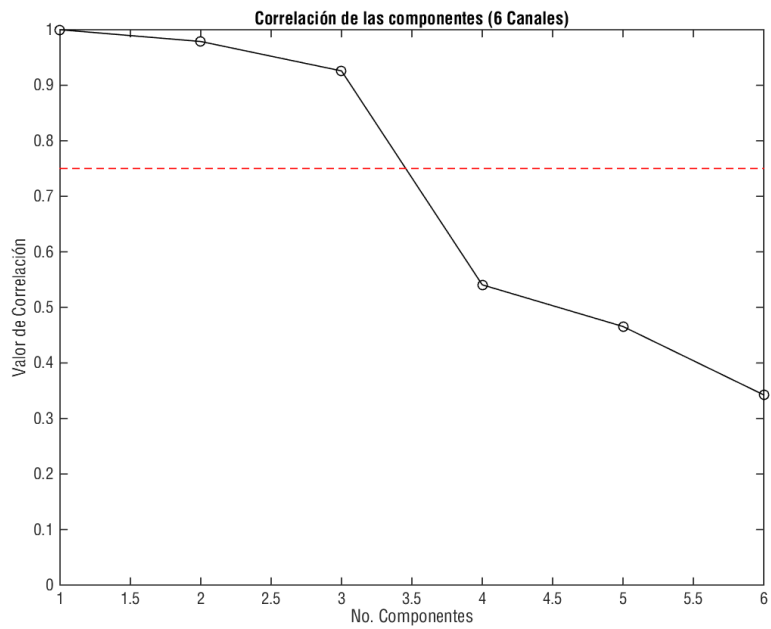
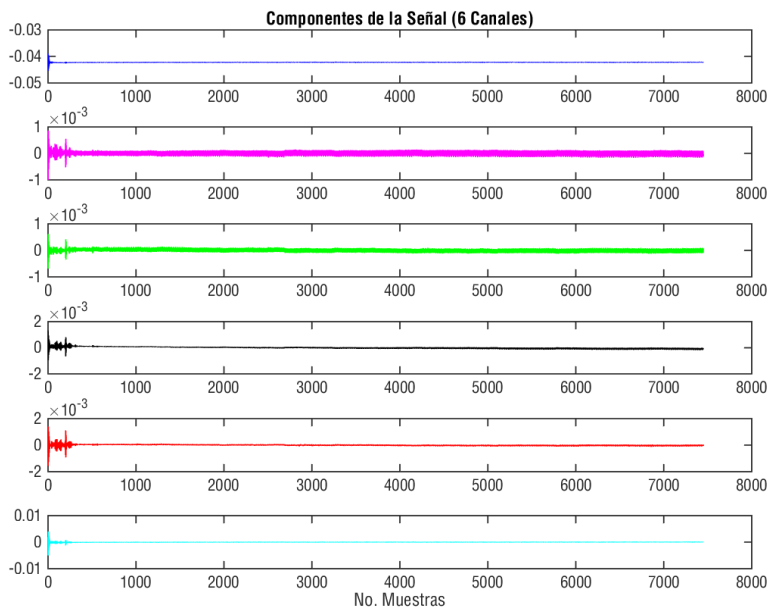


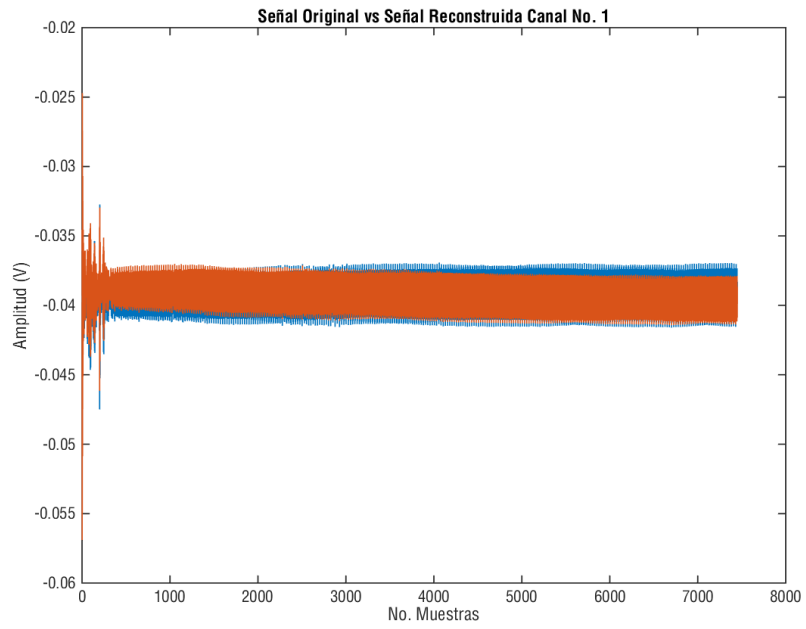
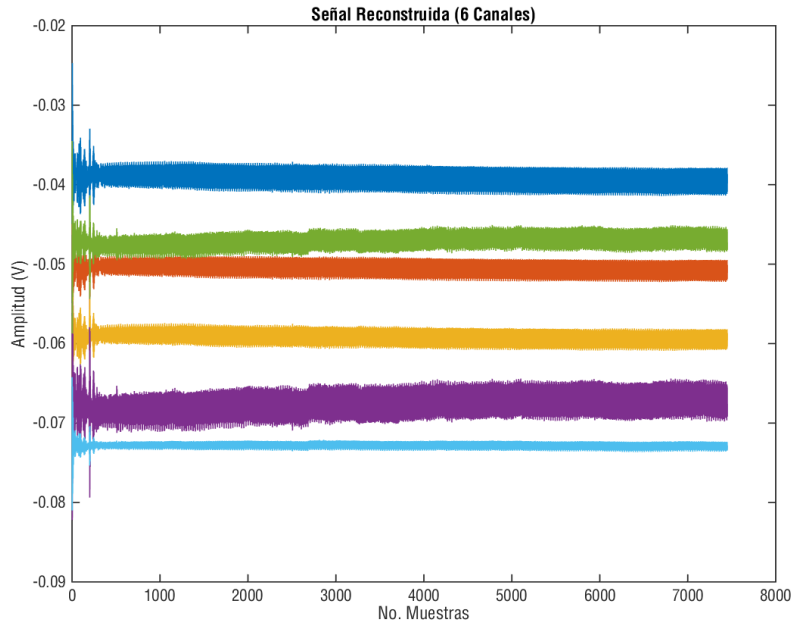


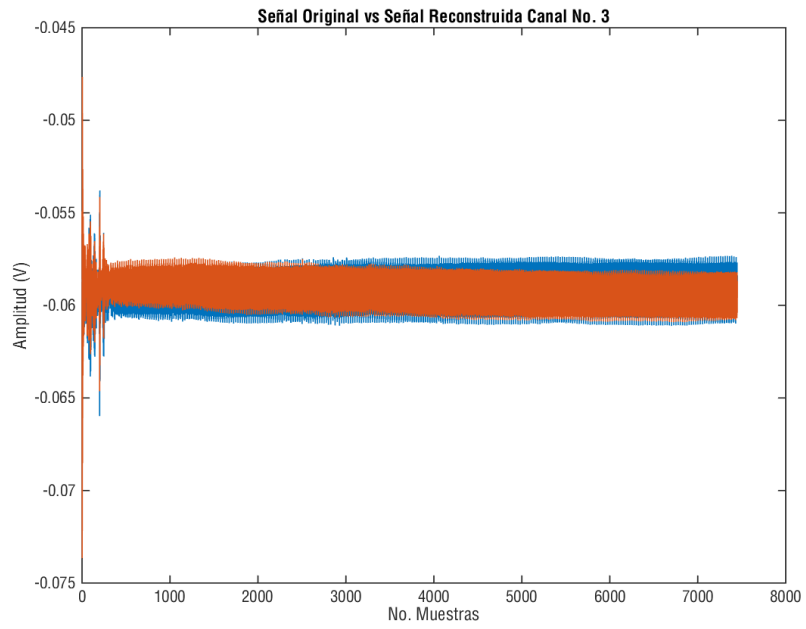
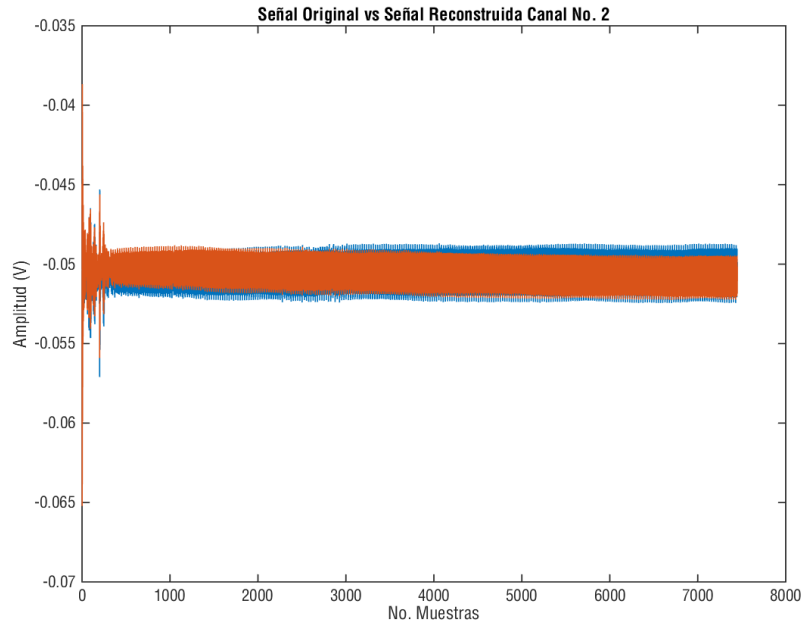


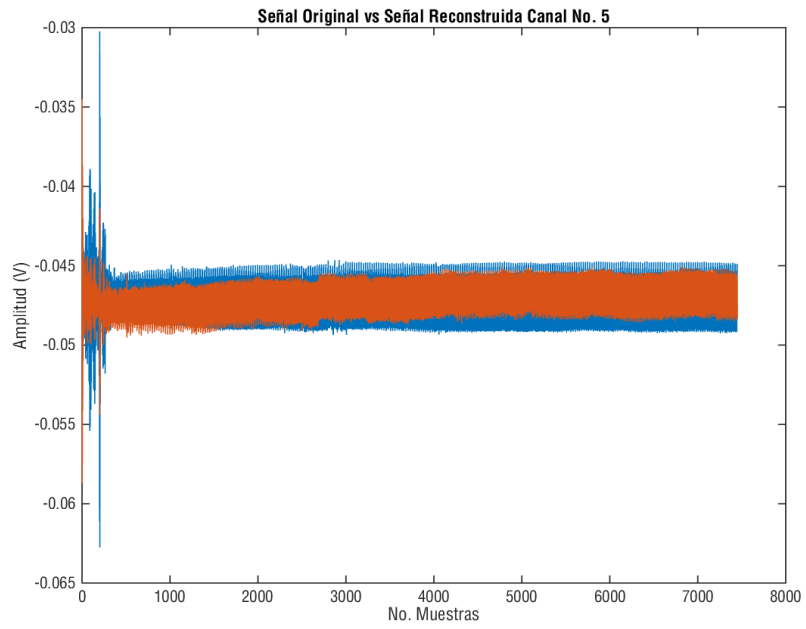
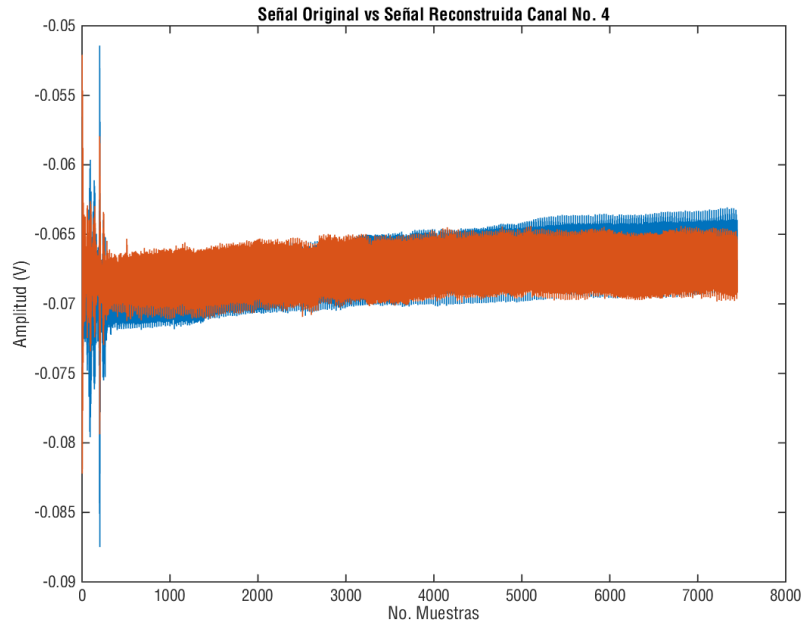
Sujeto 3. Reposo

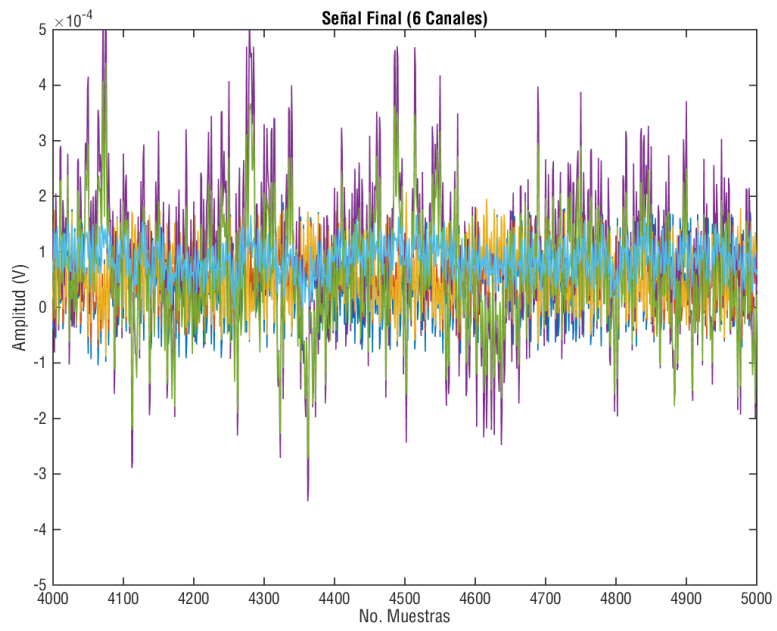
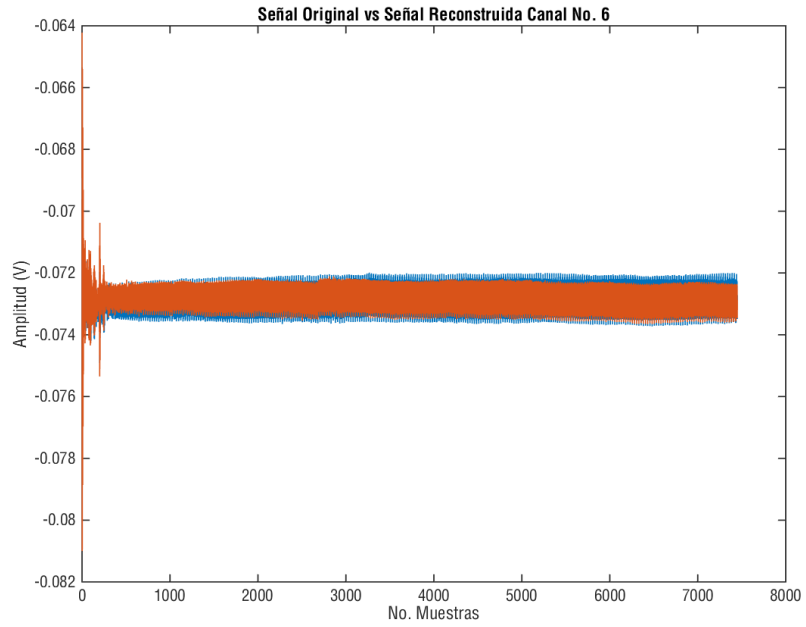




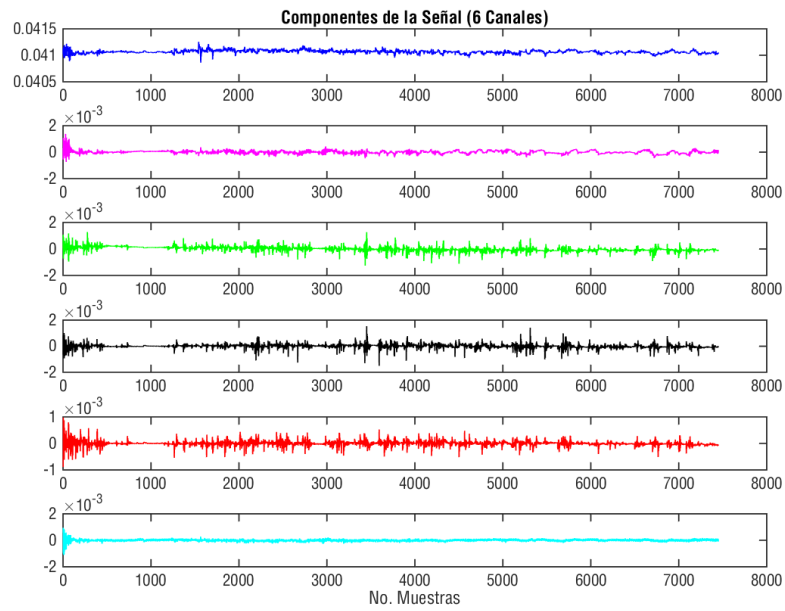
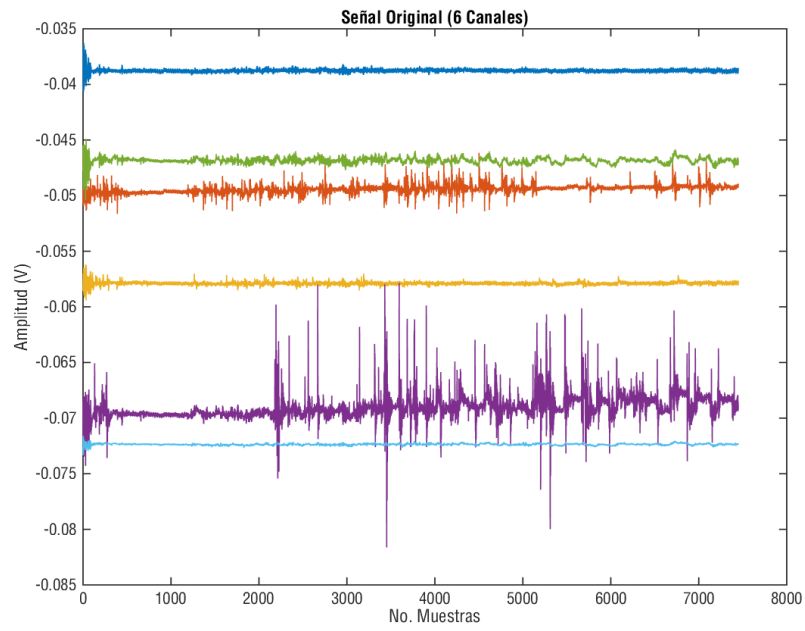


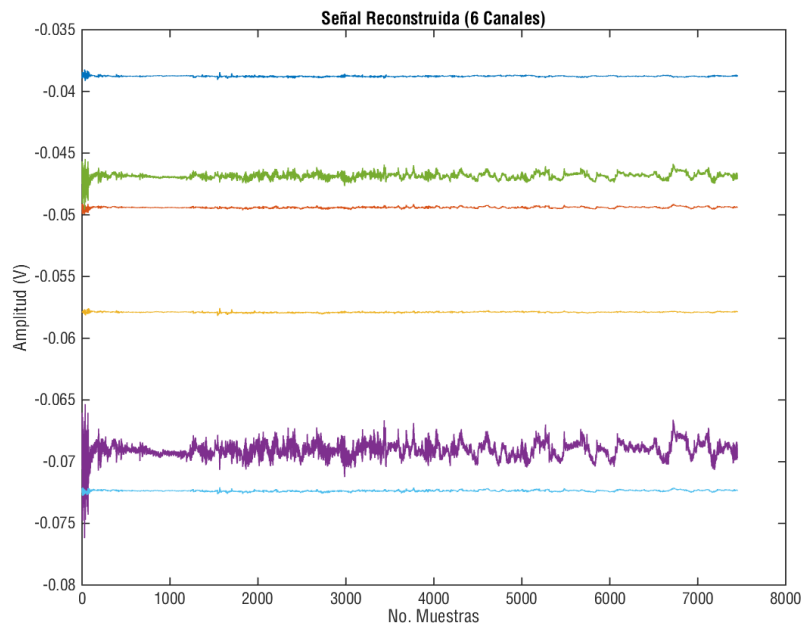
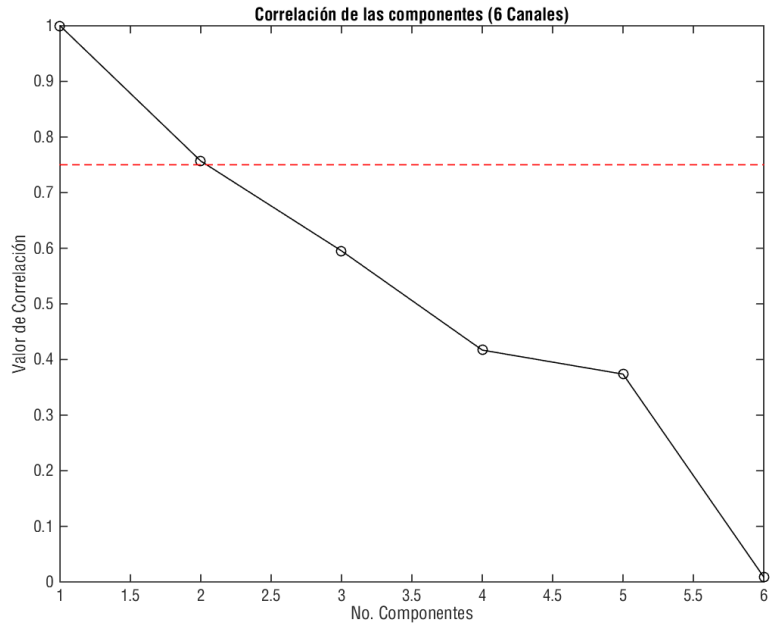


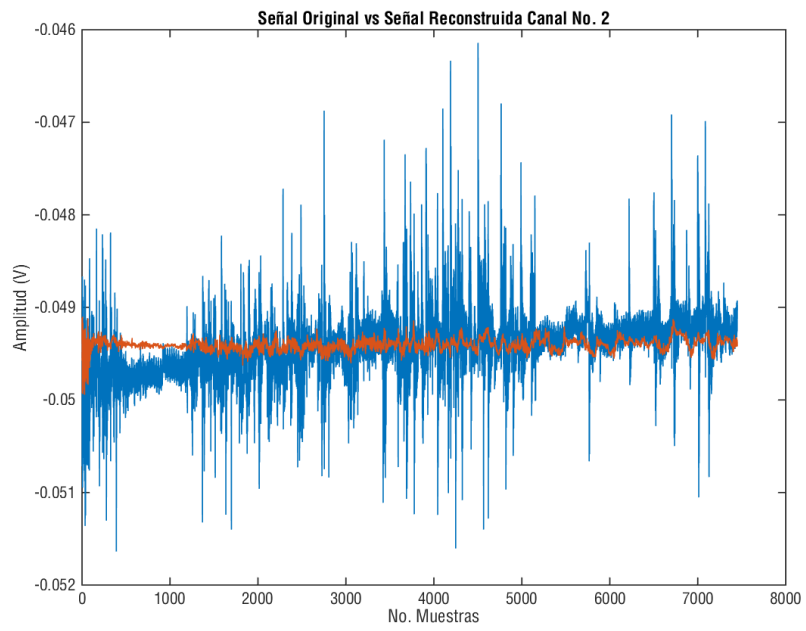
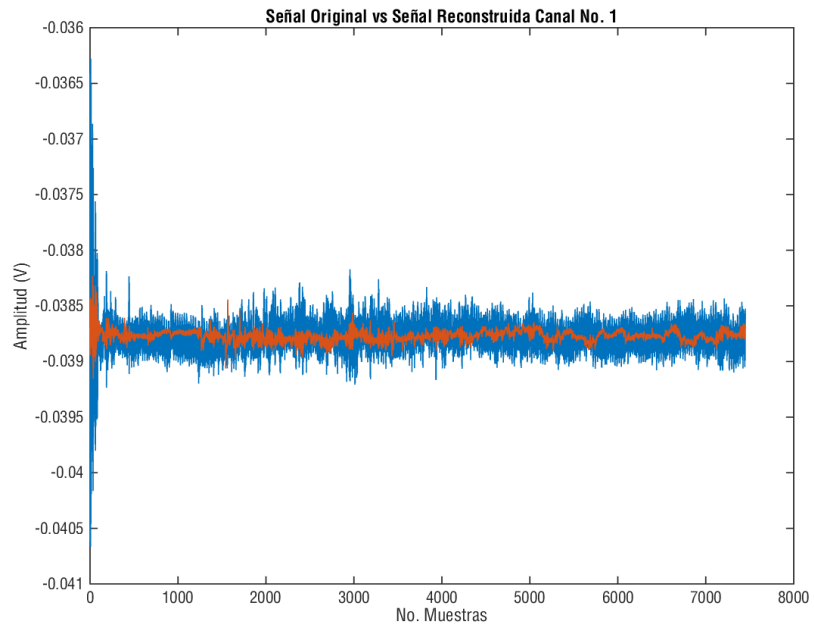


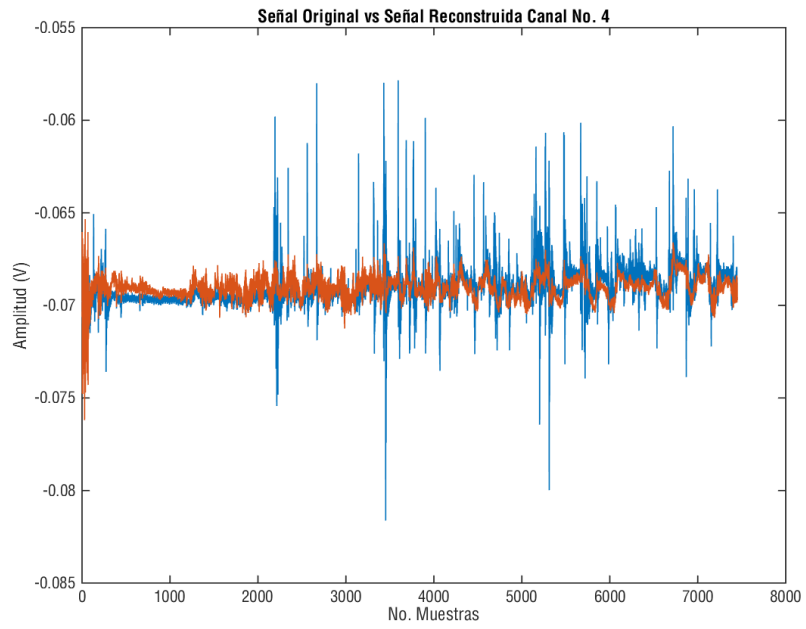
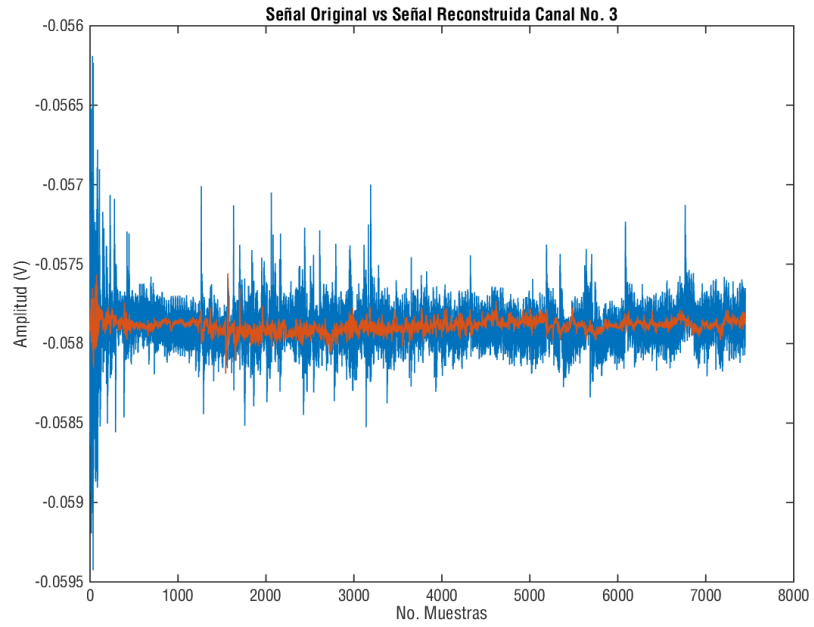


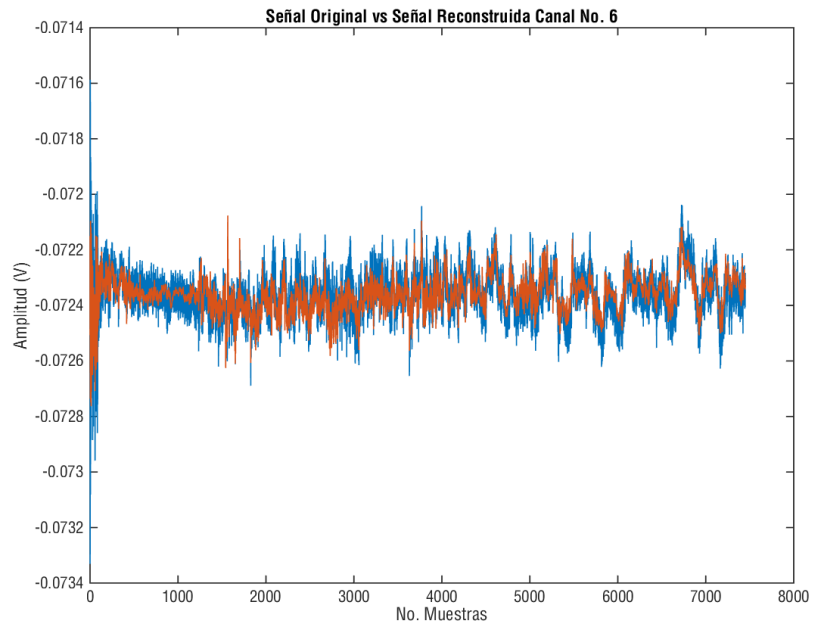
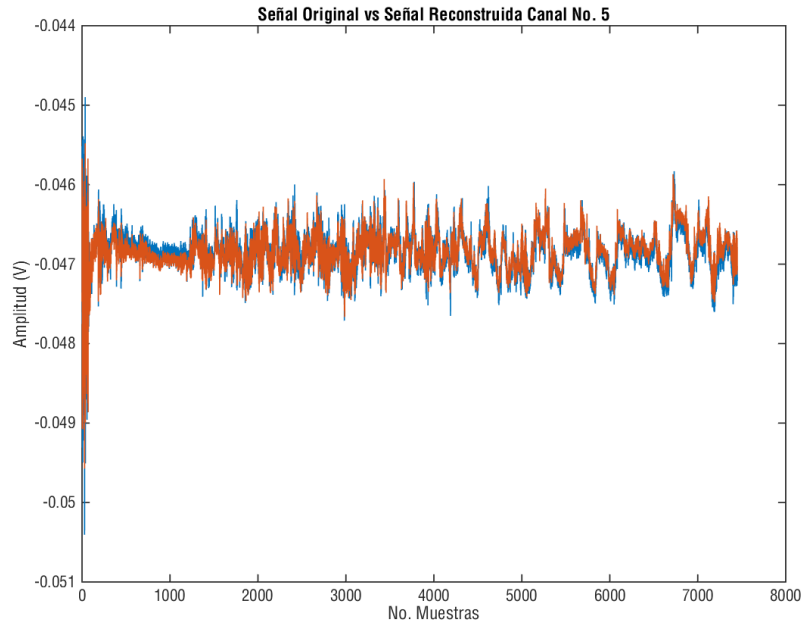
Sujeto 4. Movimiento

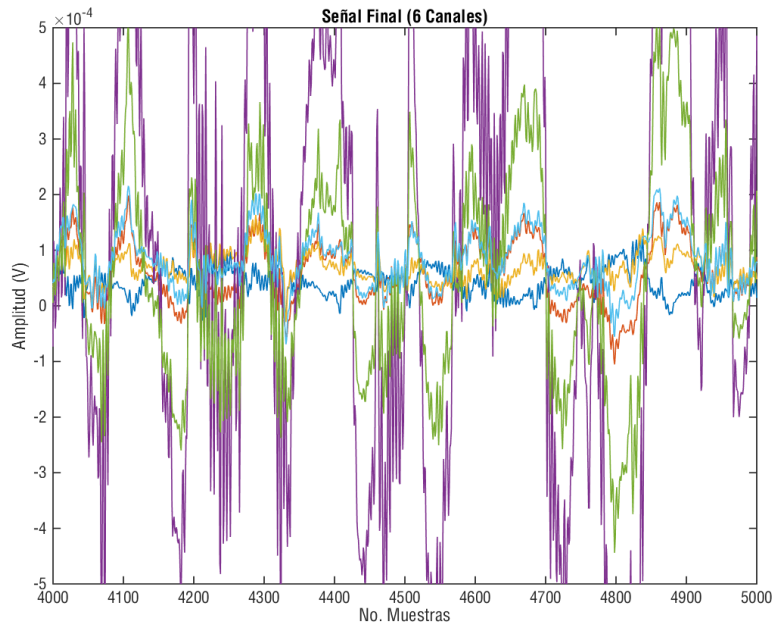




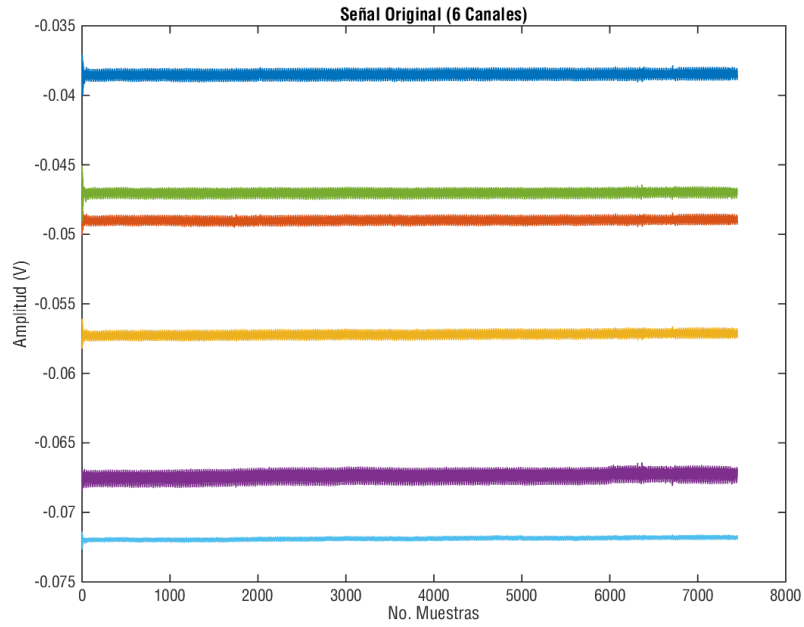


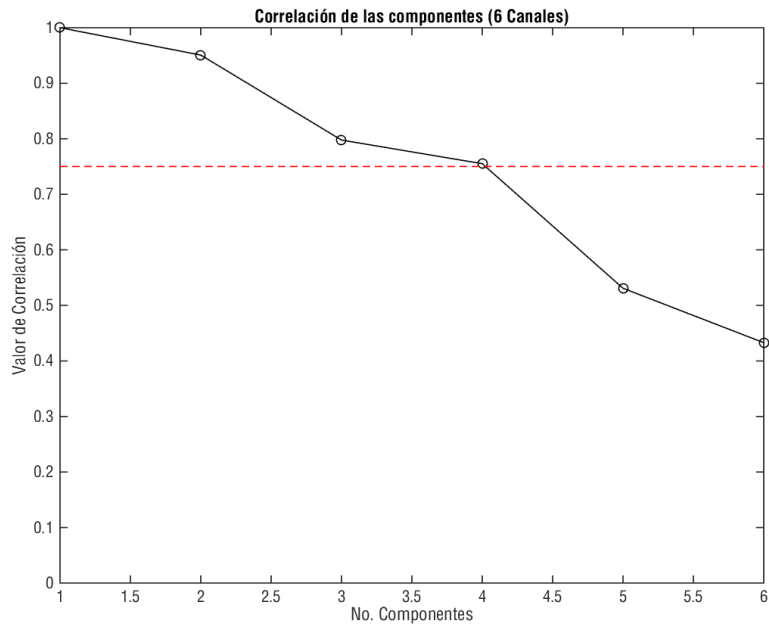
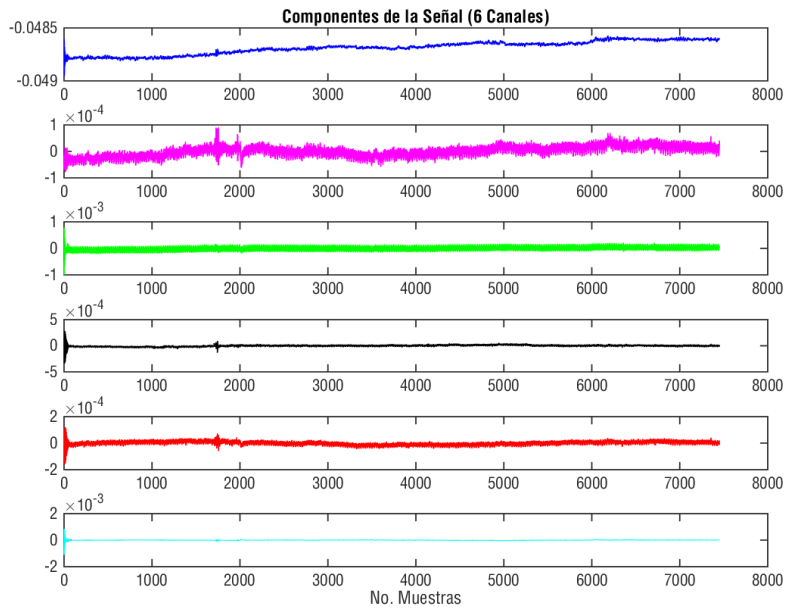


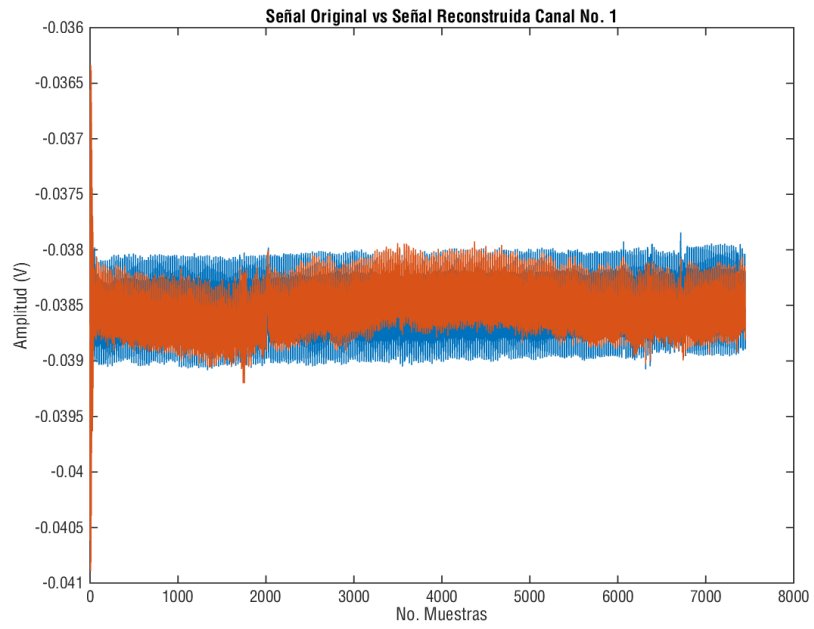
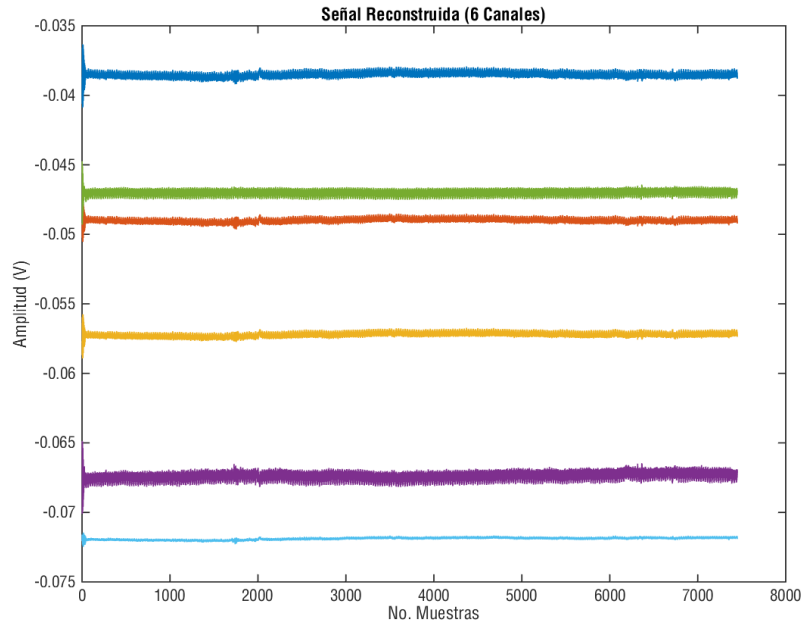


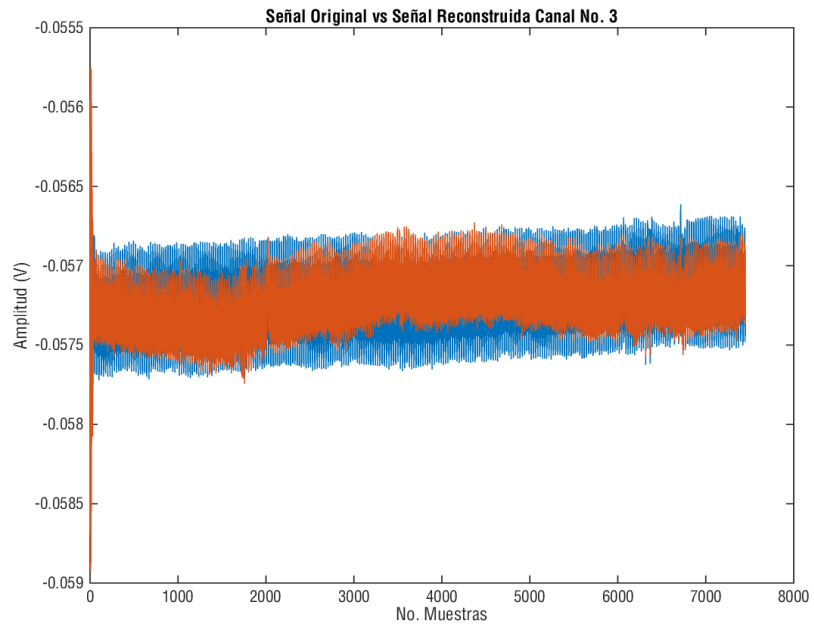
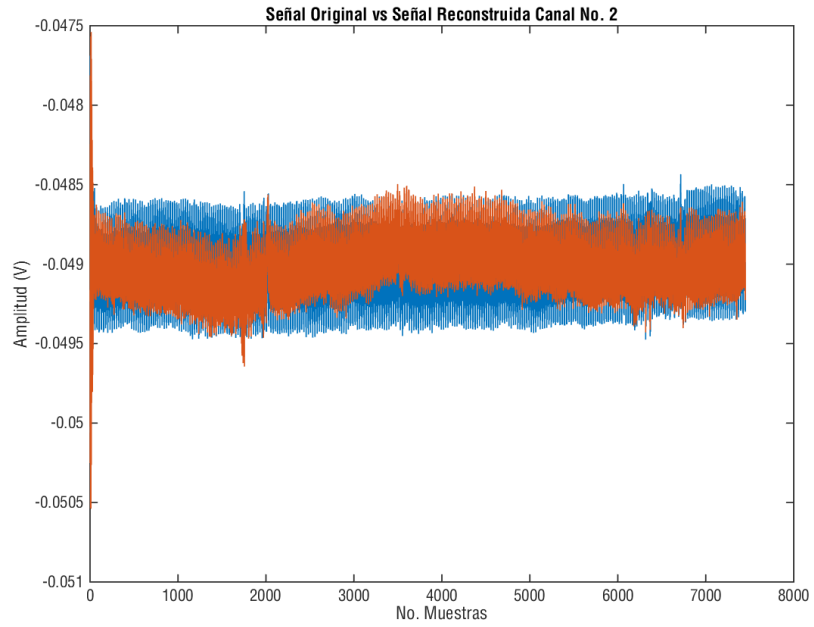


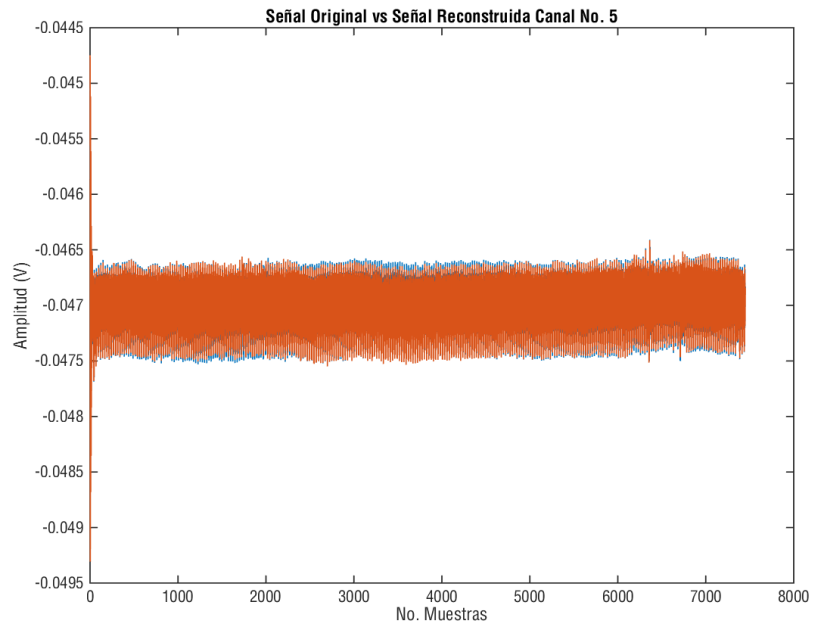
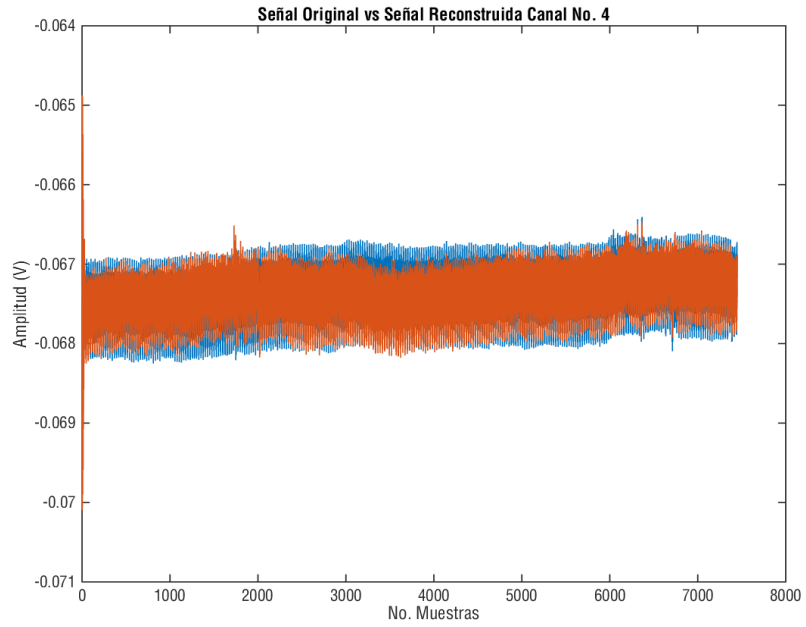
Sujeto 4. Reposo

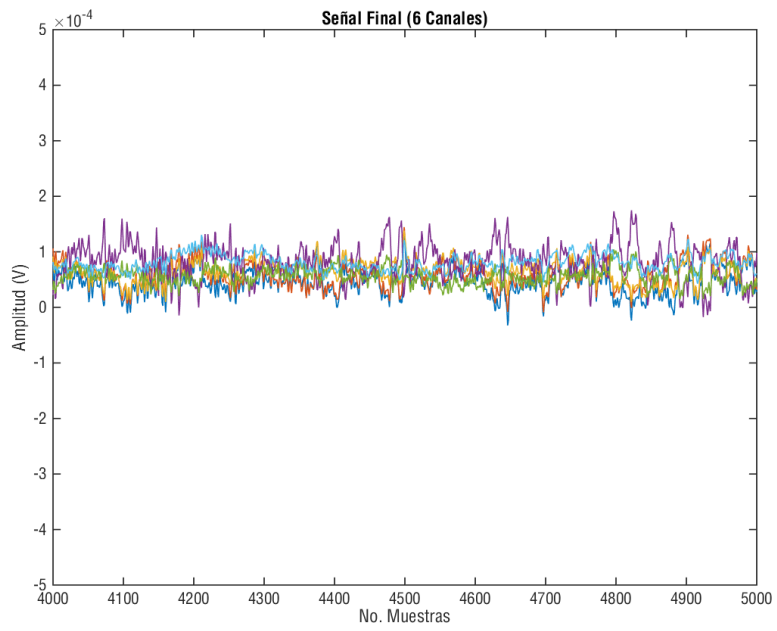
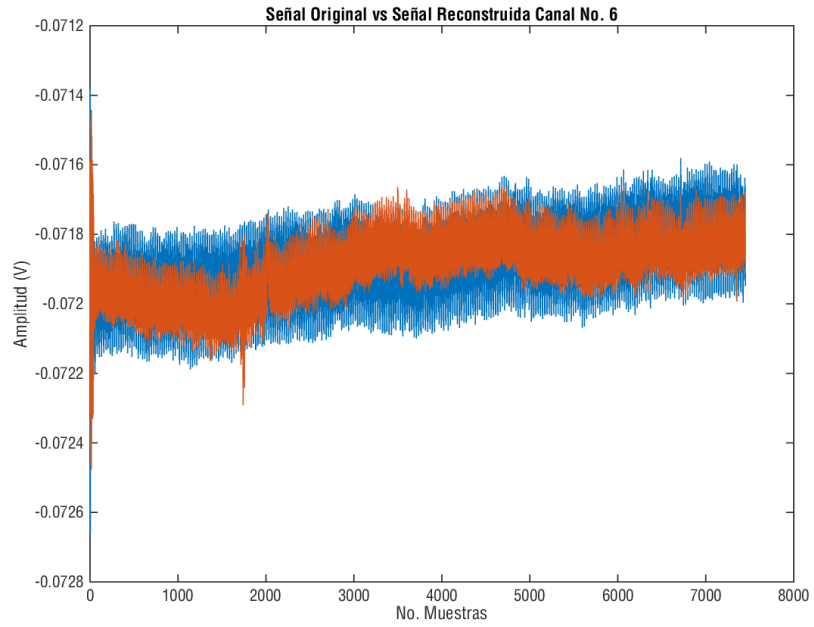




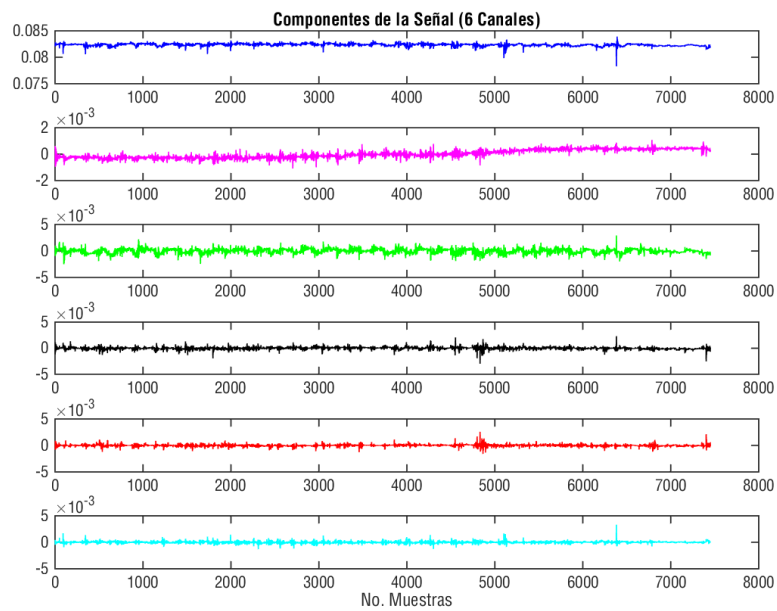
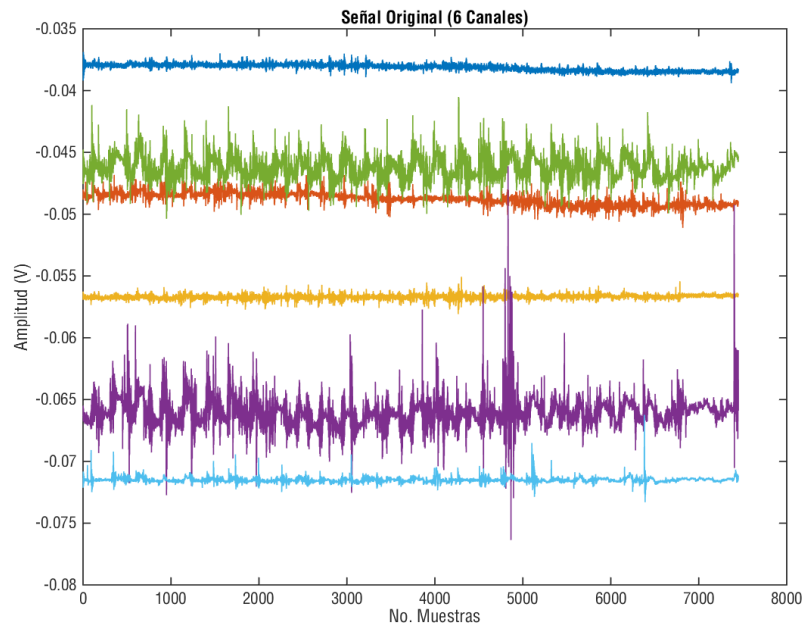


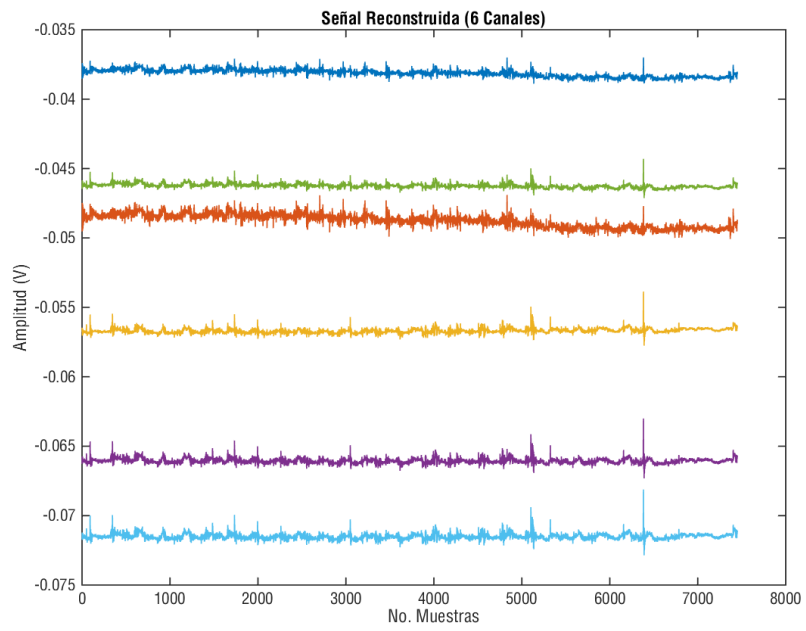
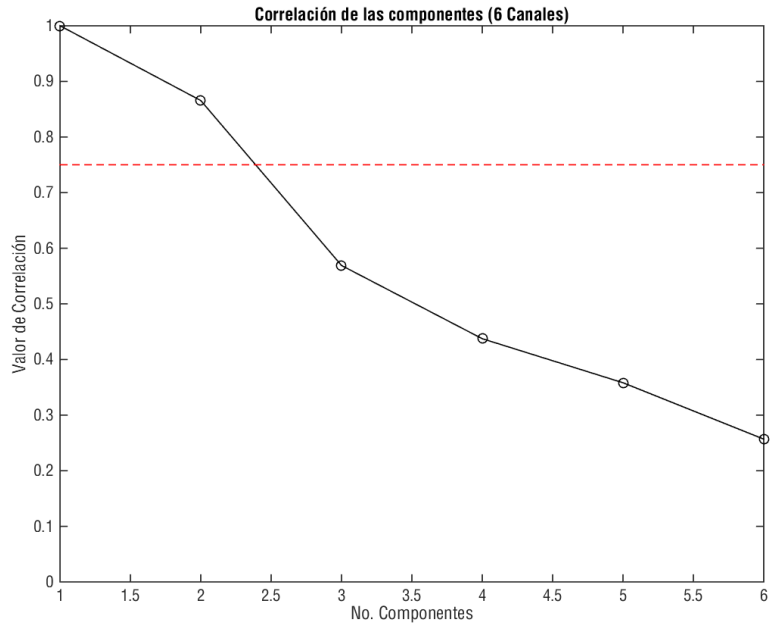


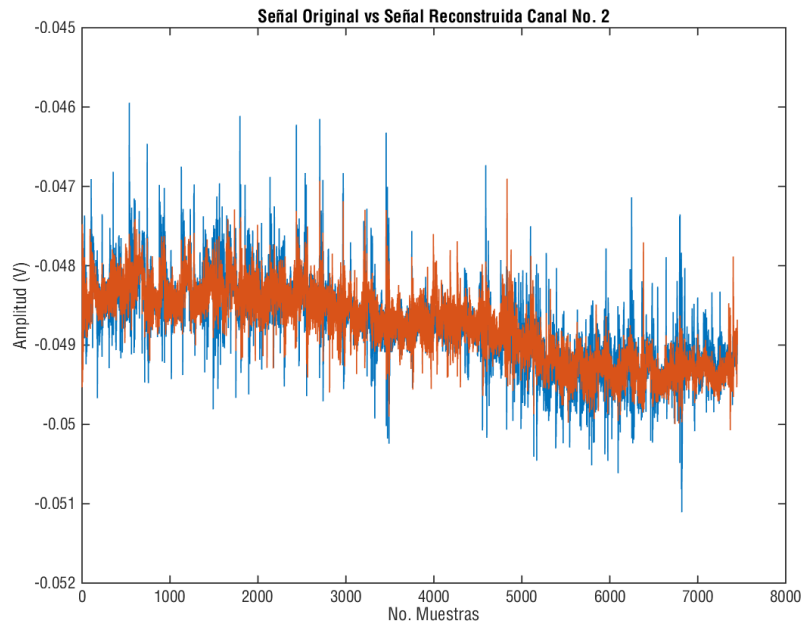
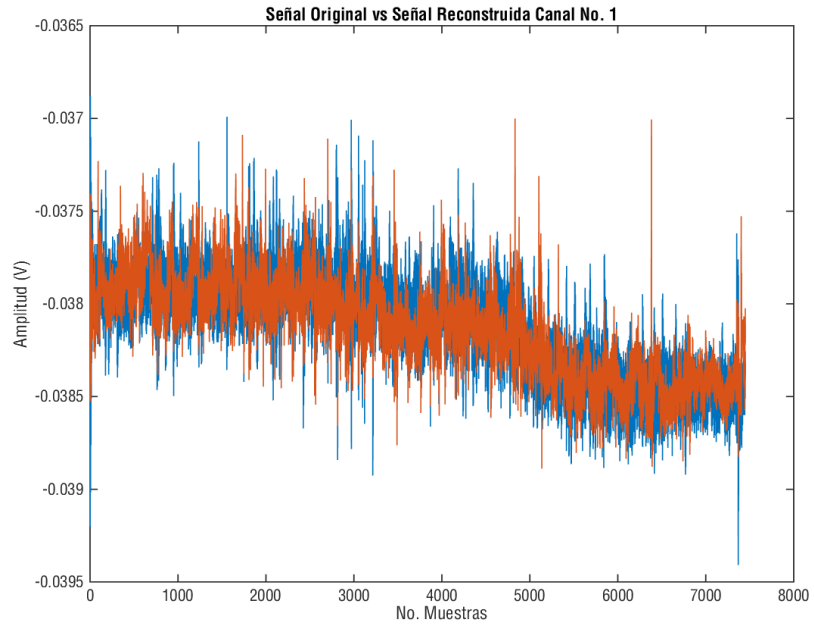


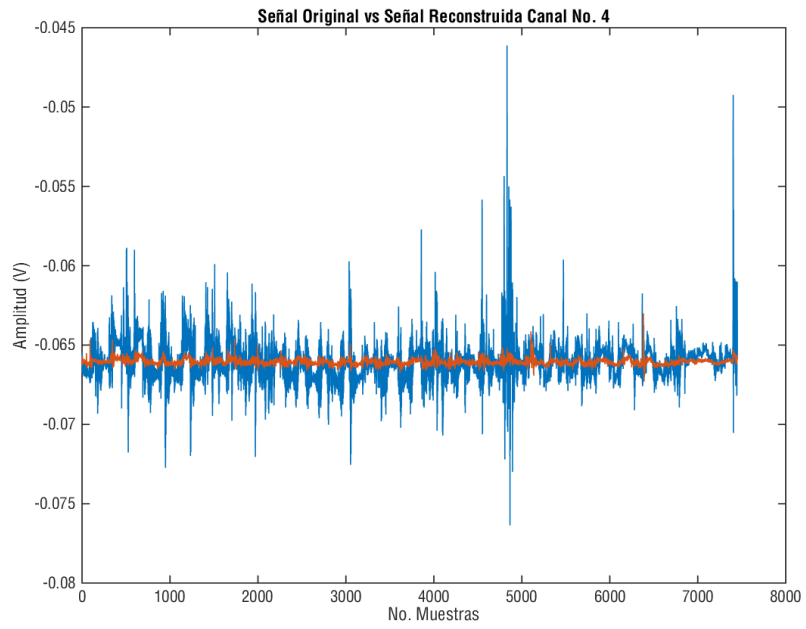
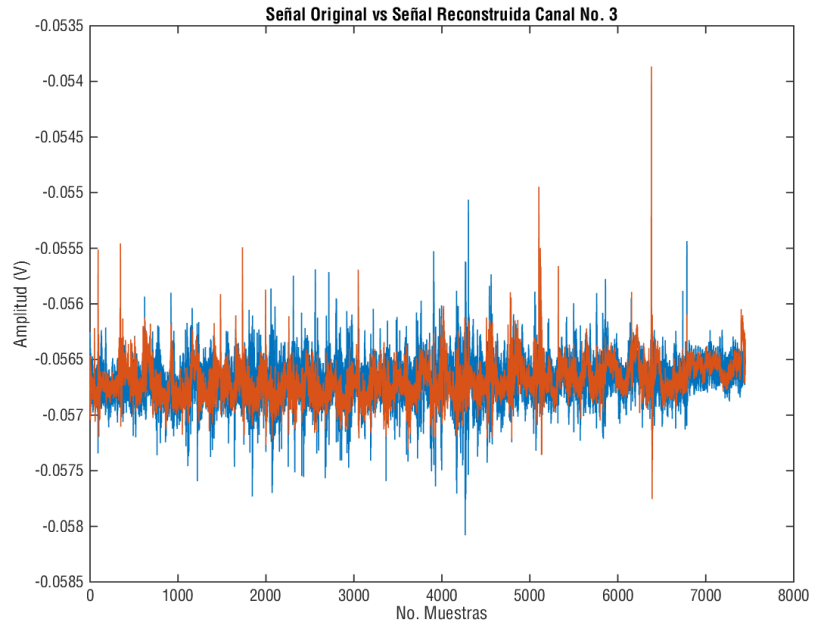


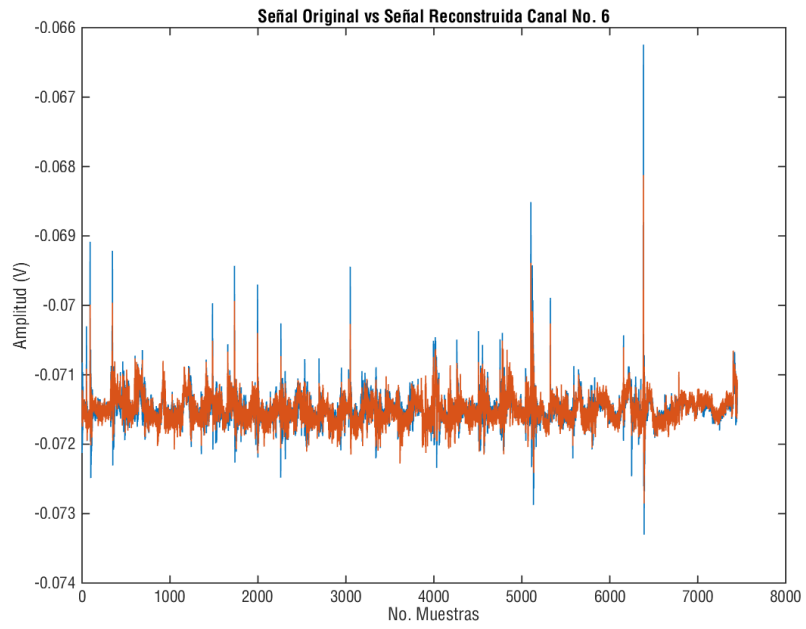
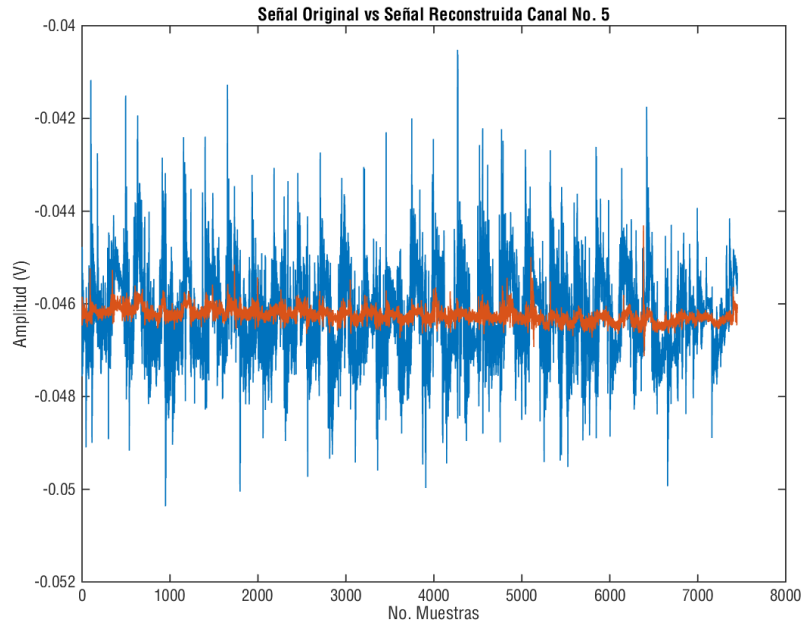
Sujeto 5. Movimiento

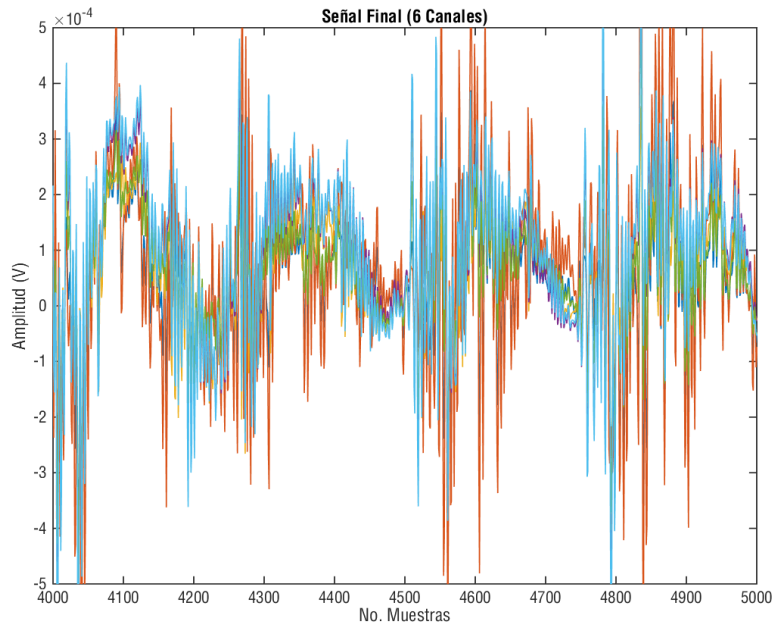




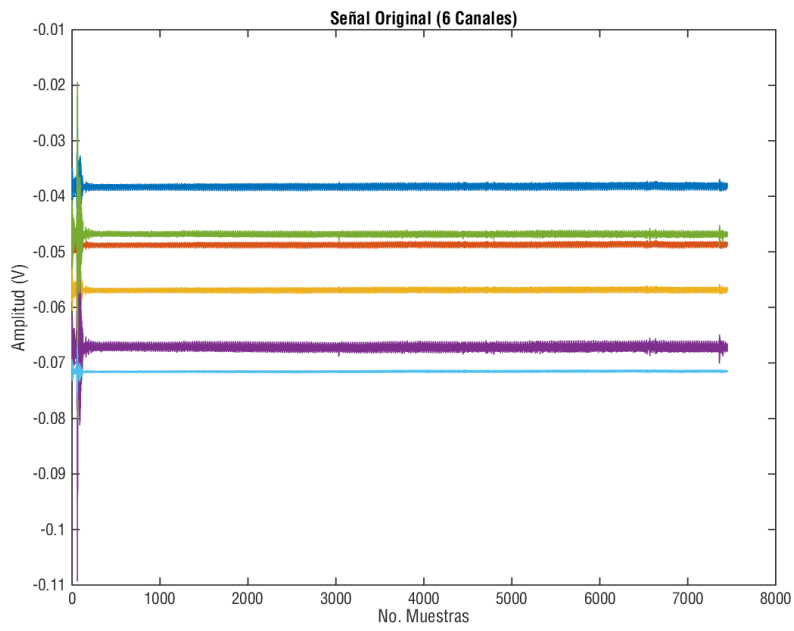


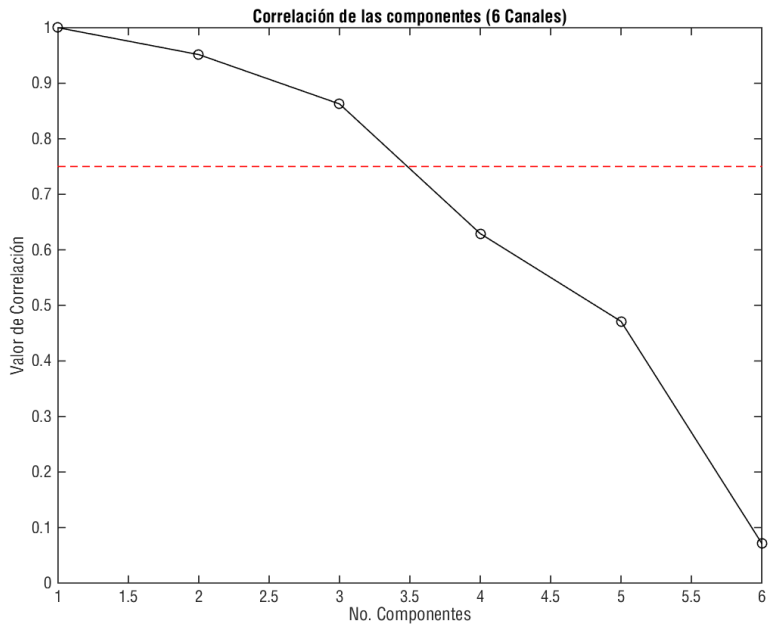
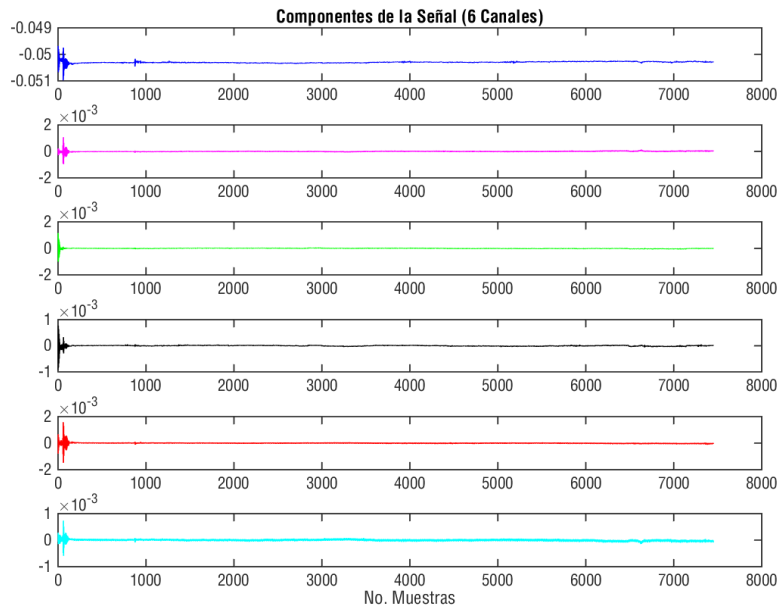


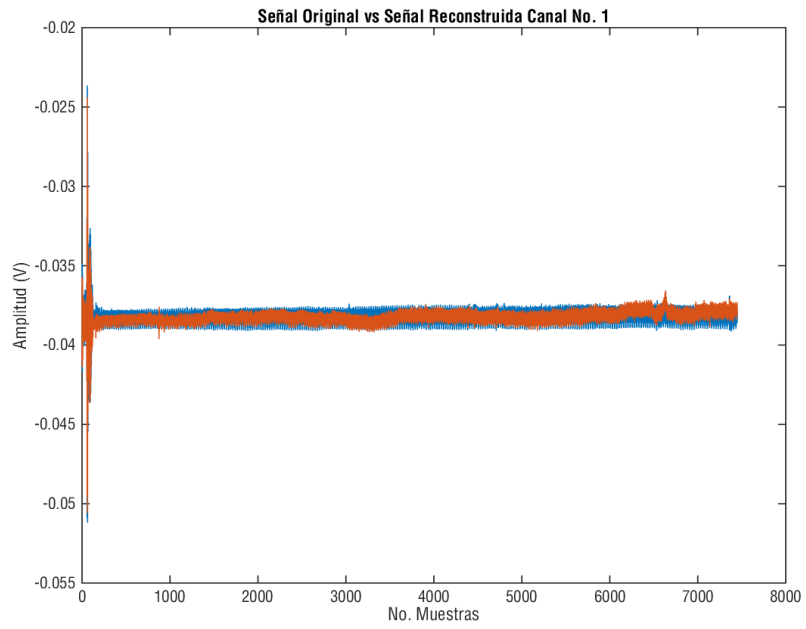
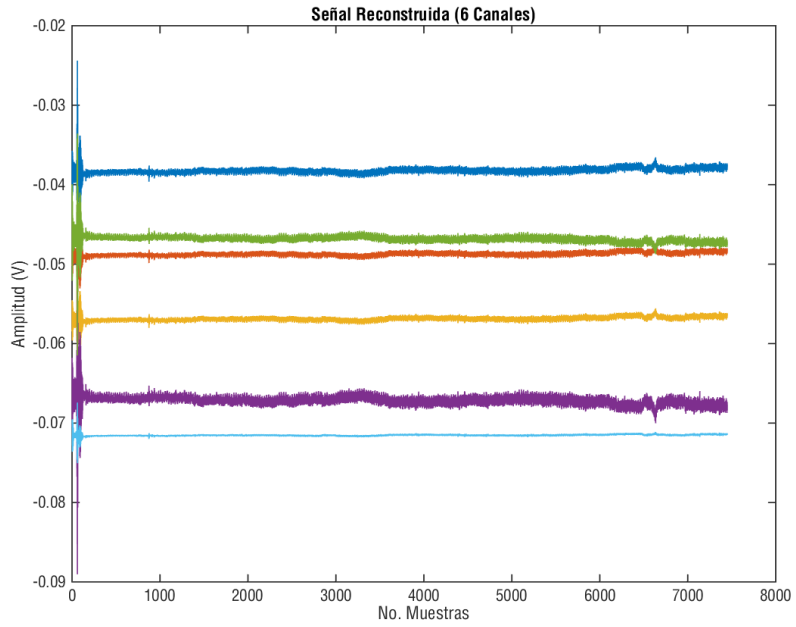


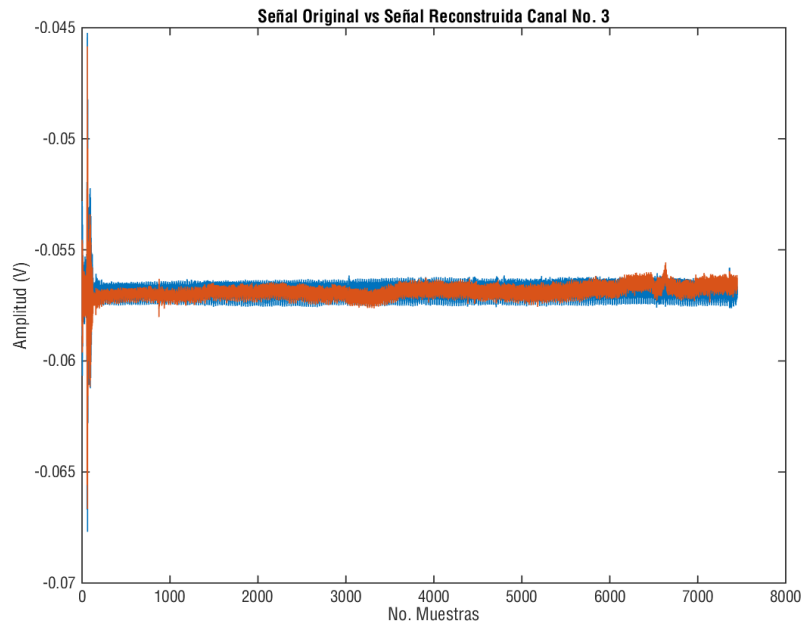
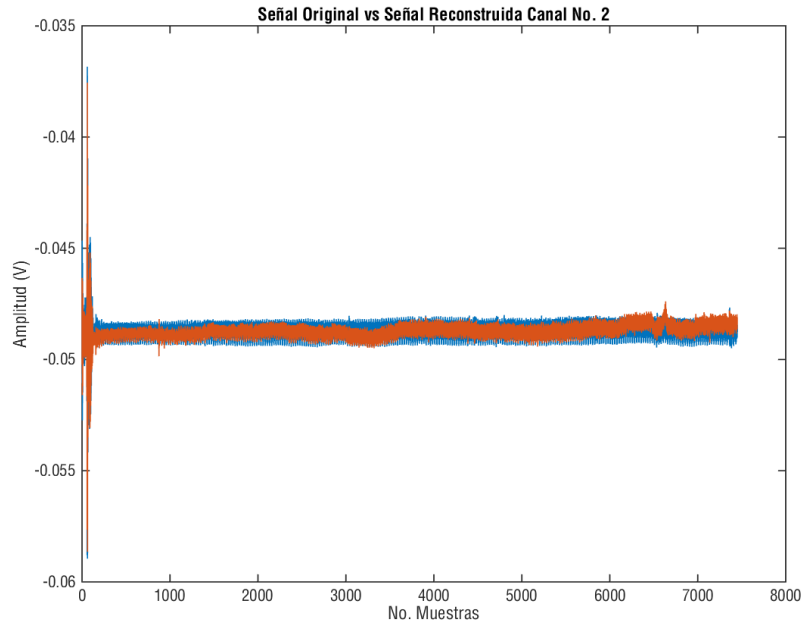


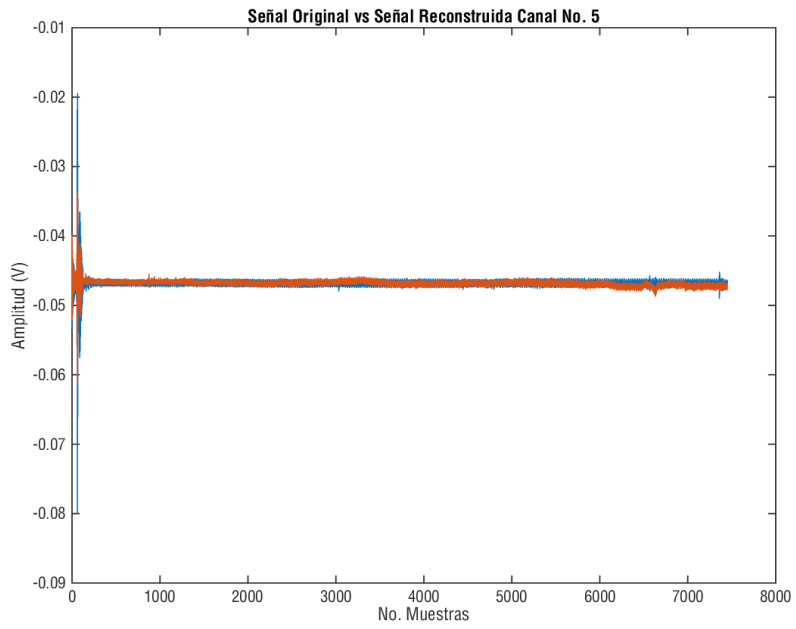
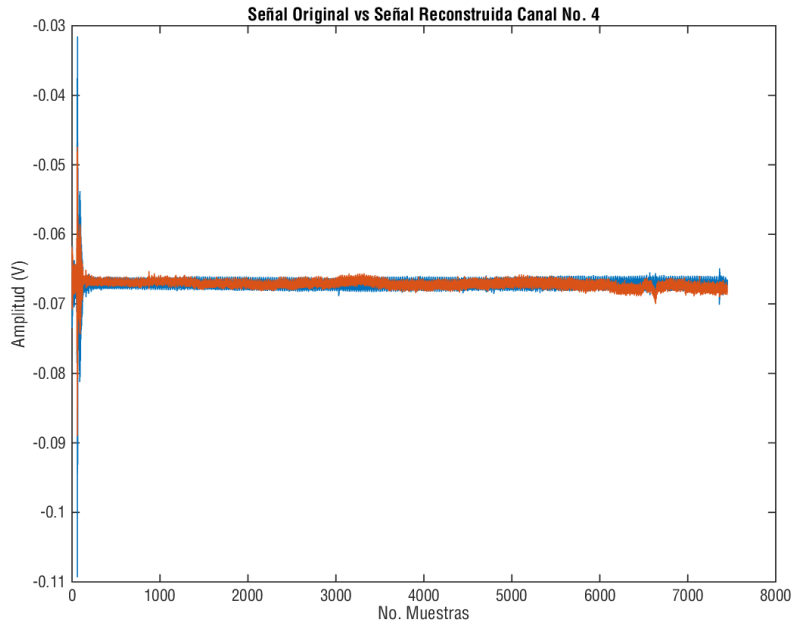
Sujeto 5. Reposo

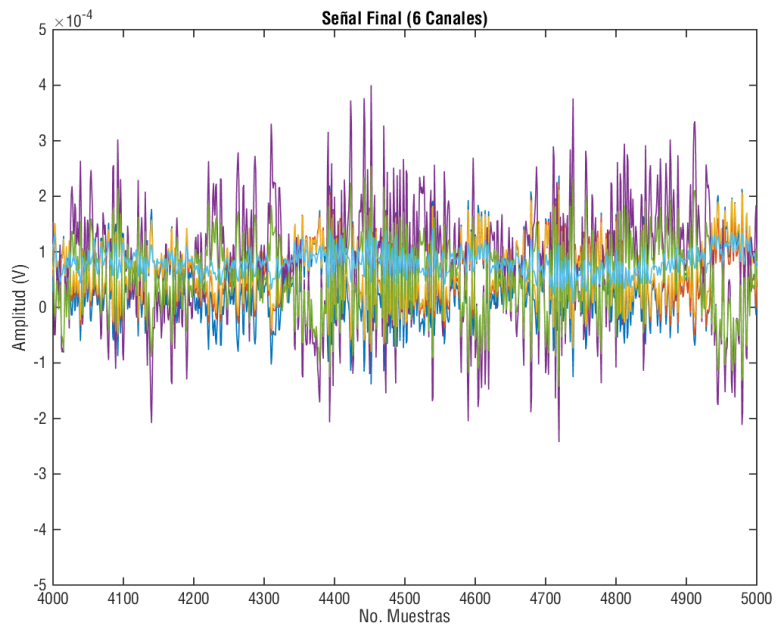
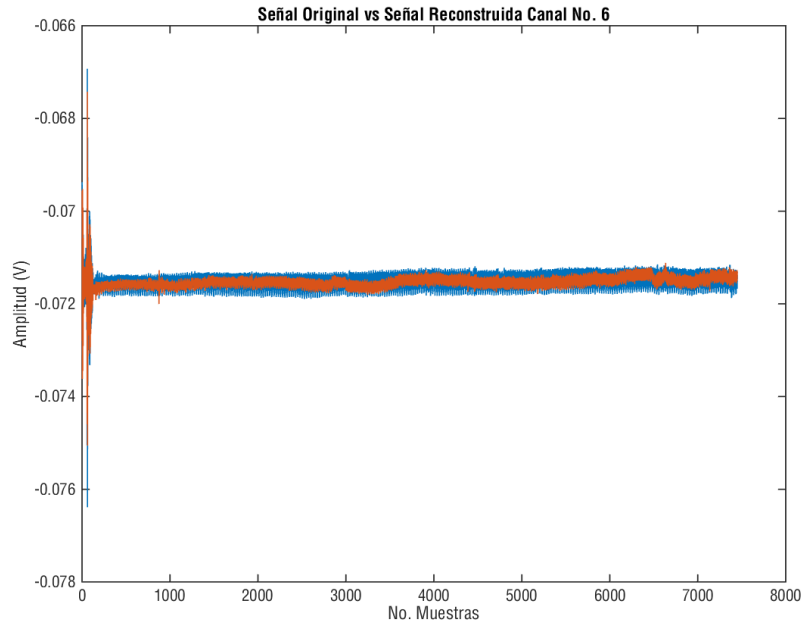












ANEXO 7.

PROCESAMIENTO DE SEÑALES EN LENGUAJE C++ PARA LA PROGRAMACIÓN DEL MICROCONTROLADOR

```
1: /*****/
2: /* Proyecto: Proyecto de Grado */
3: /* Fecha: Junio 2017 */
4: /* Autor: Ángela Patricia Guerrero C */
5:
6: /*****/
7: /* Bibliotecas a usar */
8: /*****/
9:
10: #include "mbed.h"
11: #include "ads1299.h"
12: #include <stdint.h>
13: #include <Eigen/Dense.h>
14:
15: using namespace ADS1299;
16: using namespace Eigen;
17:
18: /*****/
19: /* Definición de constantes */
20: /*****/
21:
22: #define BR 115200
23: #define LOW 0
24: #define HIGH 1
25:
26: #define N 125 // N = 125 *2; Para el submuestreo
27: #define M 6
28:
29: // #define NO_SPI // No comentar esta línea para entrar al modo de
30: // prueba sin tener en cuenta el SPI
31:
32:
33: /*****/
34: /* Definición de Salidas y Entradas */
35: /*****/
36:
37: Serial pc(SERIAL_TX, SERIAL_RX, BR); // Configuración comunicación UART
38: SPI spi(PB_15, PB_14, PB_13); // Configuración comunicación SPI
39: // El orden de la configuración es:
40: // SPI_MOSI; SPIMISO; SPI_SCK
41: // SPI_SCK; SPI_MISO; SPI_MOSI;
42: // SPI_CS
42: DigitalOut PIN_CS(PC_8,1); // Salida de START para el
43: DigitalOut PIN_START(PC_6,0); // integrado
44: // Salida de Led Indicador
45: DigitalOut LED(LED1); // Interrupción con el DRDY del
46: InterruptIn PIN_DRDY(PC_5); // integrado
47: // Es una entrada
48:
49:
50: /*****/
51: /* Prototipos */
52: /*****/
53:
54: void ads_send_command(int cmd);
55: void ads_wreg(int reg, int val);
56: int ads_rreg(int reg);
57: void data_ready(void);
58: void config_ads(void);
59: void bss_cca(void);
60: void filtro(void);
61:
```

```

62: /******
63: /* Defclaración de variables */
64: /******
65:
66: int gMaxChan,gIDval,t1,t2,t3,t4;
67: int cont,proc;
68:
69: MatrixXcd Xf(N,M);
70: const int32_t scale = ((0x7FFFFFF*24)/4.5); // max 24bit * arreglo a -1
71:
72: #ifdef NO_SPI // * gain / ref
73:
74: const double dato[N][M]= {{0.07342633, 0.862347852, 0.588721007, 0.263320171, 0.649389639, 0.110071523},
75: {0.499529333, 0.049252773, 0.815981358, 0.63465135, 0.997744981, 0.197787167},
76: {0.264698245, 0.658293782, 0.664597328, 0.930878615, 0.250238889, 0.822950583},
77: {0.465194504, 0.156189718, 0.204130975, 0.343182485, 0.85652618, 0.204758039},
78: {0.577347333, 0.568735079, 0.258402956, 0.651720792, 0.27245652, 0.487977313},
79: {0.878825841, 0.555185639, 0.052515704, 0.869149167, 0.234813111, 0.80886169},
80: {0.949023036, 0.9070019, 0.884835631, 0.794277719, 0.609666949, 0.724626224},
81: {0.729342668, 0.265481051, 0.299987242, 0.586278086, 0.666359445, 0.98768237},
82: {0.62336883, 0.797879561, 0.764268549, 0.266366621, 0.799157818, 0.749614666},
83: {0.736034126, 0.838737731, 0.075339364, 0.114167586, 0.737106861, 0.47891211},
84: {0.956563912, 0.485570635, 0.616626028, 0.319058569, 0.267866663, 0.025203833},
85: {0.4506313, 0.19193328, 0.641599883, 0.892397424, 0.04350945, 0.966286471},
86: {0.13622599, 0.199921713, 0.90032952, 0.383805307, 0.455723552, 0.513822663},
87: {0.100151782, 0.542017804, 0.885779652, 0.301355426, 0.808873494, 0.2430314},
88: {0.198906833, 0.551119757, 0.23917395, 0.753002299, 0.804779512, 0.280618181},
89: {0.067254197, 0.599548687, 0.694029681, 0.693294996, 0.682606506, 0.989687773},
90: {0.38321451, 0.305177675, 0.285365115, 0.470765153, 0.925019879, 0.253859192},
91: {0.791328916, 0.473863023, 0.661300797, 0.065446695, 0.373637764, 0.853516424},
92: {0.659288284, 0.310624607, 0.138337344, 0.351183644, 0.037319492, 0.317717171},
93: {0.426343414, 0.032750125, 0.114782992, 0.817716851, 0.701716716, 0.933749813},
94: {0.203453395, 0.114415347, 0.488665369, 0.755480525, 0.157018738, 0.095383731},
95: {0.54777548, 0.132406861, 0.302802117, 0.348045419, 0.131475315, 0.10259063},
96: {0.607348209, 0.203270461, 0.880313062, 0.836385533, 0.371569163, 0.459845707},
97: {0.913437015, 0.326336135, 0.284993839, 0.794142829, 0.673341598, 0.600289627},
98: {0.159968057, 0.762456521, 0.433252276, 0.407543671, 0.893765638, 0.038218852},
99: {0.638794262, 0.321549626, 0.507666302, 0.825761223, 0.134350499, 0.172514845},
100: {0.915908373, 0.295909487, 0.940014606, 0.836779389, 0.36048879, 0.217989055},
101: {0.556719351, 0.659545236, 0.369451601, 0.382554089, 0.180204564, 0.720098999},
102: {0.276968443, 0.019099813, 0.806502143, 0.46248803, 0.204722742, 0.328553344},
103: {0.933483232, 0.676092581, 0.322499004, 0.4456205, 0.157800133, 0.797338573},
104: {0.984052295, 0.830695679, 0.752957233, 0.012095765, 0.789801891, 0.74660351},
105: {0.669414424, 0.228908979, 0.489516781, 0.454601264, 0.375442314, 0.217198062},
106: {0.080353159, 0.106548111, 0.905712406, 0.804658102, 0.569200882, 0.791765109},
107: {0.896859434, 0.519527026, 0.529923095, 0.861277835, 0.755272505, 0.280368508},
108: {0.218954926, 0.308316456, 0.057414597, 0.084463751, 0.303754055, 0.501969542},
109: {0.054369407, 0.452936201, 0.167718913, 0.10598158, 0.646301841, 0.829031704},
110: {0.87188352, 0.004454842, 0.2822145, 0.417205983, 0.585250999, 0.568853898},
111: {0.798282666, 0.173283692, 0.249513489, 0.677961313, 0.441444228, 0.840785204},
112: {0.753216169, 0.594359795, 0.336116308, 0.868957176, 0.918452217, 0.89902652},
113: {0.788344337, 0.837099093, 0.605324642, 0.638986055, 0.933623807, 0.826251922},
114: {0.182076345, 0.447040201, 0.79122688, 0.779220025, 0.080704206, 0.118976922},
115: {0.653864675, 0.092020023, 0.521750872, 0.72538656, 0.5263392, 0.464205378},
116: {0.312989986, 0.509815702, 0.699018672, 0.216466444, 2.3562E-05, 0.213175962},
117: {0.642427072, 0.757598149, 0.631936993, 0.522740108, 0.887537006, 0.992305963},
118: {0.085533152, 0.381861311, 0.795153948, 0.794020579, 0.147530154, 0.704663383},
119: {0.610210796, 0.120132792, 0.601153552, 0.200957082, 0.292700501, 0.319915903},
120: {0.405658204, 0.795709401, 0.548937367, 0.109709842, 0.389173573, 0.882941749},
121: {0.785219734, 0.656098366, 0.531493781, 0.914029827, 0.559326929, 0.049801939},
122: {0.444348208, 0.943222393, 0.949367869, 0.811734179, 0.394437194, 0.195991834},
123: {0.528370527, 0.292788355, 0.430561839, 0.357344266, 0.229827489, 0.63950001},
124: {0.091505487, 0.708056104, 0.951468323, 0.717625737, 0.455815398, 0.277193219},
125: {0.574652465, 0.115907177, 0.861155071, 0.429685393, 0.106977759, 0.560163674},
126: {0.106753656, 0.242586008, 0.203410559, 0.892387042, 0.420385811, 0.031316184},
127: {0.231938158, 0.160978362, 0.39732094, 0.314204937, 0.425503035, 0.24984055},
128: {0.687358756, 0.34990516, 0.178305422, 0.277417512, 0.111434964, 0.713561856},
129: {0.574569255, 0.387343349, 0.602852735, 0.439308468, 0.730038169, 0.050603209},

```


130: {0.402480806, 0.096618452, 0.365084212, 0.186692812, 0.435933515, 0.268973373},
131: {0.342456716, 0.229886618, 0.183600086, 0.544731596, 0.807245002, 0.569899195},
132: {0.883955093, 0.177616782, 0.728267299, 0.337787061, 0.511901615, 0.488380141},
133: {0.029512776, 0.640853809, 0.915599863, 0.472666429, 0.708011944, 0.31497973},
134: {0.69771995, 0.72955413, 0.987765601, 0.909272501, 0.639930712, 0.180814754},
135: {0.741924991, 0.181498535, 0.962986461, 0.197721939, 0.7235113, 0.934212362},
136: {0.891291057, 0.991139785, 0.011314113, 0.853753433, 0.51759756, 0.472366385},
137: {0.147941333, 0.768158398, 0.94726265, 0.594451515, 0.298282094, 0.20603126},
138: {0.655776331, 0.166167447, 0.037070961, 0.628808202, 0.7008214, 0.560487451},
139: {0.353627782, 0.040246589, 0.03280745, 0.653388589, 0.488476171, 0.819521429},
140: {0.478732929, 0.48045269, 0.768902076, 0.035582476, 0.597673623, 0.034716498},
141: {0.626229171, 0.38665997, 0.836795281, 0.508003652, 0.609651256, 0.673007761},
142: {0.160497697, 0.842904061, 0.513151156, 0.165599513, 0.322229428, 0.193699415},
143: {0.40945889, 0.993661672, 0.687003863, 0.033907459, 0.120486994, 0.413818267},
144: {0.757275314, 0.0634698, 0.53341632, 0.7626176, 0.428773098, 0.895101833},
145: {0.189885109, 0.925490235, 0.460823297, 0.851412051, 0.41029553, 0.908705539},
146: {0.465262393, 0.790186474, 0.02936829, 0.815766016, 0.706013901, 0.424907363},
147: {0.94153792, 0.847073076, 0.871810583, 0.804741971, 0.412208788, 0.835814306},
148: {0.76423718, 0.756298346, 0.63045982, 0.930404399, 0.96937684, 0.924536896},
149: {0.187814406, 0.243794592, 0.206734014, 0.56379102, 0.95415046, 0.720178386},
150: {0.02570813, 0.094874773, 0.668009136, 0.670073624, 0.262709093, 0.312307944},
151: {0.188427291, 0.324341989, 0.222264646, 0.543261222, 0.459706377, 0.653305958},
152: {0.415999745, 0.563709307, 0.524257227, 0.052896007, 0.43889249, 0.566377639},
153: {0.086336215, 0.634209216, 0.797430086, 0.639744402, 0.103272679, 0.783554788},
154: {0.582808652, 0.273171713, 0.057235131, 0.362360428, 0.60930256, 0.448521162},
155: {0.941839074, 0.391485471, 0.547089516, 0.594771559, 0.733486298, 0.885940493},
156: {0.936841087, 0.730130977, 0.160499967, 0.210112375, 0.439651808, 0.38792988},
157: {0.071757395, 0.800479501, 0.53479445, 0.783757654, 0.338691784, 0.623154207},
158: {0.861928452, 0.939095065, 0.344660473, 0.983110241, 0.446703028, 0.971850585},
159: {0.434411307, 0.338973787, 0.733965874, 0.949233136, 0.726483303, 0.780523711},
160: {0.528110415, 0.808541192, 0.576391658, 0.64850212, 0.110432034, 0.601906503},
161: {0.288393182, 0.541847295, 0.674750044, 0.386853938, 0.875746681, 0.706764429},
162: {0.348648885, 0.886835206, 0.634615939, 0.417346376, 0.19097962, 0.330946228},
163: {0.230516968, 0.831950769, 0.587179962, 0.715830844, 0.463669902, 0.678481229},
164: {0.42550704, 0.315944673, 0.705175825, 0.313099284, 0.284208915, 0.685429341},
165: {0.082625778, 0.524908309, 0.119008564, 0.406665405, 0.395085772, 0.935211715},
166: {0.975166087, 0.437650292, 0.079762471, 0.835086186, 0.497191617, 0.596432867},
167: {0.290917944, 0.902907566, 0.819847118, 0.224957073, 0.969809768, 0.602467995},
168: {0.69222432, 0.469001366, 0.001672977, 0.936984859, 0.724676152, 0.884264187},
169: {0.311823627, 0.565207243, 0.058923018, 0.607145628, 0.987026237, 0.121132216},
170: {0.584359888, 0.196998166, 0.003173289, 0.463469956, 0.493625455, 0.337945322},
171: {0.81252003, 0.977215722, 0.872153513, 0.45561, 0.892800138, 0.89390544},
172: {0.98311152, 0.002734579, 0.025742823, 0.335311002, 0.25504871, 0.778444506},
173: {0.324801066, 0.732834881, 0.186462292, 0.294097646, 0.798017752, 0.173473729},
174: {0.33059022, 0.868265762, 0.693130496, 0.662286366, 0.926249327, 0.23712029},
175: {0.55913445, 0.715546892, 0.776532163, 0.951275095, 0.813900319, 0.31349269},
176: {0.342152911, 0.927201065, 0.770866393, 0.10292043, 0.55332296, 0.854500368},
177: {0.326621577, 0.688872578, 0.521764136, 0.862990246, 0.548335497, 0.615281763},
178: {0.510064129, 0.584973934, 0.504970891, 0.409843017, 0.25231147, 0.615358808},
179: {0.389194729, 0.174469391, 0.125409514, 0.28198219, 0.103127995, 0.111140283},
180: {0.608018795, 0.549639047, 0.150036648, 0.206950968, 0.689454852, 0.490472704},
181: {0.152734141, 0.615845234, 0.172023229, 0.659798199, 0.148898181, 0.927916537},
182: {0.21092793, 0.025908376, 0.406436922, 0.524396737, 0.024804012, 0.657202126},
183: {0.301200974, 0.106665171, 0.92563217, 0.108820661, 0.994644101, 0.964091078},
184: {0.125200785, 0.22594014, 0.575732046, 0.15227697, 0.082120992, 0.869548795},
185: {0.753906627, 0.31285774, 0.455944433, 0.931911318, 0.426780576, 0.320980531},
186: {0.530865879, 0.173982276, 0.893276939, 0.725742488, 0.524894363, 0.943659263},
187: {0.138220462, 0.58374384, 0.599037665, 0.758392418, 0.668865508, 0.633647611},
188: {0.954179149, 0.703354436, 0.043509949, 0.424069331, 0.253782867, 0.120416002},
189: {0.903189195, 0.437499499, 0.319559035, 0.804671515, 0.308171864, 0.896797988},
190: {0.561044258, 0.259258214, 0.009996289, 0.404203636, 0.039253089, 0.62288259},
191: {0.184259643, 0.285306613, 0.710111836, 0.409210233, 0.791062983, 0.871092706},
192: {0.565006756, 0.495748002, 0.246550031, 0.037783253, 0.539901771, 0.543288565},
193: {0.927375332, 0.53977712, 0.8510727, 0.565443203, 0.950485982, 0.337468412},
194: {0.783811113, 0.953304641, 0.233928344, 0.282107592, 0.280779869, 0.671907537},
195: {0.800671685, 0.259057941, 0.299591641, 0.937774754, 0.430136692, 0.812299257},
196: {0.995522703, 0.57812275, 0.769743799, 0.934769092, 0.111304154, 0.300438625},
197: {0.403649308, 0.239372396, 0.798516815, 0.916989282, 0.957043864, 0.838726995},

```

198: {0.661138786, 0.321337814, 0.44302128, 0.009067316, 0.873487864, 0.81487158}};
199:
200: #else
201:
202: double dato[N][M];
203:
204: #endif
205:
206: /******
207: /* Programa principal */
208: /******
209:
210: int main() {
211: spi.format(8,1); // Configuración del modo del SPI
212:
213: pc.printf("SystemCoreClock = %d Hz\r\n", SystemCoreClock);
214: pc.printf("start = %d us\r\n", us_ticker_read());
215:
216: #ifdef NO_SPI
217:
218: proc = true;
219:
220: #else
221:
222: wait(2); // Espera de 2 segundos para
223: // iniciar el integrado.
224: config_ads(); // Configurar el integrado.
225: proc=0;
226: PIN_DRDY.fall(&data_ready); // Interrupción de flanco de
227: // bajada.
228: pc.printf("Interrupciones config\r\n");
229: t3 = us_ticker_read();
230:
231: #endif
232:
233:
234: while(1){
235: if (proc){
236: pc.printf("Tiempo de Adq: %d us\r\n", t4-t3);
237: t1= us_ticker_read();
238:
239: bss_cca(); // Implementación del mÃ©todo
240: // BSS-CCA.
241:
242: filtro(); // Implementación de un filtro
243: // butterworth de orden 8 con
244: // frecuencias de corte 0.5-35 Hz
245:
246: t2= us_ticker_read();
247: pc.printf("Tiempo de proc: %d us\r\n", t2-t1);
248:
249: while(1);
250: }
251: else{
252: LED = 1; // LED prendido
253: wait(0.1); // 200 ms
254: LED = 0; // LED apagado
255: wait(0.1); // 1 seg
256: }
257: }
258: }
259:
260: /******
261: /* Rutina Para Enviar Comandos al Integrado */
262: /******
263:
264: void ads_send_command(int cmd)
265: {

```

```

266: PIN_CS = LOW;
267: spi.write(cmd);
268: wait_us(1);
269: PIN_CS = HIGH;
270: }
271:
272: /******
273:  /* Rutina Para Escribir en los Registros del Integrado */
274:  /******
275:
276: void ads_wreg(int reg, int val)
277: {
278: PIN_CS = LOW;
279: spi.write(WREG | reg);
280: spi.write(0);
281: spi.write(val);
282: wait_us(5);
283: PIN_CS = HIGH;
284: }
285:
286: /******
287:  /* Rutina Para Leer los Registros del Integrado */
288:  /******
289:
290: int ads_rreg(int reg){
291: volatile int out = 0;
292: PIN_CS = LOW;
293: spi.write(RREG | reg);
294: wait_us(5);
295: spi.write(0);
296: wait_us(5);
297: out = spi.write(0);
298: wait_us(5);
299: PIN_CS = HIGH;
300: return(out);
301: }
302:
303: /******
304:  /* Rutina Para Captar los Datos del Integrado y transformarlos en legibles */
305:  /******
306: #ifndef NO_SPI
307: void data_ready(void){
308:
309: t1= us_ticker_read();
310: volatile int32_t datos[M+1];
311: int i=0;
312: static int samples=0;
313: static int sps;
314: static int sps=250;
315: volatile int d0,d1,d2;
316: if(!proc){
317:
318: // Adquisición de la Señal
319: PIN_CS = LOW;
320: wait_us(5);
321: for(i=0;i<(gMaxChan+1);i++){
322: d2 = spi.write(0) & 0x00FF;
323: d1 = spi.write(0) & 0x00FF;
324: d0 = spi.write(0) & 0x00FF;
325: if(d2>0x7f)
326: datos[i] = ((d2<<16)|(d1<<8)|(d0))|0xFF000000;
327: else
328: datos[i] = (d2<<16)|(d1<<8)|(d0);
329: wait_us(5);
330: }
331: PIN_CS = HIGH;
332:
333: // Submuestreo de la Señal

```

```

334: sps=0;
335: for(i=0;i<(M+1);i=i+2){
336: dato[samples][sps] = ((double)datos[i]);
337: sps++;
338: }
339:
340:
341: // Acondicionamiento de la Señal
342: for(i=2;i<(M+2);i++){
343: dato[samples][i-2] = ((double)dato[samples][i-1])/scale*1e6;
344: }
345: samples++;
346:
347: if(samples==N){
348: samples=0;
349: proc=1;
350: PIN_DRDY.fall(NULL);
351: t4 = us_ticker_read();
352: }
353: }
354: t2= us_ticker_read();
355: }
356: #endif
357: /******
358: /* Rutina Para Configurar el Integrado */
359: /******
360:
361: void config_ads(void){
362: ads_send_command(SDATAC); // Enviar Comando SDATAC
363: wait_ms(10);
364:
365: // Determinar el número de modelo del integrado y la
366: // cantidad de canales disponibles.
367:
368: gIDval = ads_rreg(ID);
369: switch (gIDval & 0b00011111 ) {
370: case 0b10000: // 16
371: gMaxChan = 4; // ads1294
372: break;
373: case 0b10001: // 17
374: gMaxChan = 6; // ads1296
375: break;
376: case 0b10010: // 18
377: gMaxChan = 8; // ads1298
378: break;
379: case 0b11110: // 30
380: gMaxChan = 8; // ads1299
381: break;
382: default:
383: gMaxChan = 0;
384: }
385: pc.printf("ID:%X Canal:%d \r\n",gIDval,gMaxChan);
386:
387: if((gIDval & 0b00011111) == 0x1E){
388: pc.printf("Tiempo de Conf\r\n");
389: ads_wreg(CONFIG3, 0xE0);
390: wait_ms(10);
391: ads_wreg(CONFIG1, HIGH_RES_250_SPS);
392: wait_us(5);
393: ads_wreg(CONFIG2, 0xC0);
394: wait_us(5);
395: ads_wreg(MISC1, 0x00);
396: wait_us(5);
397: int i=0;
398: pc.printf("Registros config\r\n");
399: for (i=0;i<gMaxChan;i++){
400: ads_wreg((CH1SET+i), 0x50);
401: wait_us(5);

```

```

402: }
403: pc.printf("Canales configur\n");
404: PIN_START = 1;
405: wait_us(5);
406: ads_send_command(RDATAC);
407: wait_us(5);
408: pc.printf("Conf = %d us\r\n", us_ticker_read());
409: }else{
410: pc.printf("Error\r\n");
411: while(1);
412: }
413: }
414:
415: /******
416: /* Rutina Para Efectuar el Método BSS - CCA */
417: /******
418:
419: void bss_cca(void){
420: volatile int i,j,d,v;
421: d=1;
422:
423: // Matriz Data
424: MatrixXd data(M,N);
425: for (i = 0; i < M; i = i + 1)
426: {
427: for (j = 0; j < N; j = j + 1)
428: {
429: data(i,j)=dato[j][i];
430: pc.printf("data[%d][%d]: %e\r\n",i,j,data(i,j));
431: }
432: }
433:
434: // Creación de la Y
435: MatrixXd Y(M,N-1);
436: for (i = 0; i < M; i = i + 1)
437: {
438: for (j = 0; j < N-1; j = j + 1)
439: {
440: Y(i,j)=data(i,j+d);
441: }
442: }
443:
444: // Creación de la X1
445: MatrixXd X1(M,N-1);
446: for (i = 0; i < M; i = i + 1)
447: {
448: for (j = 0; j < N-1; j = j + 1)
449: {
450: X1(i,j)=data(i,(j+1)-d);
451: }
452: }
453:
454: // Creación de Cyy = T*Y*Y'
455: MatrixXd Cyy(M,M);
456: Cyy=(1/(N*1.0))*Y*(Y.transpose());
457:
458: // Creación de Cxx = T*X1*X1'
459: MatrixXd Cxx(M,M);
460: Cxx=(1/(N*1.0))*X1*(X1.transpose());
461:
462: // Creación de Cxy = T*X1*Y'
463: MatrixXd Cxy(M,M);
464: Cxy=(1/(N*1.0))*X1*(Y.transpose());
465:
466: // Creación de Cyx = Cxy'
467: MatrixXd Cyx(M,M);
468: Cyx=(Cxy.transpose());
469:

```

```

470: // Creación de ICyy = inv(Cyy)
471: MatrixXd ICyy(M,M);
472: ICyy=Cyy.inverse();
473:
474: // Creación de ICxx = inv(Cxx)
475: MatrixXd ICxx(M,M);
476: ICxx=Cxx.inverse();
477:
478: // Matriz antes de la W y la r
479: MatrixXd Prew(M,M);
480: Prew=ICxx*Cxy*ICyy
481:
482: // Obtener Eigen valores y vectores = W, r
483: EigenSolver<MatrixXd> es(Prew);
484: MatrixXcd r = es.eigenvalues();
485: MatrixXcd W = (es.eigenvectors()).transpose();
486:
487: ArrayXd rr=r.real();
488: ArrayXd ra=rr.abs();
489: ArrayXd rs=ra.sqrt();
490:
491: // [r,l] = sort('descend');
492: MatrixXd rf(M,M);
493: for (i = 0; i < M; i++)
494: {
495: v=0;
496: for (j = 0; j < M; j++)
497: {
498: if (rs(v)>rs(j)) {
499: // No Opere
500: }
501: else // Si el numero no es el mayor
502: {
503: v=j;
504: }
505: }
506: rf(0,i)=rs(v);
507: rf(1,i)=v;
508: rs(v)=0;
509: }
510:
511: for (i=0; i<M;i++){
512: pc.printf("rf[%d][%d]: %e\r\n",i,rf(0,i));
513: }
514:
515: // W = W(:,l);
516: MatrixXcd Wo(M,M);
517: for (i=0; i<M; i++)
518: {
519: for (j=0;j<M;j++)
520: {
521: Wo(i,j)=W((rf(1,i)),j);
522: }
523: }
524:
525: // La señal estimada Y
526: MatrixXcd Yf(M,N);
527: Yf=Wo*data;
528:
529: // Eliminar componentes
530:
531: for (i=0; i < M; i++)
532: {
533: if (rf(0,i) <= 0.75) {
534: for (j=0 ; j < N; j++){
535: Yf(i,j)=0;
536: }
537: }

```

```

538: }
539:
540: // Reconstrucción de la señal
541: Xf= Wo.inverse()*Yf;
542: for (i = 0; i < M; i = i + 1)
543: {
544: for (j = 0; j < N; j = j + 1)
545: {
546: pc.printf("Xf[%d][%d]: %e\r\n",i,j,Xf(i,j));
547: }
548: }
549: }
550:
551: /******
552: /* Rutina Para Implementar el Filtro Pasabandas 0,5 - 35 Hz */
553: /******
554:
555: void filtro(void){
556: volatile int i,j,l;
557: // Coeficientes obtenidos del diseño del filtro en MATLAB(R)
558: const double a [1][17]= {0.0100000000000000, -0.083546964585594, 0.316603476784828, -0.727512303062062,
559: const double b [1][17]= {0, 0.000714458855952, 0.050083716153383, -0.115079161165419, -
0.571741527481106, 560:
561: // Matriz a
562: MatrixXd ae(1,17);
563: for (i = 0; i < 17; i = i + 1)
564: {
565: for (j = 0; j < 17; j = j + 1)
566: {
567: ae(i,j)=a[i][j]*100;
568: }
569: }
570:
571: // Matriz b
572: MatrixXd be(1,17);
573: for (i = 0; i < 17; i = i + 1)
574: {
575: for (j = 0; j < 17; j = j + 1)
576: {
577: be(i,j)=b[i][j];
578: }
579: }
580:
581: // Matriz sf
582:
583: MatrixXcd sf(1,141);
584: for (i = 0; i < 141; i = i + 1)
585: {
586: for (j = 0; j < 141; j = j + 1)
587: {
588: sf(i,j)=0;
589: }
590: }
591:
592: // Matriz s0
593: MatrixXcd s0(1,17);
594: for (i = 0; i < 17; i = i + 1)
595: {
596: for (j = 0; j < 17; j = j + 1)
597: {
598: s0(i,j)=0;
599: }
600: }
601:
602: // Filtrado
603: for (l=0;l<M;l++)
604: {
605: for (i=0;i<125;i++)

```

```

606: {
607: for (j=0;j<16;j++)
608: {
609: s0(0,15-j+1)=s0(0,15-j);
610: }
611: s0(0,0)=Xf(1,i);
612: sf(0,i+16)=be(0,0)*s0(0,0);
613: for (j=1;j<17;j++)
614: {
615: sf(0,i+16)=sf(0,i+16)+(be(j)*s0(0,j))-(ae(j)*sf(0,16-j+i));
616: }
617: }
618:
619: // Matriz final
620: MatrixXcd sfinal(M,N);
621: for (i = 0; i < 1; i++)
622: {
623: for (j = 0; j < 125; j++)
624: {
625: sfinal(1,j)=sf(0,j+16);
626: pc.printf("sfinal[%d][%d]: %e\n",1,j,sfinal(1,j));
627: }
628: }
629:
630: // Matriz sf
631: for (i = 0; i < 1; i = i + 1)
632: {
633: for (j = 0; j < 141; j = j + 1)
634: {
635: sf(i,j)=0;
636: }
637: }
638: // Matriz s0
639: for (i = 0; i < 1; i = i + 1)
640: {
641: for (j = 0; j < 17; j = j + 1)
642: {
643: s0(i,j)=0;
644: }
645: }
646: }
647: }

```


ANEXO 8.

LIBRERÍA “ADS1299.h”

```
1: #ifndef ADS1299_H
2: #define ADS1299_H
3:
4: #ifdef __cplusplus
5: namespace ADS1299 {
6: #endif
7:
8: enum spi_command {
9: // system commands
10: WAKEUP = 0x02,
11: STANDBY = 0x04,
12: RESET = 0x06,
13: START = 0x08,
14: STOP = 0x0a,
15:
16: // read commands
17: RDATA0 = 0x10,
18: SDATA0 = 0x11,
19: RDATA1 = 0x12,
20:
21: // register commands
22: RREG = 0x20,
23: WREG = 0x40
24: };
25:
26: enum reg {
27: // device settings
28: ID = 0x00,
29:
30: // global settings
31: CONFIG1 = 0x01,
32: CONFIG2 = 0x02,
33: CONFIG3 = 0x03,
34: LOFF = 0x04,
35:
36: // channel specific settings
37: CHnSET = 0x04,
38: CH1SET = CHnSET + 1,
39: CH2SET = CHnSET + 2,
40: CH3SET = CHnSET + 3,
41: CH4SET = CHnSET + 4,
42: CH5SET = CHnSET + 5,
43: CH6SET = CHnSET + 6,
44: CH7SET = CHnSET + 7,
45: CH8SET = CHnSET + 8,
46: BIAS_SENSP = 0x0d,
47: BIAS_SENSN = 0x0e,
48: LOFF_SENSP = 0x0f,
49: LOFF_SENSN = 0x10,
50: LOFF_FLIP = 0x11,
51:
52: // lead off status
53: LOFF_STATP = 0x12,
54: LOFF_STATN = 0x13,
55:
56: // other
57: GPIO = 0x14,
58: MISC1 = 0x15,
59: MISC2 = 0x16,
60: CONFIG4 = 0x17,
61: // OTHER BOARDS
62: WCT1 = 0x18,
63: WCT2 = 0x19
64: };
65:
66: enum ID_bits {
67: DEV_ID7 = 0x80,
68: DEV_ID6 = 0x40,
69: DEV_ID5 = 0x20,
70: DEV_ID2 = 0x04,
71: DEV_ID1 = 0x02,
72: DEV_ID0 = 0x01,
73:
74: ID_const = 0x10,
75: ID_ADS129x = DEV_ID7,
76: ID_ADS129xR = (DEV_ID7 | DEV_ID6),
77:
78: ID_4CHAN = 0,
79: ID_6CHAN = DEV_ID0,
80: ID_8CHAN = DEV_ID1,
81:
82: ID_ADS1294 = (ID_ADS129x | ID_4CHAN),
83: ID_ADS1296 = (ID_ADS129x | ID_6CHAN),
84: ID_ADS1298 = (ID_ADS129x | ID_8CHAN),
85: ID_ADS1294R = (ID_ADS129xR | ID_4CHAN),
86: ID_ADS1296R = (ID_ADS129xR | ID_6CHAN),
87: ID_ADS1298R = (ID_ADS129xR | ID_8CHAN)
88: };
89:
90: enum CONFIG1_bits {
91: HR = 0x90,
92: DAISY_EN = 0x40,
93: CLK_EN = 0x20,
94: DR2 = 0x04,
95: DR1 = 0x02,
96: DR0 = 0x01,
97:
98: CONFIG1_const = 0x90,
99: HIGH_RES_16k_SPS = HR,
100: HIGH_RES_8k_SPS = (HR | DR0),
101: HIGH_RES_4k_SPS = (HR | DR1),
102: HIGH_RES_2k_SPS = (HR | DR1 | DR0),
103: HIGH_RES_1k_SPS = (HR | DR2),
104: HIGH_RES_500_SPS = (HR | DR2 | DR0),
105: HIGH_RES_250_SPS = (HR | DR2 | DR1)
106: };
107:
108: enum CONFIG2_bits {
109: INT_CAL = 0x10,
110: CAL_AMP = 0x04,
111: CAL_FREQ1 = 0x02,
112: CAL_FREQ0 = 0x01,
113:
114: CONFIG2_const = 0xc0
115: };
116:
117: enum CONFIG3_bits {
118: PD_REFBUF = 0x80,
119: BIAS_MEAS = 0x10,
120: BIASREF_INT = 0x08,
```

```

121: PD_BIAS = 0x04,
122: BIAS_LOFF_SENS = 0x02,
123: BIAS_STAT = 0x01,
124:
125: CONFIG3_const = 0x60
126: };
127:
128: enum LOFF_bits {
129: COMP_TH2 = 0x80,
130: COMP_TH1 = 0x40,
131: COMP_TH0 = 0x20,
132: ILEAD_OFF1 = 0x08,
133: ILEAD_OFF0 = 0x04,
134: FLEAD_OFF1 = 0x02,
135: FLEAD_OFF0 = 0x01,
136:
137: LOFF_const = 0x00,
138:
139: COMP_TH_95 = 0x00,
140: COMP_TH_92_5 = COMP_TH0,
141: COMP_TH_90 = COMP_TH1,
142: COMP_TH_87_5 = (COMP_TH1 | COMP_TH0),
143: COMP_TH_85 = COMP_TH2,
144: COMP_TH_80 = (COMP_TH2 | COMP_TH0),
145: COMP_TH_75 = (COMP_TH2 | COMP_TH1),
146: COMP_TH_70 = (COMP_TH2 | COMP_TH1 |
COMP_TH0),
147:
148: ILEAD_OFF_6nA = 0x00,
149: ILEAD_OFF_24nA = ILEAD_OFF0,
150: ILEAD_OFF_6uA = ILEAD_OFF1,
151: ILEAD_OFF_24uA = (ILEAD_OFF1 | ILEAD_OFF0),
152:
153: FLEAD_OFF_AC = FLEAD_OFF0,
154: FLEAD_OFF_DC = (FLEAD_OFF1 | FLEAD_OFF0)
155: };
156:
157: enum CHnSET_bits {
158: PDn = 0x80,
159: GAINn2 = 0x40,
160: GAINn1 = 0x20,
161: GAINn0 = 0x10,
162: SRB2 = 0x08,
163: MUXn2 = 0x04,
164: MUXn1 = 0x02,
165: MUXn0 = 0x01,
166:
167: CHnSET_const = 0x00,
168:
169: GAIN_1X = 0x00,
170: GAIN_2X = GAINn0,
171: GAIN_4X = GAINn1,
172: GAIN_6X = (GAINn1 | GAINn0),
173: GAIN_8X = GAINn2,
174: GAIN_12X = (GAINn2 | GAINn0),
175: GAIN_24X = (GAINn2 | GAINn1),
176:
177: ELECTRODE_INPUT = 0x00,
178: SHORTED = MUXn0,
179: BIAS_INPUT = MUXn1,
180: MVDD = (MUXn1 | MUXn0),
181: TEMP = MUXn2,
182: TEST_SIGNAL = (MUXn2 | MUXn0),
183: BIAS_DRP = (MUXn2 | MUXn1),
184: BIAS_DRN = (MUXn2 | MUXn1 | MUXn0)
185: };
186:
187: enum CH1SET_bits {
188: PD_1 = 0x80,
189: GAIN12 = 0x40,
190: GAIN11 = 0x20,
191: GAIN10 = 0x10,
192: MUX12 = 0x04,
193: MUX11 = 0x02,
194: MUX10 = 0x01,
195:
196: CH1SET_const = 0x00
197: };
198:
199: enum CH2SET_bits {
200: PD_2 = 0x80,
201: GAIN22 = 0x40,
202: GAIN21 = 0x20,
203: GAIN20 = 0x10,
204: MUX22 = 0x04,
205: MUX21 = 0x02,
206: MUX20 = 0x01,
207:
208: CH2SET_const = 0x00
209: };
210:
211: enum CH3SET_bits {
212: PD_3 = 0x80,
213: GAIN32 = 0x40,
214: GAIN31 = 0x20,
215: GAIN30 = 0x10,
216: MUX32 = 0x04,
217: MUX31 = 0x02,
218: MUX30 = 0x01,
219:
220: CH3SET_const = 0x00
221: };
222:
223: enum CH4SET_bits {
224: PD_4 = 0x80,
225: GAIN42 = 0x40,
226: GAIN41 = 0x20,
227: GAIN40 = 0x10,
228: MUX42 = 0x04,
229: MUX41 = 0x02,
230: MUX40 = 0x01,
231:
232: CH4SET_const = 0x00
233: };
234:
235: enum CH5SET_bits {
236: PD_5 = 0x80,
237: GAIN52 = 0x40,
238: GAIN51 = 0x20,
239: GAIN50 = 0x10,
240: MUX52 = 0x04,
241: MUX51 = 0x02,
242: MUX50 = 0x01,
243:
244: CH5SET_const = 0x00
245: };
246:
247: enum CH6SET_bits {
248: PD_6 = 0x80,
249: GAIN62 = 0x40,
250: GAIN61 = 0x20,
251: GAIN60 = 0x10,
252: MUX62 = 0x04,
253: MUX61 = 0x02,
254: MUX60 = 0x01,
255:

```

```

256: CH6SET_const = 0x00
257: };
258:
259: enum CH7SET_bits {
260: PD_7 = 0x80,
261: GAIN72 = 0x40,
262: GAIN71 = 0x20,
263: GAIN70 = 0x10,
264: MUX72 = 0x04,
265: MUX71 = 0x02,
266: MUX70 = 0x01,
267:
268: CH7SET_const = 0x00
269: };
270:
271: enum CH8SET_bits {
272: PD_8 = 0x80,
273: GAIN82 = 0x40,
274: GAIN81 = 0x20,
275: GAIN80 = 0x10,
276: MUX82 = 0x04,
277: MUX81 = 0x02,
278: MUX80 = 0x01,
279:
280: CH8SET_const = 0x00
281: };
282:
283: enum RLD_SENSP_bits {
284: RLD8P = 0x80,
285: RLD7P = 0x40,
286: RLD6P = 0x20,
287: RLD5P = 0x10,
288: RLD4P = 0x08,
289: RLD3P = 0x04,
290: RLD2P = 0x02,
291: RLD1P = 0x01,
292:
293: RLD_SENSP_const = 0x00
294: };
295:
296: enum RLD_SENSN_bits {
297: RLD8N = 0x80,
298: RLD7N = 0x40,
299: RLD6N = 0x20,
300: RLD5N = 0x10,
301: RLD4N = 0x08,
302: RLD3N = 0x04,
303: RLD2N = 0x02,
304: RLD1N = 0x01,
305:
306: RLD_SENSN_const = 0x00
307: };
308:
309: enum LOFF_SENSP_bits {
310: LOFF8P = 0x80,
311: LOFF7P = 0x40,
312: LOFF6P = 0x20,
313: LOFF5P = 0x10,
314: LOFF4P = 0x08,
315: LOFF3P = 0x04,
316: LOFF2P = 0x02,
317: LOFF1P = 0x01,
318:
319: LOFF_SENSP_const = 0x00
320: };
321:
322: enum LOFF_SENSN_bits {
323: LOFF8N = 0x80,
324: LOFF7N = 0x40,
325: LOFF6N = 0x20,
326: LOFF5N = 0x10,
327: LOFF4N = 0x08,
328: LOFF3N = 0x04,
329: LOFF2N = 0x02,
330: LOFF1N = 0x01,
331:
332: LOFF_SENSN_const = 0x00
333: };
334:
335: enum LOFF_FLIP_bits {
336: LOFF_FLIP8 = 0x80,
337: LOFF_FLIP7 = 0x40,
338: LOFF_FLIP6 = 0x20,
339: LOFF_FLIP5 = 0x10,
340: LOFF_FLIP4 = 0x08,
341: LOFF_FLIP3 = 0x04,
342: LOFF_FLIP2 = 0x02,
343: LOFF_FLIP1 = 0x01,
344:
345: LOFF_FLIP_const = 0x00
346: };
347:
348: enum LOFF_STATP_bits {
349: IN8P_OFF = 0x80,
350: IN7P_OFF = 0x40,
351: IN6P_OFF = 0x20,
352: IN5P_OFF = 0x10,
353: IN4P_OFF = 0x08,
354: IN3P_OFF = 0x04,
355: IN2P_OFF = 0x02,
356: IN1P_OFF = 0x01,
357:
358: LOFF_STATP_const = 0x00
359: };
360:
361: enum LOFF_STATN_bits {
362: IN8N_OFF = 0x80,
363: IN7N_OFF = 0x40,
364: IN6N_OFF = 0x20,
365: IN5N_OFF = 0x10,
366: IN4N_OFF = 0x08,
367: IN3N_OFF = 0x04,
368: IN2N_OFF = 0x02,
369: IN1N_OFF = 0x01,
370:
371: LOFF_STATN_const = 0x00
372: };
373:
374: enum GPIO_bits {
375: GPIOD4 = 0x80,
376: GPIOD3 = 0x40,
377: GPIOD2 = 0x20,
378: GPIOD1 = 0x10,
379: GPIOC4 = 0x08,
380: GPIOC3 = 0x04,
381: GPIOC2 = 0x02,
382: GPIOC1 = 0x01,
383:
384: GPIO_const = 0x00
385: };
386:
387: enum PACE_bits {
388: PACEE1 = 0x10,
389: PACEE0 = 0x08,
390: PACEO1 = 0x04,
391: PACEO0 = 0x02,

```

```

392: PD_PACE = 0x01,
393:
394: PACE_const = 0x00,
395:
396: PACEE_CHAN2 = 0x00,
397: PACEE_CHAN4 = PACEE0,
398: PACEE_CHAN6 = PACEE1,
399: PACEE_CHAN8 = (PACEE1 | PACEE0),
400:
401: PACEO_CHAN1 = 0x00,
402: PACEO_CHAN3 = PACEE0,
403: PACEO_CHAN5 = PACEE1,
404: PACEO_CHAN7 = (PACEE1 | PACEE0)
405: };
406:
407: enum RESP_bits {
408: RESP_DEMOD_EN1 = 0x80,
409: RESP_MOD_EN1 = 0x40,
410: RESP_PH2 = 0x10,
411: RESP_PH1 = 0x08,
412: RESP_PH0 = 0x04,
413: RESP_CTRL1 = 0x02,
414: RESP_CTRL0 = 0x01,
415:
416: RESP_const = 0x20,
417:
418: RESP_PH_22_5 = 0x00,
419: RESP_PH_45 = RESP_PH0,
420: RESP_PH_67_5 = RESP_PH1,
421: RESP_PH_90 = (RESP_PH1 | RESP_PH0),
422: RESP_PH_112_5 = RESP_PH2,
423: RESP_PH_135 = (RESP_PH2 | RESP_PH0),
424: RESP_PH_157_5 = (RESP_PH2 | RESP_PH1),
425:
426: RESP_NONE = 0x00,
427: RESP_EXT = RESP_CTRL0,
428: RESP_INT_SIG_INT = RESP_CTRL1,
429: RESP_INT_SIG_EXT = (RESP_CTRL1 |
RESP_CTRL0)
430: };
431:
432: enum CONFIG4_bits {
433: SINGLE_SHOT = 0x08,
434: PD_LOFF_COMP = 0x02,
435: CONFIG4_const = 0x00,
436: };
437:
438: enum WCT1_bits {
439: aVF_CH6 = 0x80,
440: aVL_CH5 = 0x40,
441: aVR_CH7 = 0x20,
442: aVR_CH4 = 0x10,
443: PD_WCTA = 0x08,
444: WCTA2 = 0x04,
445: WCTA1 = 0x02,
446: WCTA0 = 0x01,
447:
448: WCT1_const = 0x00,
449:
450: WCTA_CH1P = 0x00,
451: WCTA_CH1N = WCTA0,
452: WCTA_CH2P = WCTA1,
453: WCTA_CH2N = (WCTA1 | WCTA0),
454: WCTA_CH3P = WCTA2,
455: WCTA_CH3N = (WCTA2 | WCTA0),
456: WCTA_CH4P = (WCTA2 | WCTA1),
457: WCTA_CH4N = (WCTA2 | WCTA1 | WCTA0)
458: };
459:
460: enum WCT2_bits {
461: PD_WCTC = 0x80,
462: PD_WCTB = 0x40,
463: WCTB2 = 0x20,
464: WCTB1 = 0x10,
465: WCTB0 = 0x08,
466: WCTC2 = 0x04,
467: WCTC1 = 0x02,
468: WCTC0 = 0x01,
469:
470: WCT2_const = 0x00,
471:
472: WCTB_CH1P = 0x00,
473: WCTB_CH1N = WCTB0,
474: WCTB_CH2P = WCTB1,
475: WCTB_CH2N = (WCTB1 | WCTB0),
476: WCTB_CH3P = WCTB2,
477: WCTB_CH3N = (WCTB2 | WCTB0),
478: WCTB_CH4P = (WCTB2 | WCTB1),
479: WCTB_CH4N = (WCTB2 | WCTB1 | WCTB0),
480:
481: WCTC_CH1P = 0x00,
482: WCTC_CH1N = WCTC0,
483: WCTC_CH2P = WCTC1,
484: WCTC_CH2N = (WCTC1 | WCTC0),
485: WCTC_CH3P = WCTC2,
486: WCTC_CH3N = (WCTC2 | WCTC0),
487: WCTC_CH4P = (WCTC2 | WCTC1),
488: WCTC_CH4N = (WCTC2 | WCTC1 | WCTC0)
489: };
490:
491:
492:
493:
494:
495: #ifdef __cplusplus
496: }
497: #endif
498: #endif

```

ANEXO 9.

CÓDIGO PARA PROCESAMIENTO Y GRÁFICA DE LAS SEÑALES CORRESPONDIENTES A LAS PRUEBAS DE FUNCIONAMIENTO EN MATALB®

```
%% PROCESAMIENTO Y GRÁFICA DE LAS SEÑALES DE LAS SEÑALES ADQUIRIDAS CON EL
% DISPOSITIVO DISEÑADO

clear all, close all, clc
x=xlsread('Resultados.xlsx');
X=x;
figure, plot((X))
% axis([1000 3000 -0.1 0.1])
title('Señal Original')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F1')

%% DESCOMPOSICIÓN DE LA SEÑAL MÉTODO BSS - CCA:

% Descomposición de la señal por el método BSS-CCA a través del algoritmo
% que se encuentra en el Anexo 2.

[Wc,rc] = bsscca(X',1) % El retraso del método se
% sugiere en 1. La matriz Wc es
% la matriz de desmezcla y el
% vector rc es el vector de
% correlaciones.

Yc=Wc*X';

% Se realiza una búsqueda dentro del vector de correlaciones y de acuerdo
% con lo relacionado al método BSS-CCA se eliminan las componentes que
% tengan una correlación menor a 0.75.

for i=1:length(rc)
    if rc(i)<= 0.75
        Yc(i,:)=0;
    end
end

% Proceso de reconstrucción de la señal después de eliminar componentes

Ac=inv(Wc);
Xc=Ac*Yc; % Al realizar esta operación
% se obtiene una señal Xc que
% es una estimación de la señal
% original con los artefactos
% eliminados.

Xc=Xc';
figure, plot(Xc)
% axis([800 1100 -250 250])
title('Señal Reconstruida Salida de Matlab')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F2')

%% FILTRADO DE LA SEÑAL RECONSTRUIDA

Fs=160; % Frecuencia de muestreo de la
% base de datos de Physionet.

% El filtro Butterworth Pasa Banda con frecuencias de corte de 0.5 - 35 Hz
% para hacer una ventana de las frecuencias cerebrales útiles.
Wn=[0.5 35]/(Fs/2);
[b,a]=butter(8,Wn,'bandpass');
sig_pb=filter(b,a,Xc); % Señal procesada sin
```

```
% artefactos utilizando el
% método BSS-CCA.

figure, plot(sig_pb)
% axis([800 1100 -250 250])
title('Señal Final Salida de Matlab')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F3')

%% RESULTADOS ARROJADOS POR EL MICROCONTROLADOR
y=xlsread('Resultados.xlsx', 'Hoja2');
z=xlsread('Resultados.xlsx', 'Hoja3');

% Graficar la señal reconstruida
figure, plot(y)
% axis([800 1100 -250 250])
title('Señal Reconstruida Salida del Microcontrolador')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F4')

% Graficar la señal filtrada
figure, plot(z)
% axis([800 1100 -250 250])
title('Señal Filtrada Salida del Microcontrolador')
xlabel('No. Muestras');
ylabel('Amplitud (V)');
print(gcf, '-dpng', 'F5')

%% FIN DEL CÓDIGO
```

ANEXO 10.

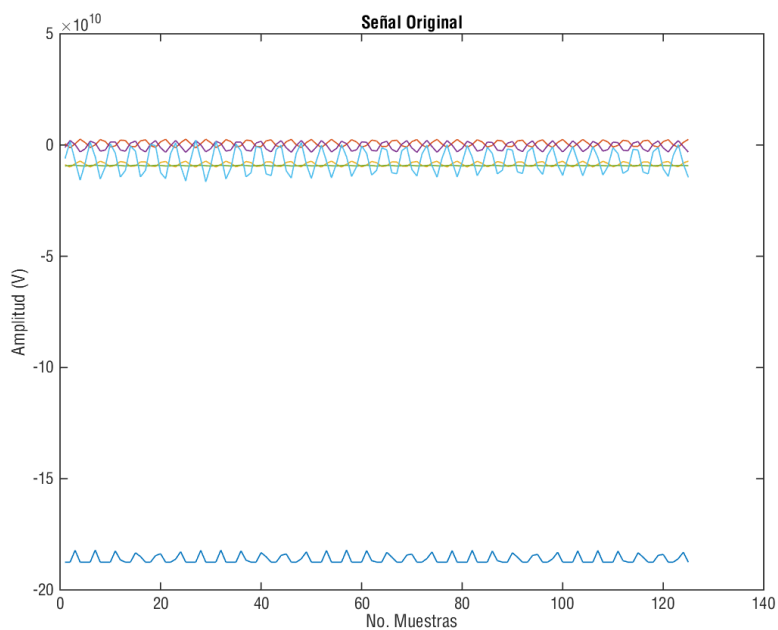
RESULTADOS DE LAS PRUEBAS DE FUNCIONAMIENTO.

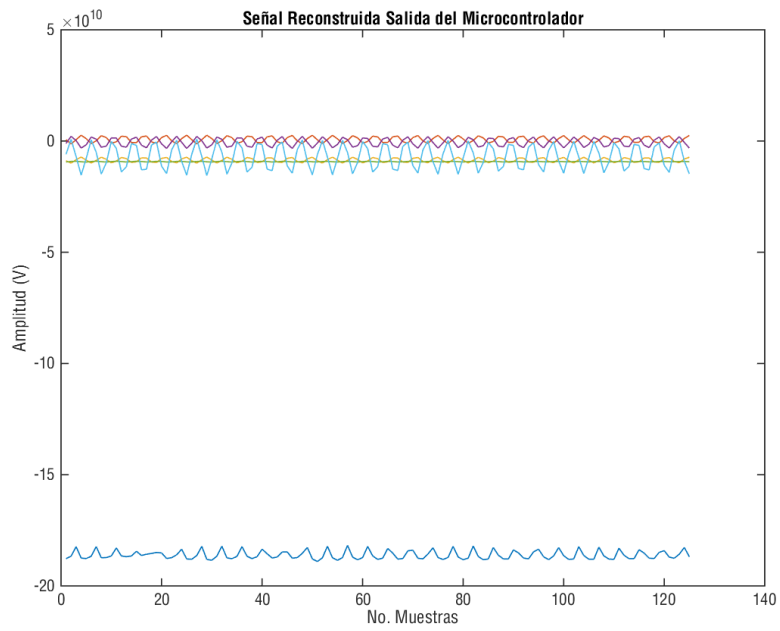
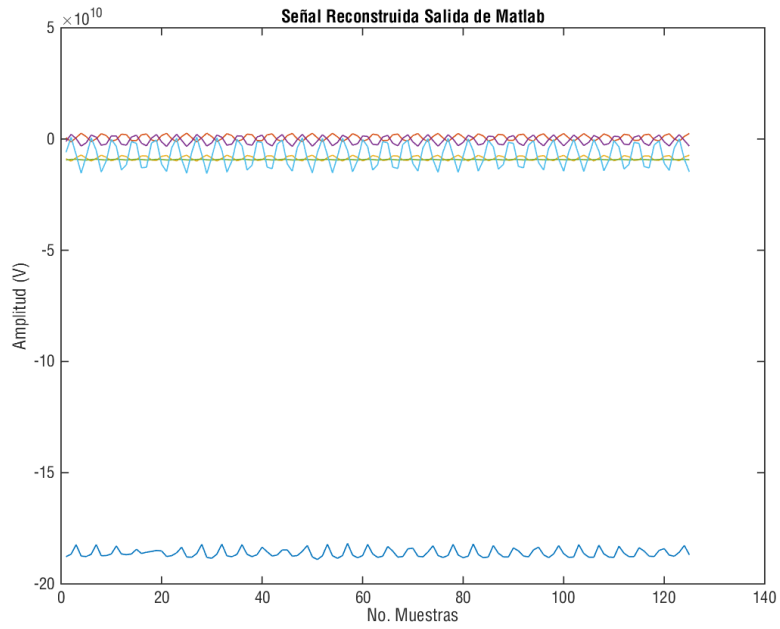
A continuación se presenta la tabla de resultados de los coeficientes de correlación para cada uno de los sujetos en los estados, de movimiento y de reposo.

	1 MOV	1 REP	2 MOV	2 REP
Lugar No. 6	99,99962	100,00000	99,99984	100,00000
Lugar No. 5	99,82913	99,99682	99,80686	99,99855
Lugar No. 4	99,79571	99,99411	99,77074	99,99279
Lugar No. 3	79,86057	94,87686	86,83013	92,35993
Lugar No. 2	54,63767	92,68739	31,88443	85,70274
Lugar No. 1	44,49151	16,31477	24,12395	9,53598

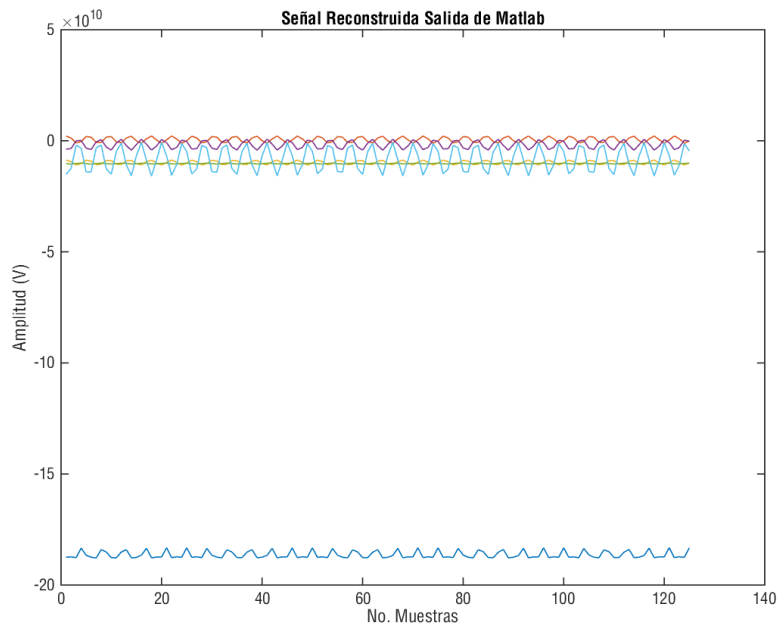
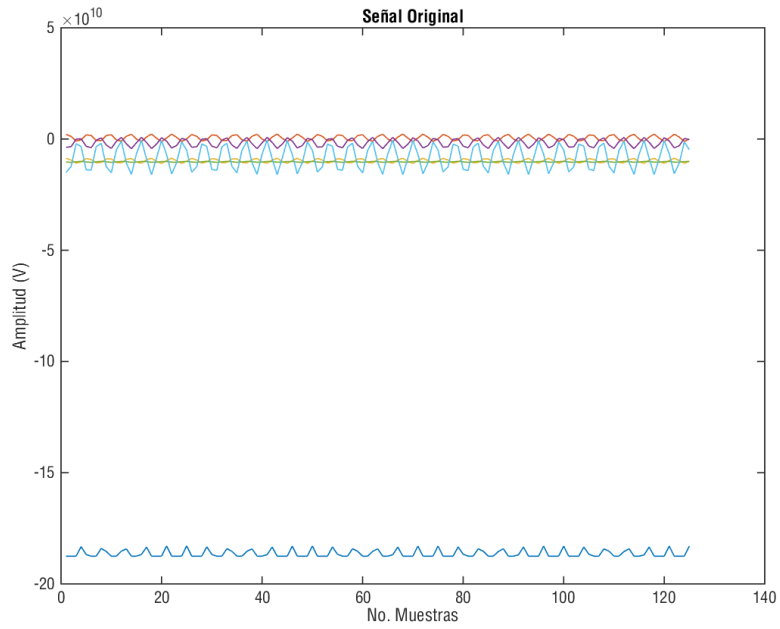
3 MOV	3 REP	4 MOV	4 REP	5 MOV	5 REP
99,99988	100,00000	99,99999	100,00000	99,99996	100,00000
99,74587	99,99760	99,86755	99,99549	97,79241	99,99137
99,64521	99,98547	99,78431	99,99416	97,35959	99,96729
81,69204	94,05015	90,36122	93,28403	87,01675	93,74370
27,50251	90,53026	21,93403	52,17251	56,64092	91,88732
10,34442	2,66212	13,48778	16,18174	28,52576	15,78539

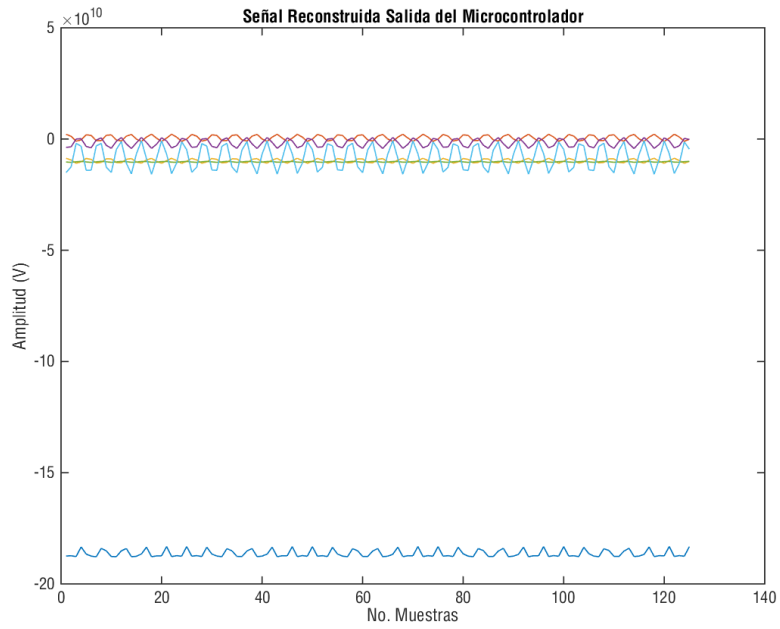
Sujeto 1. Movimiento



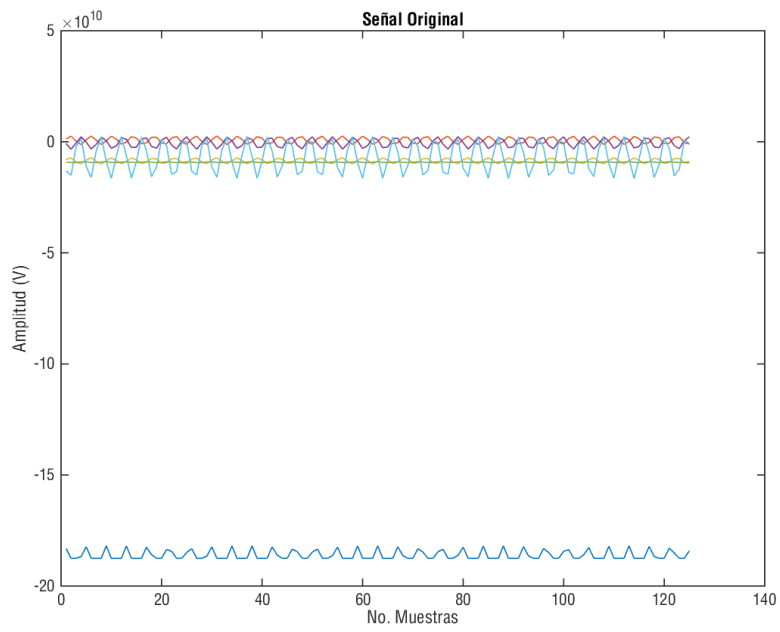
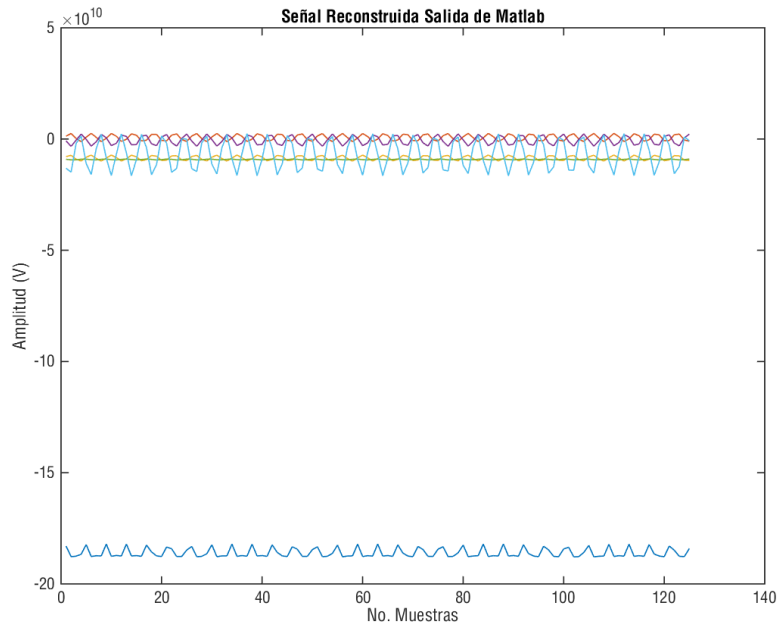


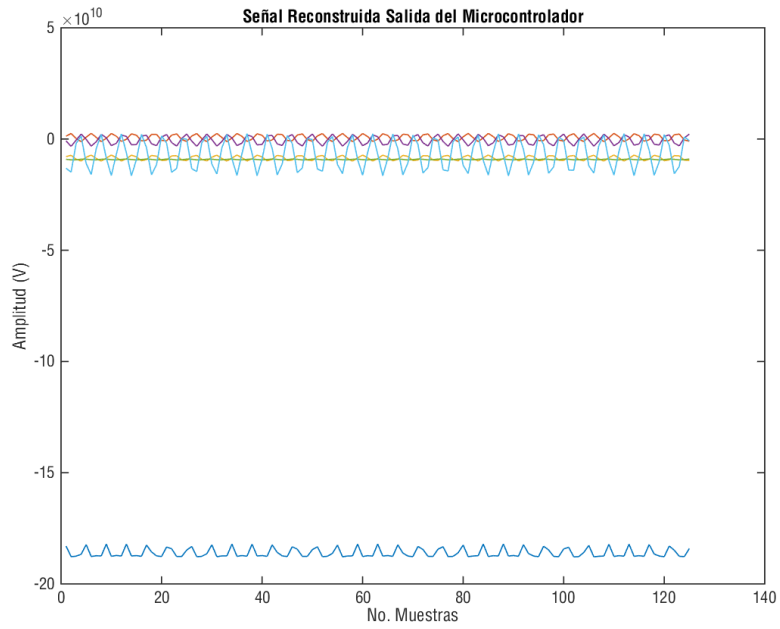
Sujeto 1. Reposo



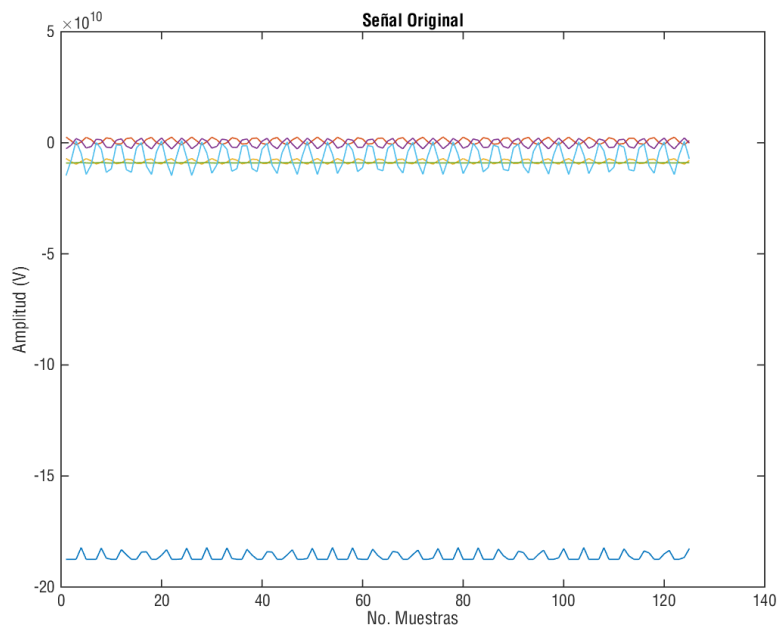


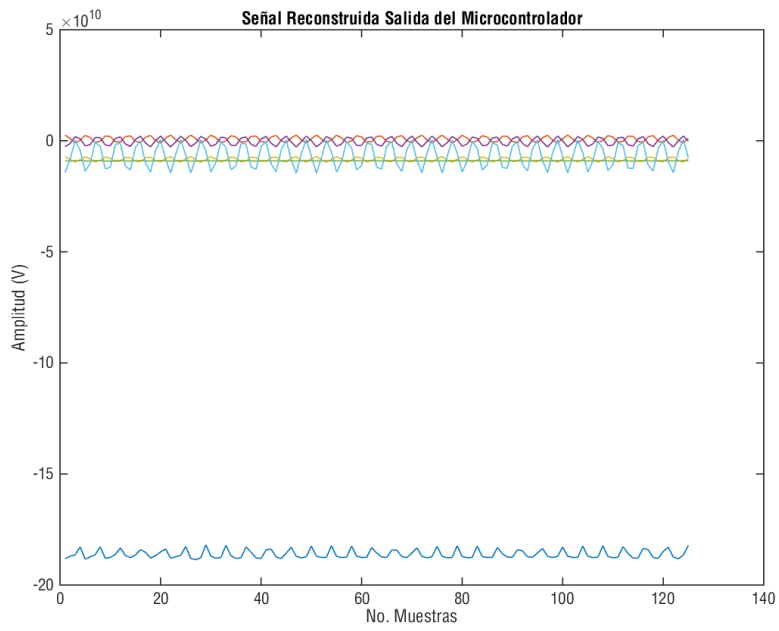
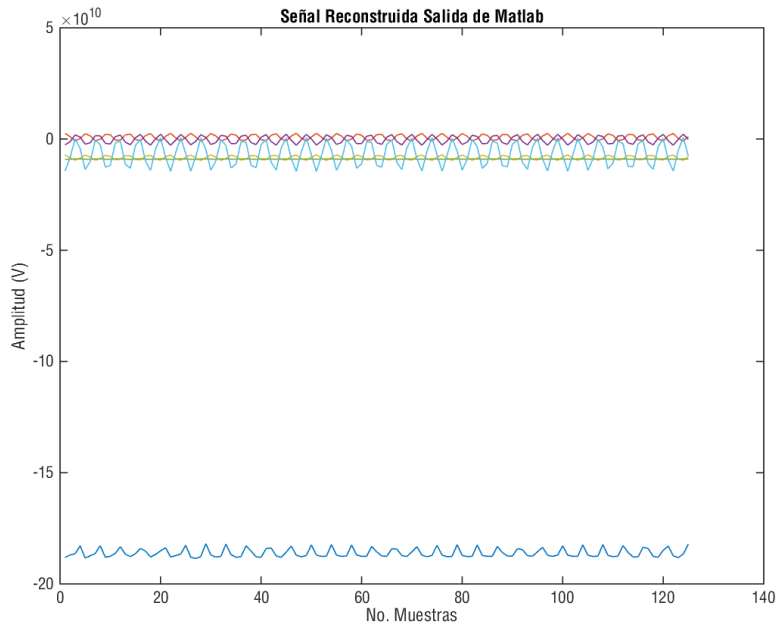
Sujeto 2. Movimiento



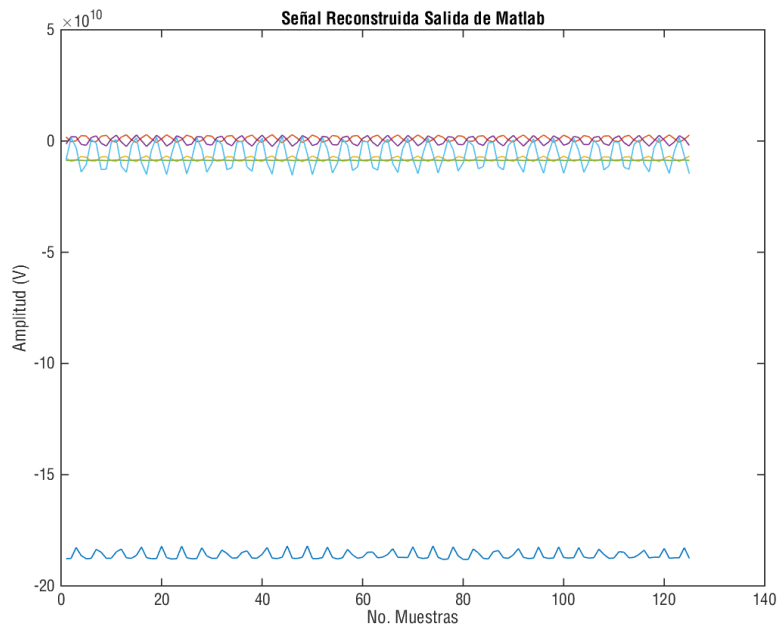
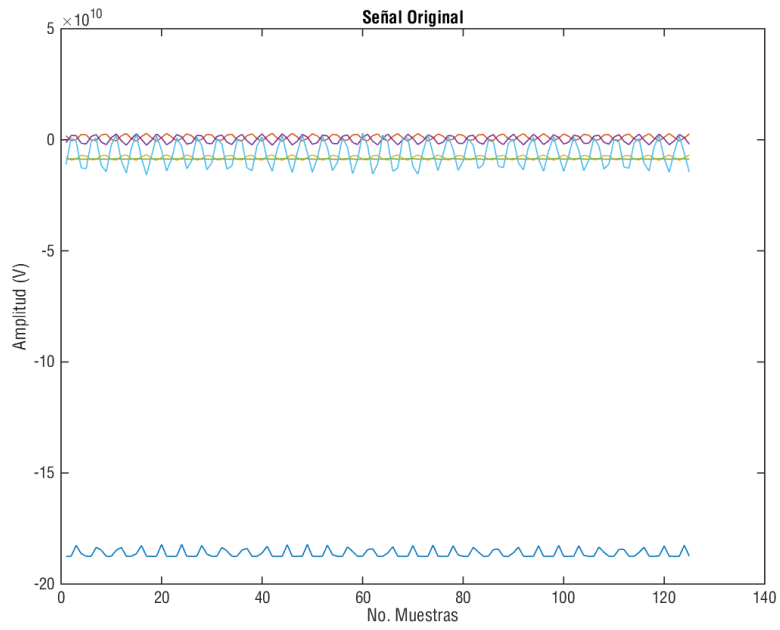


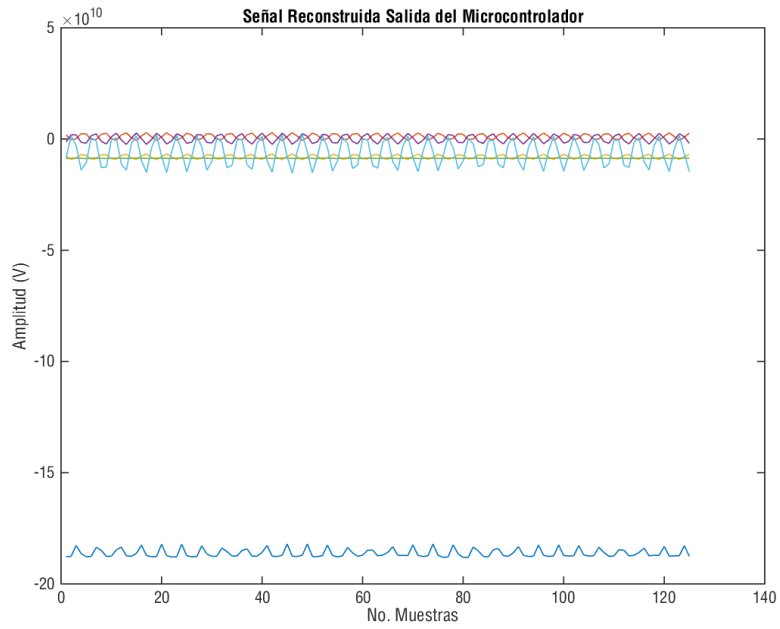
Sujeto 2. Reposo



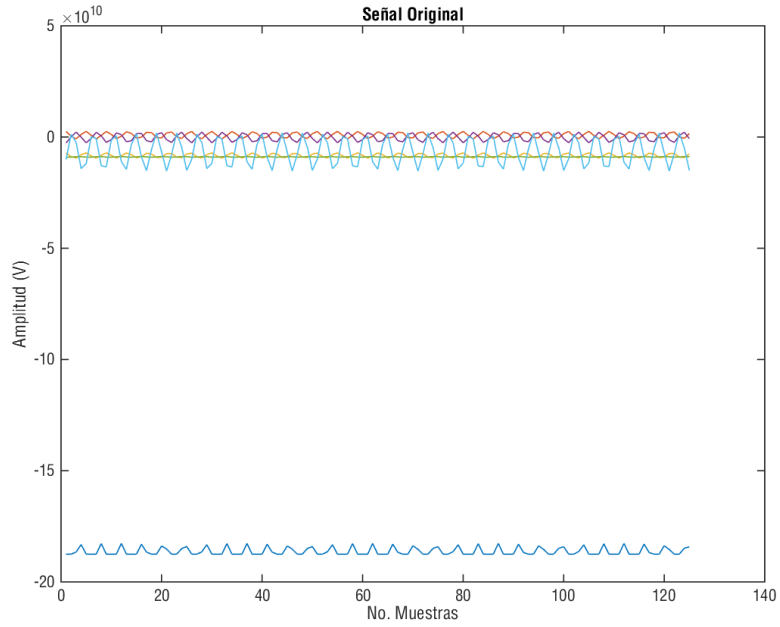


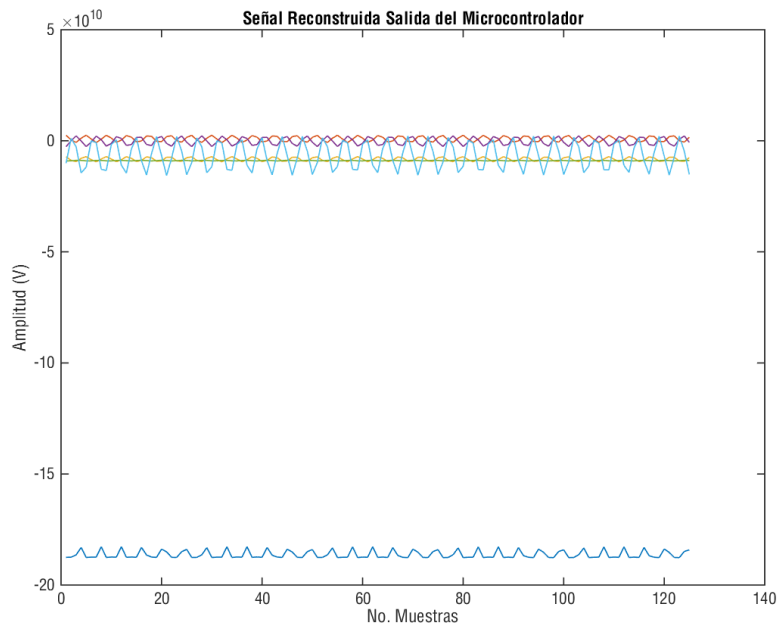
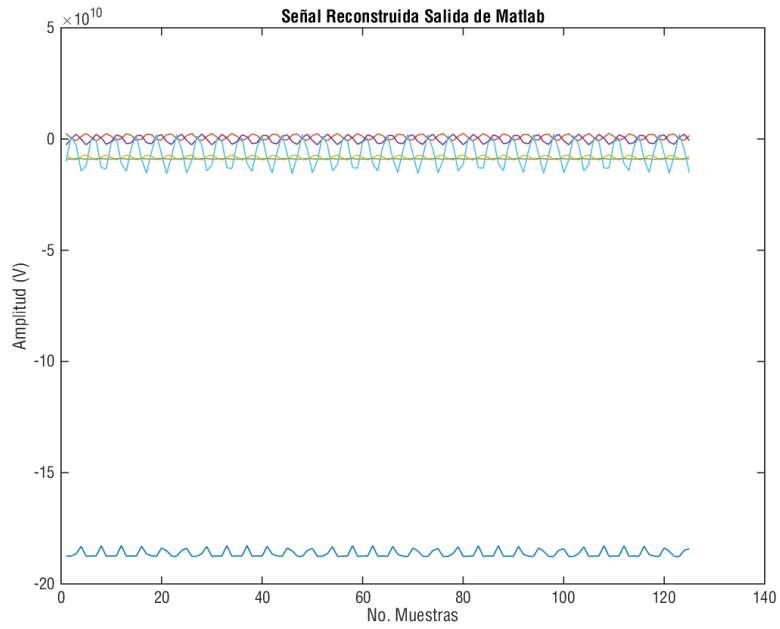
Sujeto 3. Movimiento



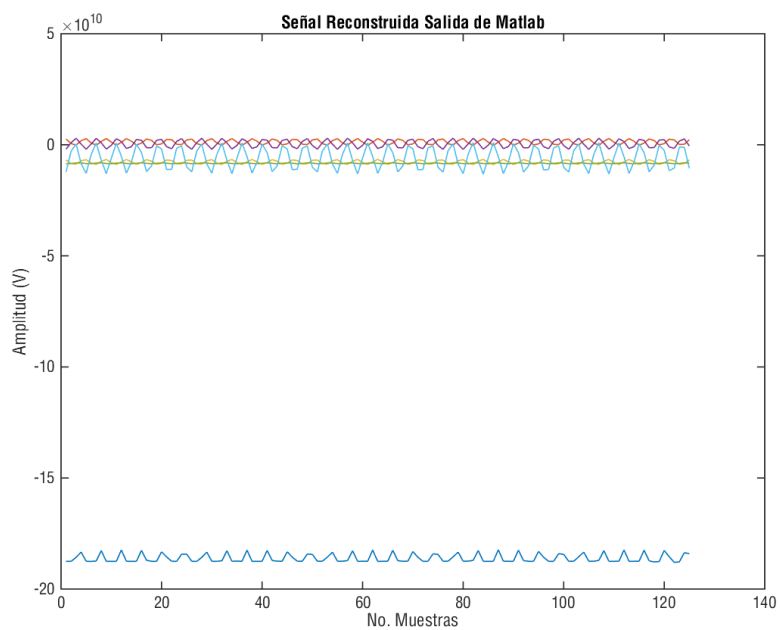
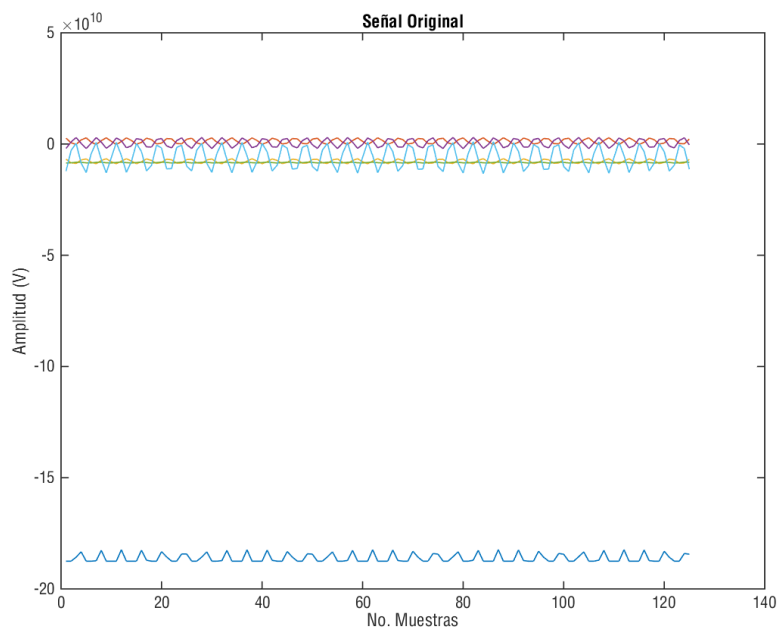


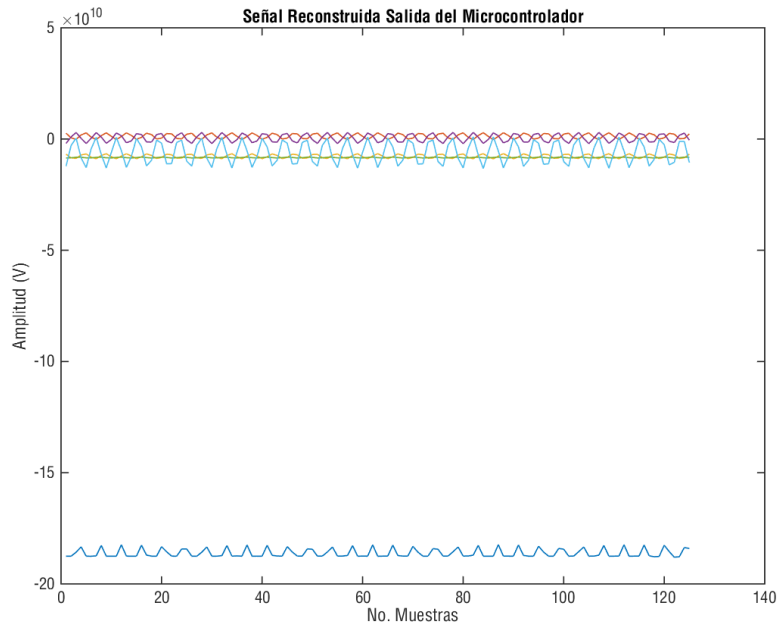
Sujeto 3. Reposo



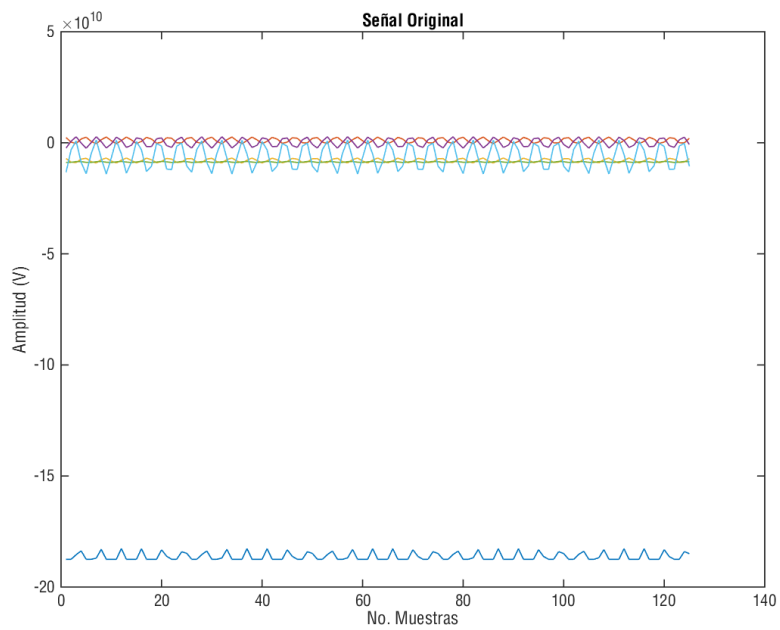


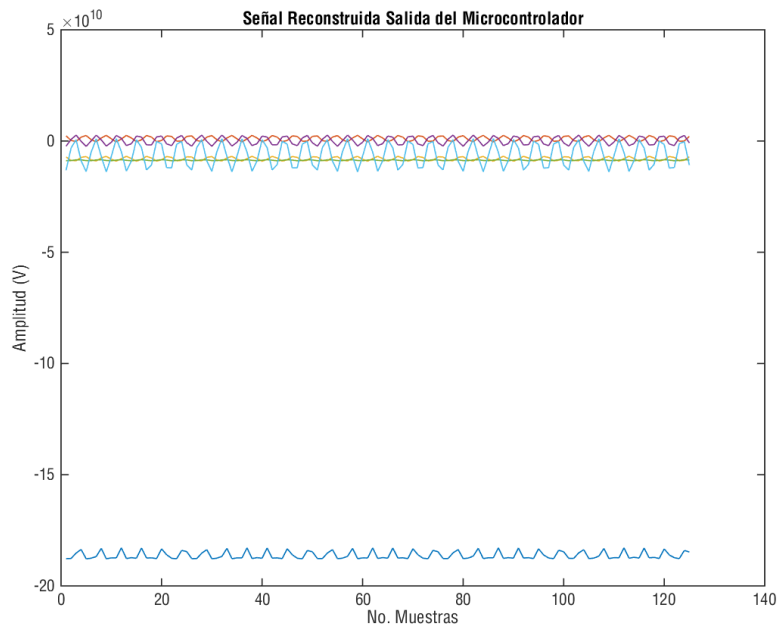
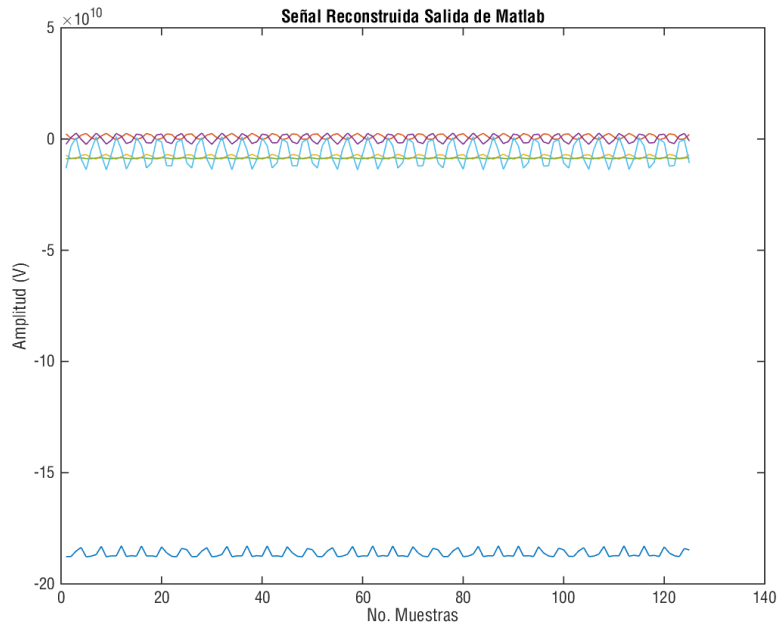
Sujeto 4. Movimiento



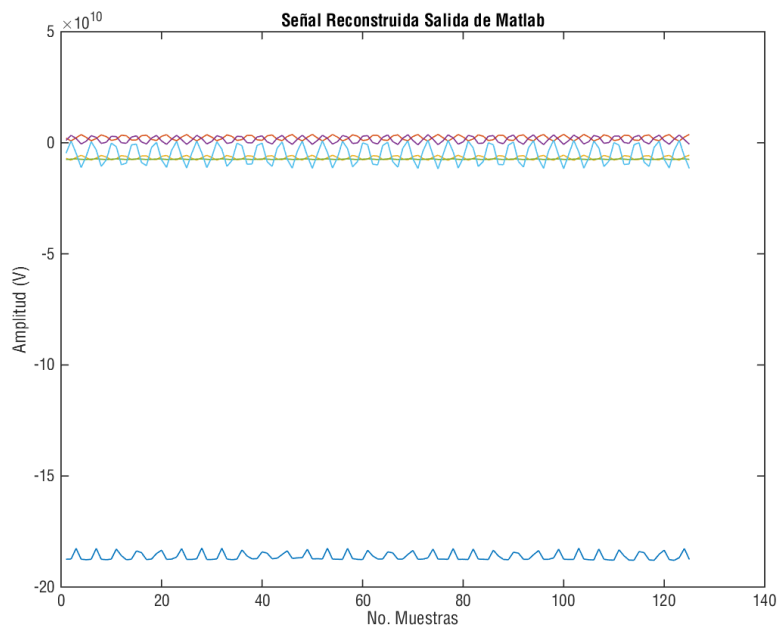
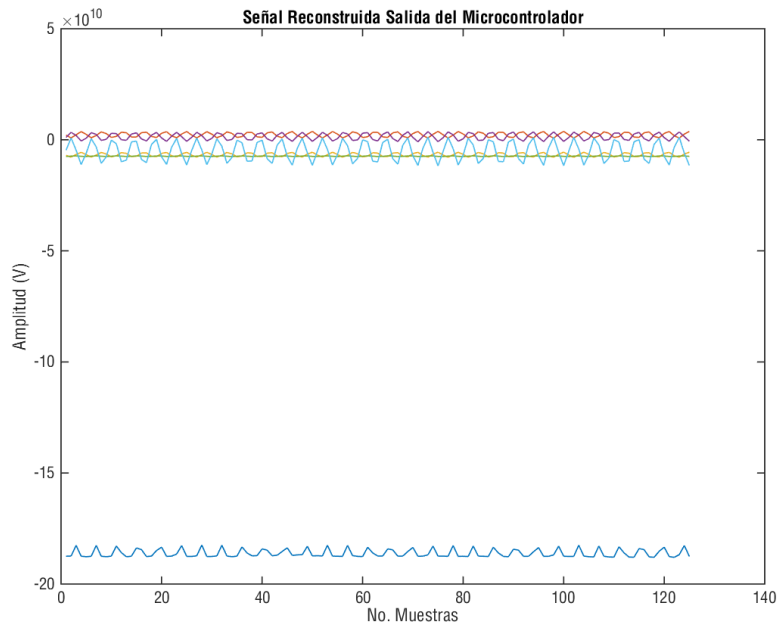


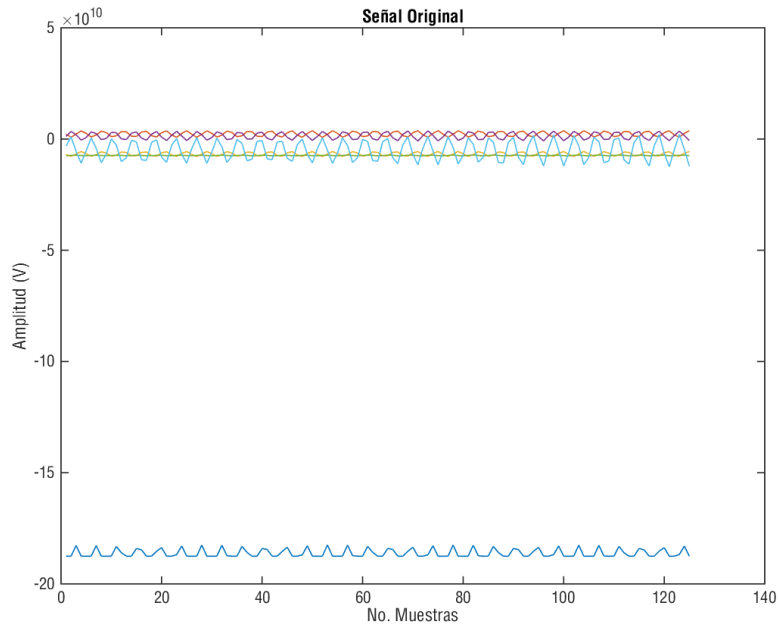
Sujeto 4. Reposo



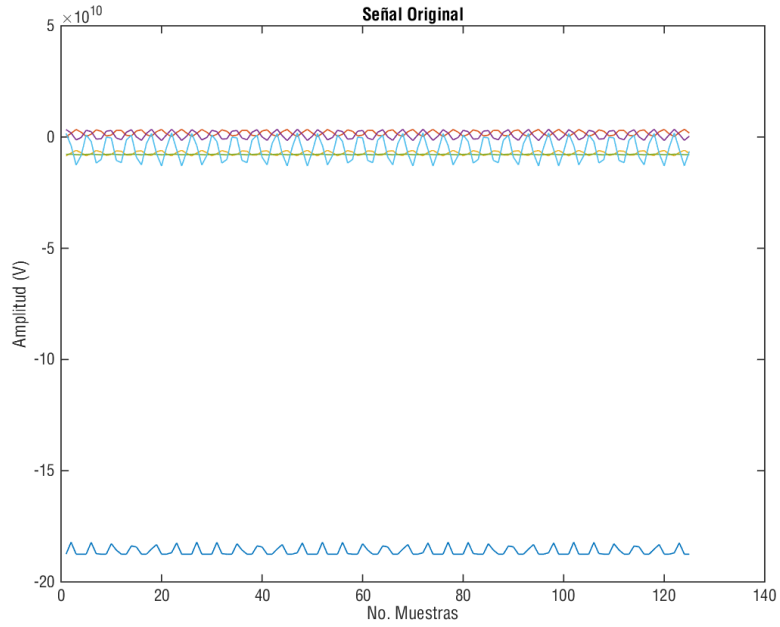


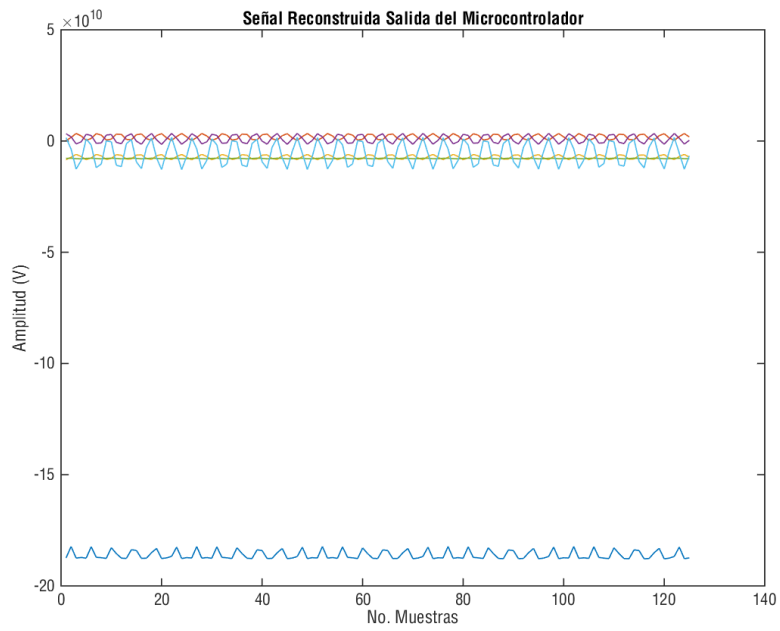
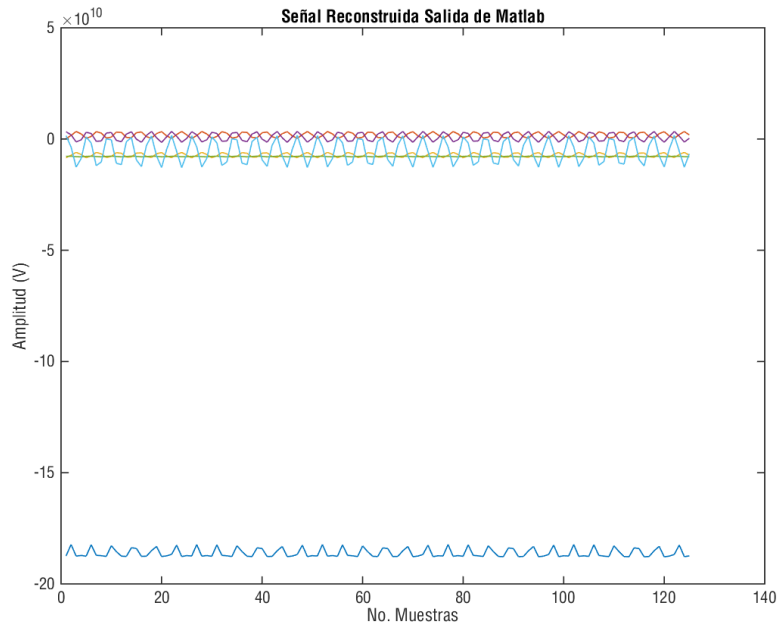
Sujeto 5. Movimiento





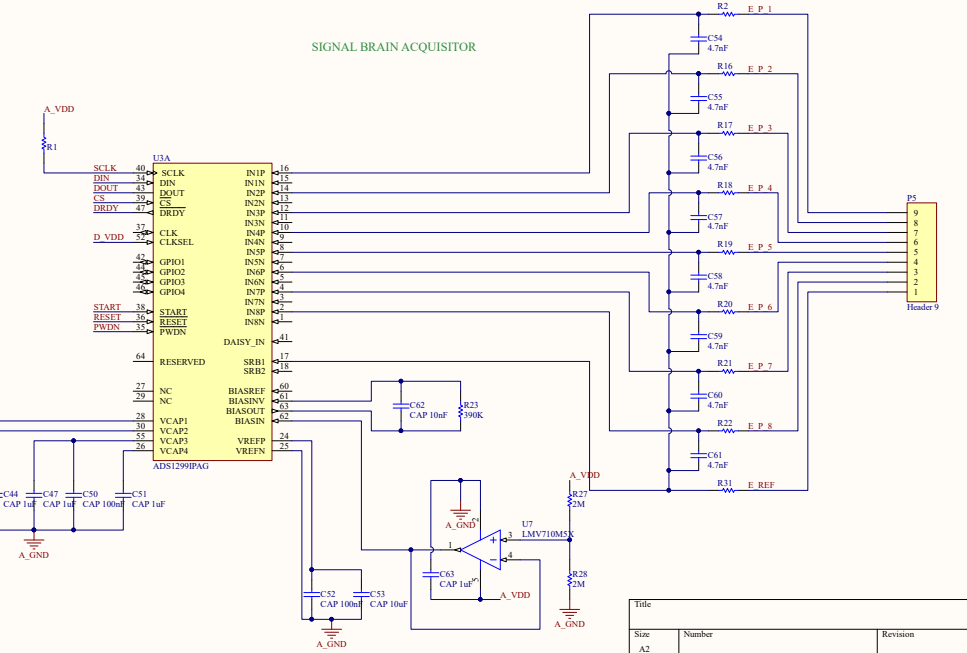
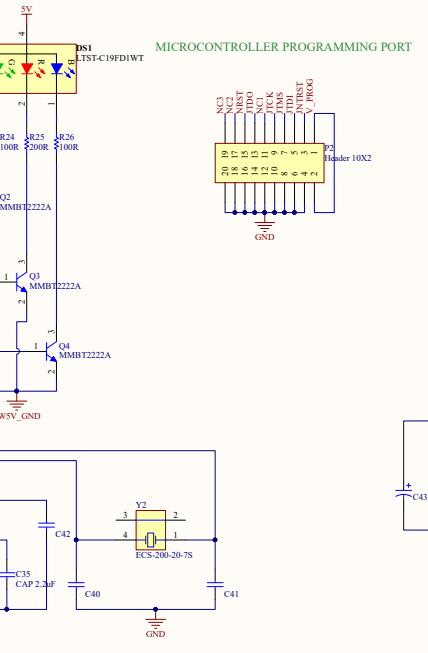
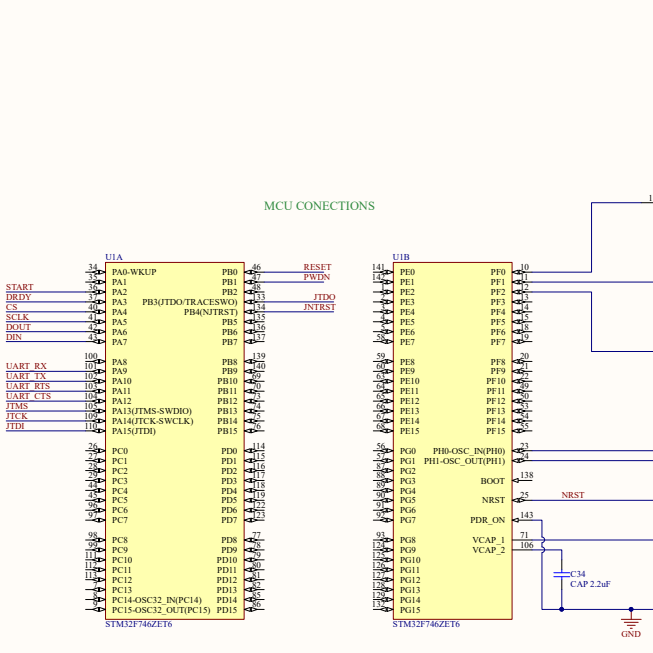
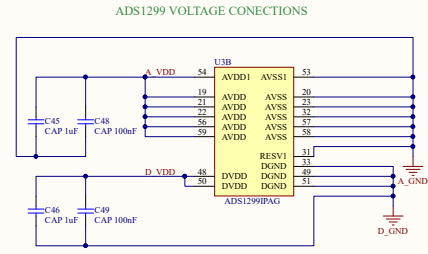
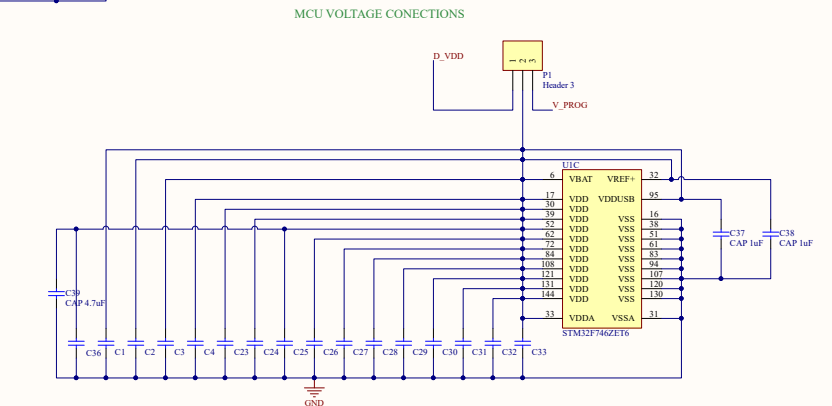
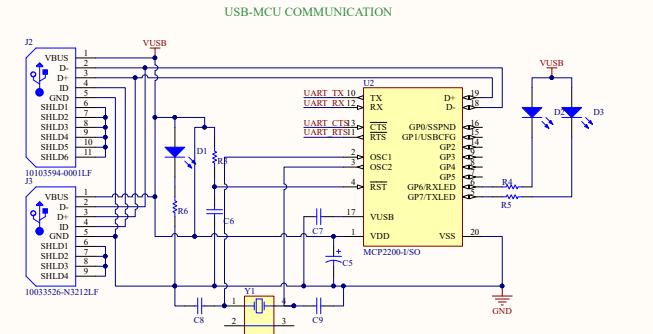
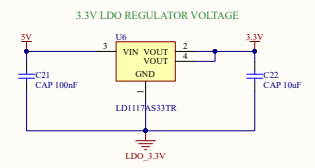
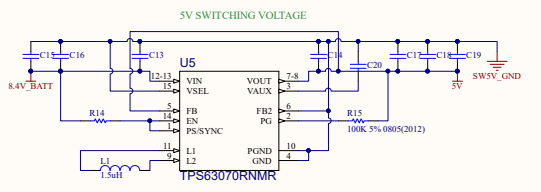
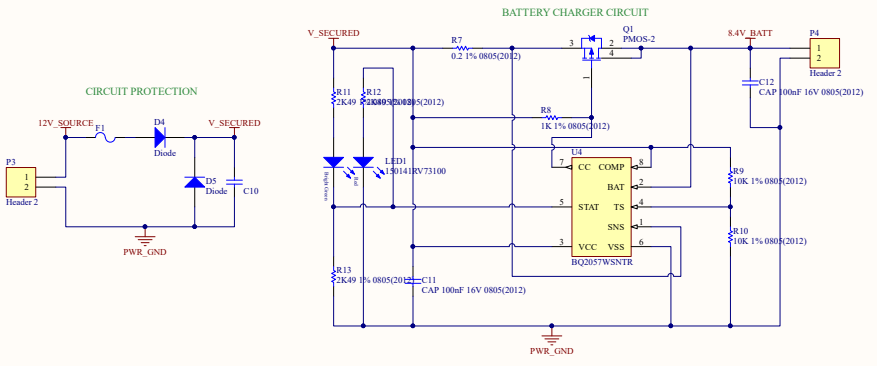
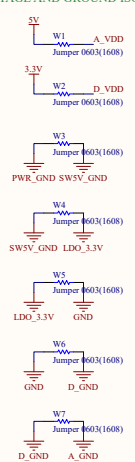
Sujeto 5. Reposo





ANEXO 11. ESQUEMÁTICO GENERAL DEL DISPOSITIVO DISEÑADO.

VOLTAGE AND GROUND ISOLATIONS



Title	Number	Revision
A2		
Date:	26/06/2017	Sheet of
File:	C:\Users\EEG3\SdbDoc	Drawn By:

ANEXO 12.

LISTA DE COMPONENTES DEL DISEÑO DEL DISPOSITIVO.

REFERENCIA	DESCRIPCIÓN	EMPAQUETADO	TIENDA	IDENTIFICADOR EN LA PCB	CANTIDAD	PRECIO UNITARIO (USD)	PRECIO TOTAL (USD)	PRECIO TOTAL (PESOS)
INSUMOS INTERNACIONALES								
595-ADS1299IPAG	Adquisidor	TQFP-64	MOUSER	U2	2	\$ 56,43	\$ 112,86	\$ 339.709
80-C0805C104K4R	0,1 μ F	805	MOUSER	C1, C2, C3, C4, C6, C11, C12, C20, C21, C23-C33, C42, C48, C49, C50, C52	25	\$ 0,05	\$ 1,20	\$ 3.612
71-CRCW0805J-10K-E3	10 K Ω	805	MOUSER	R3, R9, R10, R14	4	\$ 0,08	\$ 0,32	\$ 963
667-ERA-6AED472V	4,7 K Ω	805	MOUSER	R2	1	\$ 0,25	\$ 0,25	\$ 753
579-MCP2200-I/SO	USB-SERIAL	SOIC-20	MOUSER	U3	1	\$ 2,12	\$ 2,12	\$ 6.381
581-TAJD107K016SNJ	100 μ f	7343-31	MOUSER	C5, C43	2	\$ 1,48	\$ 2,96	\$ 8.910
859-LTST-C171KRKT	Red LED	805	MOUSER	D2	1	\$ 0,32	\$ 0,32	\$ 963
859-LTST-C170TBKT	Blue LED	805	MOUSER	D3	1	\$ 0,39	\$ 0,39	\$ 1.174
859-LTST-C171GKT	Green LED	805	MOUSER	D1	1	\$ 0,30	\$ 0,30	\$ 903
71-CRCW0805-50	50 Ω	805	MOUSER	R5	1	\$ 0,14	\$ 0,14	\$ 421
755-KTR10EZPJ151	150 Ω	805	MOUSER	R4, R6	2	\$ 0,12	\$ 0,24	\$ 722
80-C0805C474K5RAUTO	470 nF	805	MOUSER	C7	1	\$ 0,33	\$ 0,33	\$ 993
520-ECX1200-18-X	12 MHz CRYSTAL	7SX	MOUSER	Y1, Y2	2	\$ 1,68	\$ 3,36	\$ 10.114
80-C0805C220F5G	22 pF	805	MOUSER	C8, C9, C40, C41	4	\$ 0,77	\$ 3,08	\$ 9.271

576-1812L110/33MR	FUSE	1812	MOUSER	F1	1	\$ 0,57	\$ 0,57	\$ 1.716
621-B360A-F	SCHOTTKY DIODE	SMA	MOUSER	D4	1	\$ 0,44	\$ 0,44	\$ 1.324
621-SMAJ28CA-13-F	TVS DIODE	SMA	MOUSER	D5	1	\$ 0,41	\$ 0,41	\$ 1.234
581-TAJR104K035RNJ	0.1 µF TANTALUM	805	MOUSER	C10	1	\$ 0,74	\$ 0,74	\$ 2.227
71-WSL0805R2000FEA18	0,2 Ω	805	MOUSER	R7	1	\$ 0,76	\$ 0,76	\$ 2.288
71-TNPW08051K00DEEA	1 KΩ	805	MOUSER	R8	1	\$ 0,88	\$ 0,88	\$ 2.649
522-ZXMP10A18GTA	P - MOSFET	SOT-223-3	MOUSER	Q1	1	\$ 1,56	\$ 1,56	\$ 4.696
710-150141RV73100	Bi - Color LED RED/GREEN	1411	MOUSER	LED1	1	\$ 0,30	\$ 0,30	\$ 903
71-TNPW08052K49BEEA	2,49 KΩ	805	MOUSER	R11, R12, R13	3	\$ 0,71	\$ 2,13	\$ 6.411
595-BQ2057WSNTR	BATT LINEAR CHARGER IC	SOIC-8	MOUSER	U4	1	\$ 2,00	\$ 2,00	\$ 6.020
595-TPS63070RNMR	SWITCHING VOLTAGE REGULATOR 5V	VQFN-15-HR	MOUSER	U5	1	\$ 2,74	\$ 2,74	\$ 8.247
810-C1608X5R1E106M	10 µF	603	MOUSER	C13, C14	2	\$ 0,88	\$ 1,76	\$ 5.298
994-XFL4020-152MEC	1.5 µH	4,3 x 4,3 mm	MOUSER	L1	1	\$ 1,99	\$ 1,99	\$ 5.990
81-GRM21BC71E106KE1L	10 µF	805	MOUSER	C15, C16, C22	3	\$ 0,64	\$ 1,92	\$ 5.779
963-LMK212BC6226MG-T	22 µF	805	MOUSER	C17, C18, C19	3	\$ 0,74	\$ 2,22	\$ 6.682
71-TNPW0805100KBEEN	100 KΩ	805	MOUSER	R15	1	\$ 0,78	\$ 0,78	\$ 2.348
511-LD1117AS33	3.3V LDO REGULATOR	SOT-223-3	MOUSER	U6	1	\$ 0,64	\$ 0,64	\$ 1.926
71-CRCW08056M80FKEA	6,8 MΩ	805	MOUSER	R1	1	\$ 0,10	\$ 0,10	\$ 301
511-STM32F767ZIT6	STM Microcontroller	LQFP144	MOUSER	U1	1	\$ 15,98	\$ 15,98	\$ 48.100

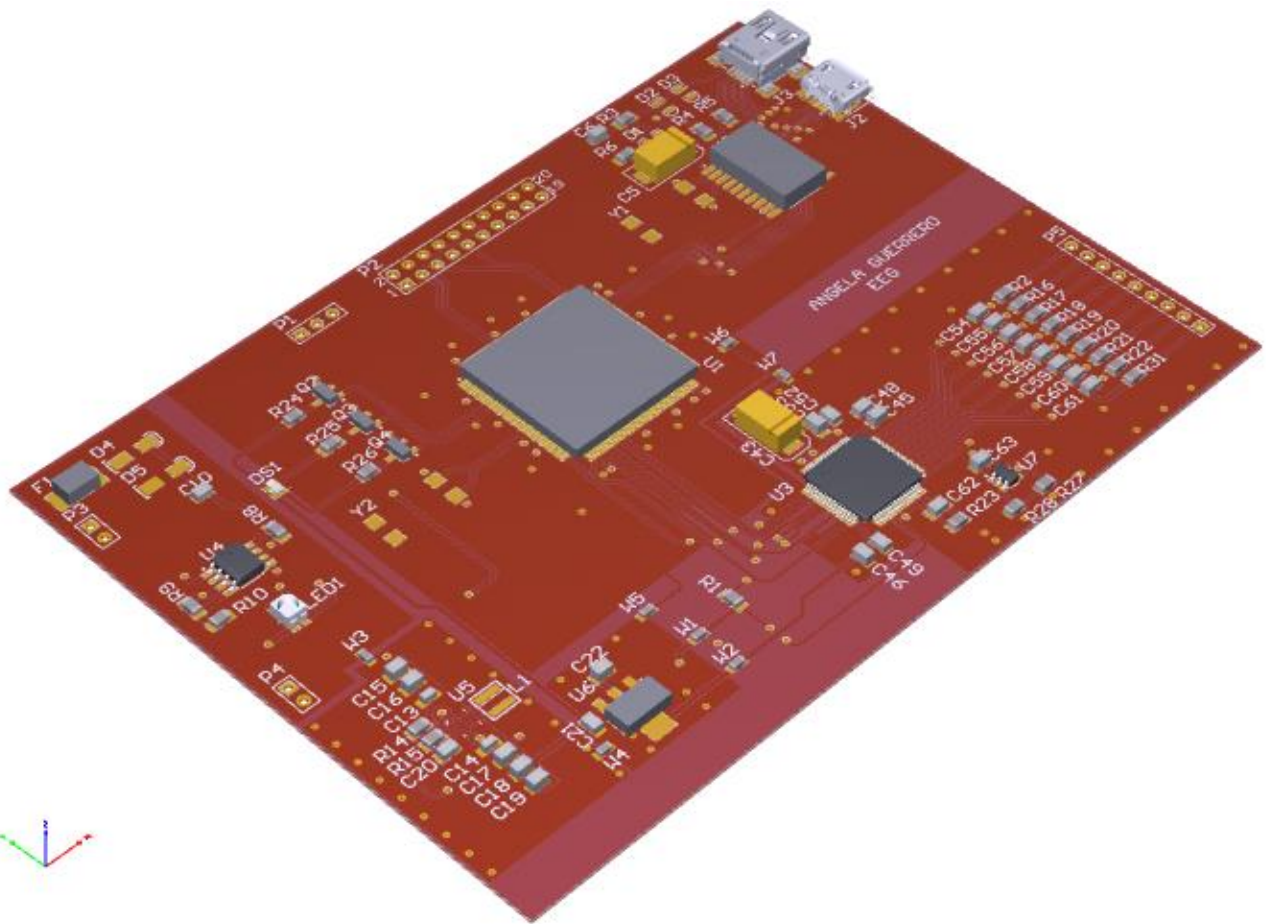
81-GRM21BR71A225KA1K	2,2 µf	805	MOUSER	C34, C35	2	\$ 0,16	\$ 0,32	\$ 963
581-08053C105KAT4A	1 µf	805	MOUSER	C36, C37, C38, C44, C45, C46, C47, C51, C63	8	\$ 0,10	\$ 0,80	\$ 2.408
81-GRM21BR61E475MA2L	4.7 µf	805	MOUSER	C39	1	\$ 0,16	\$ 0,16	\$ 482
667-ERA-6AEB4991V	4,99 KΩ	805	MOUSER	R2, R16-R22, R31	9	\$ 0,63	\$ 5,67	\$ 17.067
80-C0805C472F5GACTU	4.7 nF	805	MOUSER	C54-C61	8	\$ 2,41	\$ 19,28	\$ 58.033
80-C0805C103K5R	10 nF	805	MOUSER	C62	1	\$ 0,10	\$ 0,10	\$ 301
859-LTST-C19FD1WT	LED RGB	805	MOUSER	DS1	1	\$ 0,60	\$ 0,60	\$ 1.806
667-ERJ-6ENF3903V	390 KΩ	805	MOUSER	R23	1	\$ 0,10	\$ 0,10	\$ 301
512-MMBT2222A	NPN TRANSISTOR	SOT-23-3	MOUSER	Q2, Q3, Q4	3	\$ 0,20	\$ 0,60	\$ 1.806
667-ERA-6AEB101V	100 Ω	805	MOUSER	R24, R26	2	\$ 0,63	\$ 1,26	\$ 3.793
71-CRCW0805-200-E3	200 Ω	805	MOUSER	R25	1	\$ 0,10	\$ 0,10	\$ 301
926-LMV710M5X/NOPB	Op amplifier	SOT-23-5	MOUSER	U7	1	\$ 1,12	\$ 1,12	\$ 3.371
71-CRCW0805-2M-E3	2 MΩ	805	MOUSER	R27, R28	2	\$ 0,10	\$ 0,20	\$ 602
INSUMOS NACIONALES								
REGLETA SIMPLE			BOGOTA CENTRO		1	\$ 500	\$ 500	\$ 500
REGLETA DOBLE			BOGOTA CENTRO		1	\$ 2.500	\$ 2.500	\$ 2.500
REGLETA EN L			BOGOTA CENTRO		1	\$ 1.300	\$ 1.300	\$ 1.300
BATERIA 9V			BOGOTA CENTRO		1	\$ 30.000	\$ 30.000	\$ 30.000
TORNILLOS BAQUELITA			BOGOTA CENTRO		1	\$ 2.100	\$ 2.100	\$ 2.100
SILICONA			BOGOTA CENTRO		1	\$ 3.000	\$ 3.000	\$ 3.000
CABLE ROJO			BOGOTA CENTRO		1	\$ 2.000	\$ 2.000	\$ 2.000

CABLE NEGRO			BOGOTA CENTRO		1	\$ 2.000	\$ 2.000	\$ 2.000
CAJA DE RECUBRIMIENTO			BOGOTA CENTRO		1	\$ 15.000	\$ 15.000	\$ 15.000
BAQUELITA Main board			BOGOTA CENTRO		1	\$ 7.900	\$ 7.900	\$ 7.900
TOTAL EN PESOS								\$ 656.561

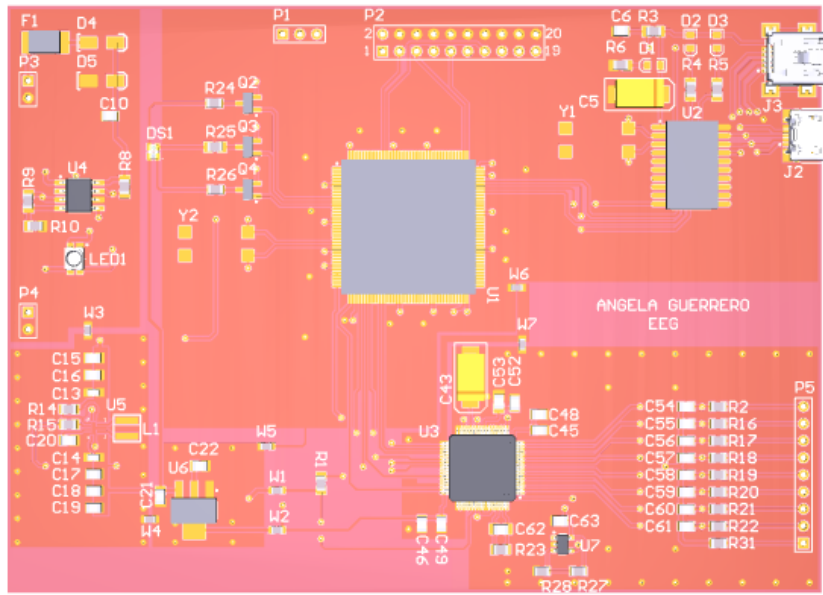
ANEXO 13.

MODELO 3D DEL DISEÑO DEL DISPOSITIVO.

Vista por defecto:



Vista desde arriba:



Vista desde abajo:

