



## SOFTWARE “BIOMEPRO” PARA ACONDICIONAMIENTO DE SEÑALES EMG SUPERFICIALES Y OBTENCIÓN DE PARÁMETROS RELEVANTES<sup>1</sup>

SOFTWARE “BIOMEPRO” PARA ACONDICIONAMENTO DE SINAIS EMG SUPERFICIAIS E OBTENÇÃO DE PARÂMETROS RELEVANTES<sup>2</sup>

“BIOMEPRO” SOFTWARE FOR THE CONDITIONING OF SUPERFICIAL EMG SIGNALS AND OBTENTION OF RELEVANT PARAMETERS<sup>3</sup>

Robin Blanco,<sup>4</sup> Andrés Cifuentes,<sup>5</sup> Mauricio Plaza<sup>6</sup>  
Universidad Militar Nueva Granada

---

### CIENCIA Y PODER AÉREO

ISSN 1909-7050 / E- ISSN 2389-2468 / Volumen 11/ Enero-diciembre de 2016/ Colombia/ Pp. 194-201

Recibido: 21/08/2016

Aprobado evaluador interno: 02/09/2016

Aprobado evaluador externo: 12/09/2016

Doi: <http://dx.doi.org/10.18667/cienciaypoderaereo.531>



**Para citar este artículo:**

Blanco, R., Cifuentes, A., & Plaza, M. (2016). Software BIOMEPRO para acondicionamiento de señales EMG superficiales y obtención de parámetros relevantes. *Ciencia y Poder Aéreo*, 11 (1), 194-201. Doi: <http://dx.doi.org/10.18667/cienciaypoderaereo.531>

<sup>1</sup> Artículo científico original derivado del proyecto IMP-ING-2135 titulado "Diseño de dispositivo exoesquelético mecánico para ayuda de personas con problema de tensión muscular y actividades repetitivas de la Universidad Militar Nueva Granada (UMNG), Bogotá D.C., Colombia (2016).

<sup>2</sup> Artigo científico original derivado do projeto IMP-ING-2135 com título "Desenho de um dispositivo exoesquelético mecânico para ajuda de pessoas com problemas de tensão muscular e atividades repetitivas da Universidade Militar Nueva Granada (UMNG), Bogotá D.C., Colômbia (2016).

<sup>3</sup> Original scientific article derived from the IMP-ING-2135 project titled "Design of mechanical exoskeletal device to help people with muscular tension problems and repetitive activities of the Nueva Granada Military University (UMNG), Bogota D.C., Colombia (2016).

<sup>4</sup> Ingeniero Mecatrónico de la Universidad Nacional de Colombia, Estudiante de la Maestría en Ingeniería Mecatrónica de la Universidad Militar Nueva Granada. Correo electrónico: u3900208@unimilitar.edu.co

<sup>5</sup> Ingeniero Mecatrónico de la Universidad Nacional de Colombia, Magíster en Ingeniería Mecatrónica de la Universidad Militar Nueva Granada. Correo electrónico: andres.cifuentes@unimilitar.edu.co

<sup>6</sup> Ingeniero Eléctrico Universidad de los Andes, Doctor en Ingeniería con énfasis en Realidad Virtual y Bioingeniería. Correo electrónico: mauricio.plaza@unimilitar.edu.co

**Resumen:** en el presente artículo se da a conocer el desarrollo de BioMePro, una herramienta de *software* para el Procesamiento de Muestras de Electromiografía (EMG) superficial, implementado en JAVA, un lenguaje multiplataforma, orientado a objetos y de código interpretado, lenguaje escogido para facilitar la portabilidad del *software* en mención. BioMePro facilita el trabajo de extracción de las características de señales EMG superficiales, útiles para sistemas de inferencia como redes neuronales o máquinas de soporte vectorial, a partir de una interfaz clara y fácil de usar, con un conjunto variado de opciones que permiten seleccionar el mejor procesamiento que se requiera para una aplicación específica. Como sistema de adquisición de los datos se utilizó el equipo BiosignalPlux®, las señales de EMG de prueba del software que fueron capturadas por medio de la implementación de un protocolo de registro de movimientos de pronación en antebrazo derecho en pacientes jóvenes sanos, este protocolo fue elegido por su simplicidad de implementación y fiabilidad en las muestras.

**Palabras clave:** electromiografía; envoltura RMS; Java; procesamiento de señales.

**Resumo:** no presente artigo divulga se o desenvolvimento de BioMePro, uma ferramenta de software para o processamento de a mostrás de eletromiografia (EMG) superficial, implementado em JAVA, uma linguagem multi-plataforma orientada para objetos e de código interpretado, linguagem escolhida para facilitar a portabilidade do software, mencionado. BioMePro facilita o trabalho de extração de características de sinais EMG superficiais, úteis para sistemas de inferência quanto redes neurais ou máquinas de suporte vetorial, a partir duma interface clara e fácil de usar com um conjunto variado de opções que permitem eleger o melhor processamento que é requerido para uma aplicação específica. Como sistema de aquisição dos dados se usou o equipamento BiosignalPlux®, os sinais de EMG de prova do software que foram capturados por meio da implementação dum protocolo de registro de movimentos de pronação no antebraço direito em pacientes jovens e sadios, protocolo elegido pela sua simplicidade de implementação e confiabilidade nas a mostrás.

**Palavras-chave:** eletromiografia; envoltura RMS; Java; processamento de sinais.

**Abstract:** This article presents the development of BioMePro, a software tool for the processing of superficial electromyography (EMG) samples, implemented in JAVA, a multiplatform language, object oriented and interpreted code, language chosen in order to facilitate the portability of the Software in mention. BioMePro facilitates the work of extracting characteristics of superficial EMG signals, useful for inference systems such as neural networks or vector support machines, from a clear and easy to use interface with a varied set of options that allow to select the best processing that is required for a specific application. As a data acquisition system, the BiosignalPlux® equipment was used, the EMG test signals from the software that were captured through the implementation of a protocol for recording right forearm pronation movements in healthy young patients, a protocol chosen by its simplicity of implementation and reliability in the samples.

**Key Words:** Electromyography; Java; RMS Wrapping; Signal Processing.

### Introducción

El desarrollo de herramientas de *software* para el procesamiento de bioseñales ha venido en aumento, debido a las ventajas que ha mostrado para el apoyo del personal médico en el diagnóstico, tratamiento y paliación de cualquier afectación a la salud de las personas (Kazerooni & Strausser, 2011). Dichos programas van desde aquellos que solamente graban y/o visualizan las bioseñales, hasta programas que procesan los datos y obtienen parámetros importantes para un diagnóstico; como, por ejemplo, la frecuencia cardíaca y la frecuencia respiratoria.

Estas herramientas son muy útiles para el desarrollo de investigaciones en el área médica y su posterior aplicación práctica (Romo, Realpe, & Jojoa, 2007) (Figueroa-García, y otros, 2014) (Momen, Krishnan, & Chau, 2007). En etapas de investigación tempranas suele no ser favorable la adquisición de una amplia gama de estas herramientas, por el costo de algunas, que se suma al de los equipos de adquisición de bioseñales (Baturone & Gersnoviez, 2010). La idea de herramientas de *software* portables, de bajo costo, para el procesamiento complejo de bioseñales es una idea llamativa para una gran cantidad de proyectos que requieren la obtención de información de señales electrofisiológicas, para aplicaciones más especializadas.

Una de las bioseñales más procesadas en el ámbito de la investigación biomédica, es la de electromiografía (EMG) (Figueroa-García, y otros, 2014) (Saikia, Mazumdar, Sahai, Paul, & Bhatia, 2016), la cual corresponde a la medición de la actividad eléctrica en el músculo debida a una acción de movimiento, la información obtenida de esta clase de señales ha sido muy útil en desarrollos ingenieriles como *interfaces* para el control de sistemas robóticos tipo exoesqueletos (Kazerooni & Strausser, 2011). Esta señal se puede medir de forma superficial sin generar mayor incomodidad al paciente en estudio, ni comprometer la riqueza de la información relacionada con la medición. A la hora de trabajar con señales de EMG superficiales es necesario considerar el hecho de que en esta no solo se encuentra la información de la actividad eléctrica de una única célula muscular o unidad motora, sino que se registra la suma de muchas actividades de las células cercanas a la colocación del electrodo, por lo tanto, al visualizar la señal, esta suele presentar un patrón similar al ruido eléctrico, lo que obliga a definir una estrategia para obtener parámetros relevantes que indiquen información clara del movimiento en estudio (Saikia, Mazumdar, Sahai, Paul, & Bhatia, 2016).

### Metodología

El desarrollo de la herramienta de *software* BioMePro (Universidad Militar Nueva Granada, 2016) partió de unos re-

querimientos iniciales obtenidos de la interacción directa con investigadores del área para conocer las necesidades relacionadas con adquisición y procesamiento de bioseñales en sus respectivos proyectos; estos requerimientos suelen conocerse como requerimientos de cliente. En este caso para el *software* diseñado las necesidades encontradas fueron:

- El programa debe recibir archivos de texto plano.
- El programa debe permitir configurar todas las opciones de registro y debe visualizar todo procesamiento realizado.
- En lo posible el programa debe ser multiplataforma.
- Debe tener una interfaz clara y de uso intuitivo.
- El procesamiento debe ser rápido, preciso y exacto.
- Los datos de salida deben ser fácilmente exportables a cualquier programa que realice inferencias.

Estos requerimientos fueron objetivos a cumplir en el diseño, para lo cual se realizó un seguimiento en las pruebas finales del *software* con el fin de corroborar si fueron satisfechos y evaluar la calidad de ello.

En el proceso de diseño, para la selección del lenguaje de programación, la metodología de selección empleada fue la matriz PUGH (Dieter, 2009), los elementos enfrentados para comparación fueron diferentes lenguajes de programación comunes en el desarrollo de este tipo de aplicaciones, la aplicación del método se resume en la siguiente tabla:

Tabla 1.

| Selección del lenguaje de programación |   |       |   |   |   |   |
|--|---|-------|---|---|---|---|
|  | A | B     | C | D | E | F |
| Multiplataforma                        | - |       | + | + | S | + |
| Desarrollo de GUI                      | + |       | + | + | + | - |
| Rapidez                                | + |       | + | + | S | + |
| Librerías a disposición                | - | DATUM | - | - | S | - |
| Experiencia del desarrollador          | - |       | - | + | S | S |
| Costo                                  | + |       | + | + | S | + |
| Σ+                                     | 3 |       | 4 | 5 | 1 | 3 |
| Σ-                                     | 3 |       | 2 | 1 | 0 | 2 |

Fuente: elaboración propia.



Donde las opciones de lenguajes de programación que se evaluaron fueron:

- C#. (A)
- Matlab®. (B)
- Python. (C)
- Java. (D)
- Labview®. (E)
- C++. (F)

Se observó que la opción predominante para los criterios de diseño elegidos fue JAVA. En una discusión al interior del grupo de diseño, este lenguaje fue seleccionado sin aplicar metodologías de selección adicionales por su característica de multiplataforma, que permite al desarrollo explicado en el presente artículo su extrapolación sencilla a proyectos en diferentes plataformas de *software*.

En la etapa de materialización del diseño de software se definieron los datos de entrada, los cuales correspondieron a registros de señales de EMG superficiales del músculo pronador redondo en antebrazo derecho, realizando movimientos de pronación siguiendo el protocolo propuesto en (Robin, Andres, & Mauricio, 2016). La adquisición de la muestra para las pruebas de funcionamiento se realizó por medio de electrodos superficiales de AgCl en configuración bipolar (Hargrove, Englehart, & Hudgins, 2007), en este caso 3 electrodos fueron empleados, 2 sobre la parte media del músculo y uno sobre la cabeza distal del hueso cúbito Figura 1.

Se utilizó la herramienta BiosignalPlux® para la adquisición de señales de EMG superficiales, la cual se configuró a una frecuencia de muestreo de 1 KHz y a una resolución de 16 bits, los datos obtenidos son enviados a un PC en donde el *software Opensignals*®, de descarga libre, guarda los datos en archivos con extensión TXT para ser abiertos luego por el *software BioMePro*. Este desarrollo fue hecho en el entorno NetBeans®, en donde se realizaron la escritura, compilación y depuración del programa.

Las funciones principales del *software* diseñado se listan a continuación:

1. Abrir los archivos de registro de las señales EMG en formato de texto plano.
2. Graficar los datos, que permiten hacer acercamientos en amplitud y tiempo.
3. Obtención de parámetros en el dominio del tiempo como por ejemplo nivel DC y RMS.
4. Filtrado de las señales con un banco preestablecido de filtros.

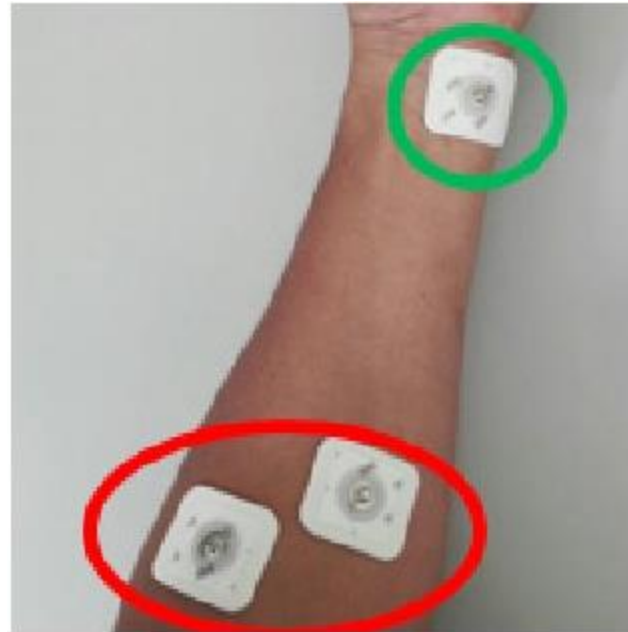


Figura 1. Disposición de los electrodos para el registro de la señal de EMG en antebrazo derecho ante movimientos de pronación.

Fuente: Robin, Andres.m & Mauricio, (2016).

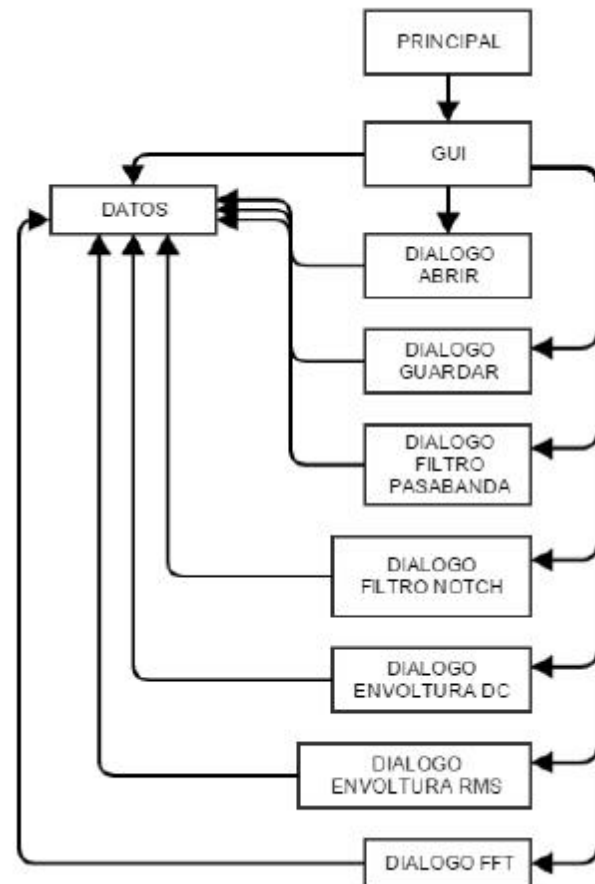


Figura 2. Diagrama de clase del software BioMePro.

Fuente: elaboración propia.

5. Armonizado de las señales con envoltura RMS o DC (Robin, Andres, & Mauricio, 2016).
6. Análisis de la transformada rápida de Fourier.
7. Guardar las señales procesadas y los registros obtenidos.

Para lograr los requerimientos del *software* propuestos, se diseñó una interfaz gráfica consistente de 10 clases (Figura 2). La clase principal configura los parámetros iniciales de funcionamiento del *software*, la clase correspondiente a la interfaz gráfica principal (diferente a la clase principal) contiene todos los componentes gráficos y eventos con los que interactúa el usuario, 7 clases adicionales de tipo interfaz gráfica, consistentes en cuadros de diálogo contienen la implementación de los módulos de funcionamiento propios del *software*, y finalmente, una clase adicional, contiene y comparte los datos con las demás clases por medio de la asignación de referencias a objetos. La interfaz gráfica de usuario se presenta en la Figura 3.

Los 7 diálogos presentados en la Figura 2, son los módulos internos del *software* los cuales comparten la misma referencia a la clase "datos", donde se encuentran los datos sin procesar y los datos del proceso de filtrado y armonizado. Una descripción breve de los diálogos se presenta a continuación:

**Diálogo abrir:** aparte de importar los datos, realiza la conversión a miliVoltios del valor entregado por el openSignals®, debido a que los datos registrados por el BiosignalPlux® vienen en unidades enteras del ADC del equipo. Esta conversión se logra con un ajuste lineal a partir de 2 parámetros: *span* (pendiente) y *offset* (desplazamiento), que son preguntados al usuario.

**Diálogo guardar:** permite construir un archivo de texto plano con los parámetros extraídos de la señal y su habilitación por parte del usuario.

**Diálogo Filtro PasaBanda:** dependiendo la frecuencia de muestreo ingresada por el usuario en la interfaz gráfica, este diálogo le indica la respuesta en frecuencia del filtro pasabanda pre programado.

**Diálogo filtro notch:** dependiendo de la frecuencia de muestreo, este diálogo le indica la respuesta en frecuencia del filtro, rechaza banda tipo Notch predefinido y espera su confirmación para ser aplicado.

**Diálogo envoltura D.C.:** este diálogo realiza un armonizado de la señal, consistente en reemplazar cada valor en la muestra por el promedio de la suma de los valores absolutos de los datos en su vecindad, el tamaño de la vecindad debe definirse menor al tamaño total de la muestra y suele ser un tamaño

significativamente menor (Universidad Militar Nueva Granada, 2016).

**Diálogo envoltura RMS:** este diálogo permite un armonizado de la señal a partir del cálculo del valor cuadrático medio de una pequeña ventana, de forma similar a lo explicado en el ítem anterior. (Universidad Militar Nueva Granada, 2016).

**Diálogo FFT:** realiza la Transformada rápida de Fourier (del inglés *Fast Fourier Transform* FFT), usando librerías y códigos para Java, desarrollados en Princeton University (Princeton University, 2016). Se debe tener cuidado de que el tamaño de la muestra a analizar corresponda a una potencia de 2, los resultados son presentados en un diagrama de magnitud de la FFT.

El banco de filtros fue diseñado en Matlab®, y está compuesto de 8 pares de filtros: un pasabanda y una *notch*. Son 8 pares porque hay 8 diferentes frecuencias de muestreo que el *software* admite: 300, 400, 500, 600, 700, 800, 900 y 1000 Hz. Los filtros pasabanda son de tipo FIR (*Finite Impulse Response*), diseñados para cada frecuencia de muestreo usando el método de filtros óptimos de rizado constante (Dipen, Shashwat, Siddharth, Sukrut, & Bhavdip, 2015), con una banda de paso de 50 a 150 Hz aproximadamente, ancho seleccionado debido a que en este rango se encuentra la información relevante para EMG (Robin, Andres, & Mauricio, 2016). Los filtros *notch* son de tipo IIR (*Infinite Impulse Response*) y fueron diseñados para rechazar la frecuencia de 60Hz presente en el sistema eléctrico de Colombia. Como estos filtros ya estaban preestablecidos al usuario, solo se le preguntó si los deseaba aplicar al mostrarle su respuesta en frecuencia por medio del diagrama de Bode de amplitud Figura 4.

## Análisis de resultados

Se observaron las gráficas de la muestra importada por el *software*, con sus respectivas opciones de acondicionamiento, como se muestra en la Figura 5, estos resultados gráficos fueron comparados con las gráficas obtenidas en Matlab® Figura 6.

Los parámetros relevantes obtenidos (Valor DC, Valor RMS, Tiempo de Duración y Frecuencia Pico) se guardaron en archivos de texto plano (Figura 7), para su uso posterior en otro tipo de programa que requiera los parámetros que caracteriza a los registros de señales de EMG.

También se compararon los tiempos de ejecución entre el *software* BioMePro y el programa ya realizado en Matlab®, realizando las operaciones de filtro pasabanda y

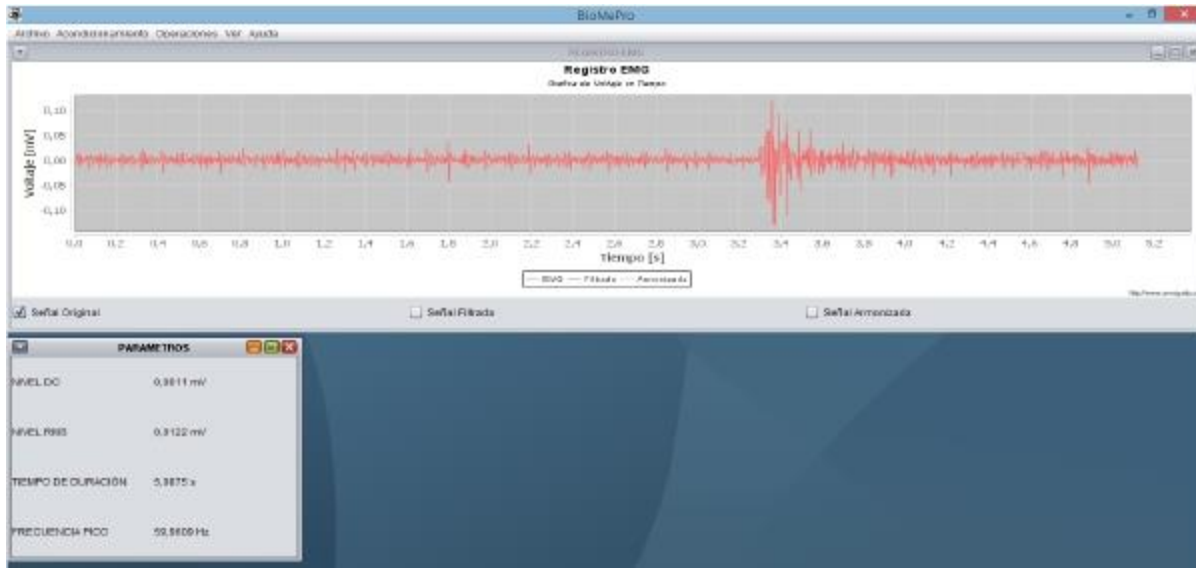


Figura 3. Apariencia del software BioMePro.  
Fuente: elaboración propia.

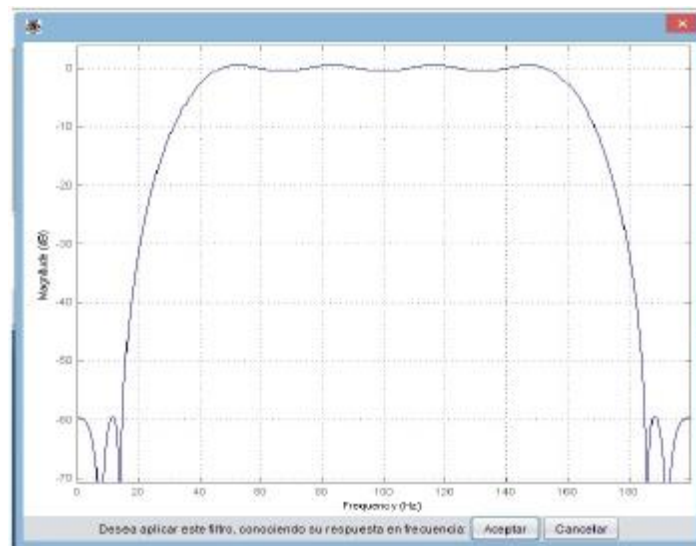


Figura 4. Ejemplo de la respuesta en frecuencia de uno de los filtros del banco preestablecido.  
Fuente: elaboración propia.

*notch*, y las armonizaciones por envoltura DC y RMS. Se obtuvieron los datos de la siguiente tabla:

Tabla 2.

Comparación de tiempo de ejecución.

|                  | BioMePro | Matlab®   |
|------------------|----------|-----------|
| Filtro Pasabanda | 8 ms     | 2.096 ms  |
| Filtro Notch     | 2 ms     | 4.68 ms   |
| Envoltura DC     | 6 ms     | 24.319 ms |
| Envoltura RMS    | 5 ms     | 7.018 ms  |

Fuente: elaboración propia.

Se observa que, aunque el proceso del filtrado pasabanda es más demorado en el *software* BioMePro, sin embargo, en el total de las operaciones el ahorro de tiempo es considerable y más si se proyecta implementar a futuro en aplicaciones en tiempo real.

## Conclusiones

Se obtuvo un programa de características similares a códigos realizados en Matlab® con la ventaja de que ahora son multiplataforma y son de registro libre porque ya no se requiere de la compra de la licencia del lenguaje de programación. Además, se logró un ahorro de tiempo considerable, el cual se puede aprovechar más si se consideran

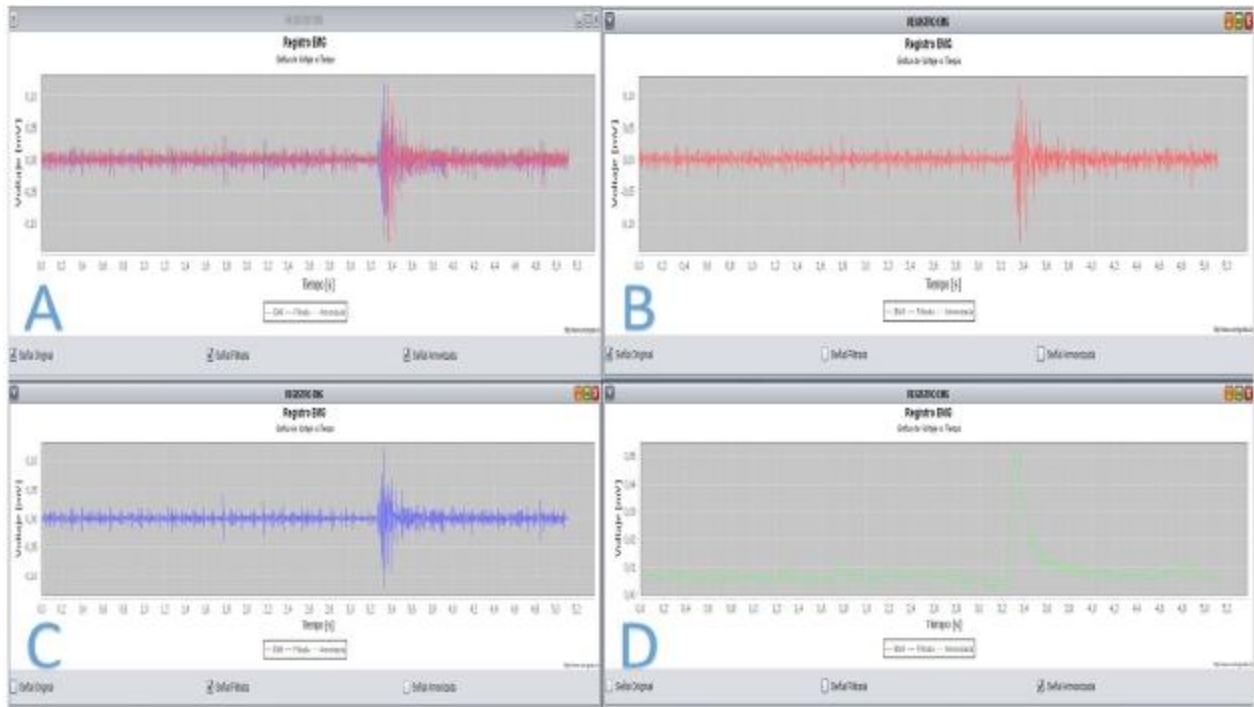


Figura 5. Opciones de visualización de un registro en BioMePro, (A) Visualización de la señal original, señal filtrada y señal armonizada, (B) Visualización de solo la señal original, (C) Visualización de solo la señal filtrada y (D) Visualización de solo la señal armonizada.

Fuente: elaboración propia.

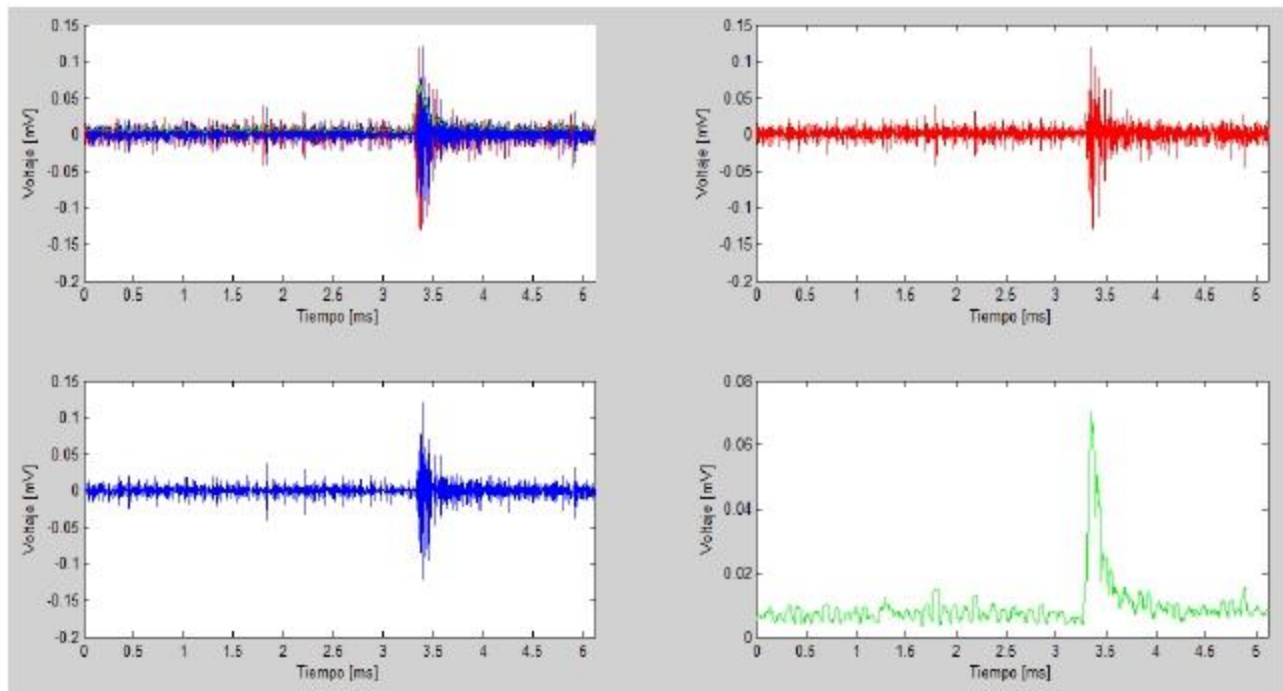


Figura 6. Opciones de visualización de un registro en Matlab®, (A) Visualización de la señal original, señal filtrada y señal armonizada, (B) Visualización de solo la señal original, (C) Visualización de solo la señal filtrada y (D) Visualización de solo la señal armonizada.

Fuente: elaboración propia.



```

salida: Bloc de notas
Archivo Edición Formato Ver Ayuda
Nivel DC:      0.0011133634425972917
Nivel RMS:    0.011981375006426775
Tiempo de duración: 5.0875
Frecuencia pico: 119.921875

```

Figura 7. Ejemplo de archivo de salida del software.

Fuente: elaboración propia.

propiedades de simetría en los coeficientes del filtro FIR para la operación de filtrado pasabanda.

Un aspecto importante que se ganó es que a la hora de comparar el peso del código realizado en Matlab® donde se obtuvo 44KB comparado al ejecutable BioMePro de 2,8 MB no parece ventajoso, pero si es analizado que ya no se requiere el *software* Matlab® para el procesamiento de las señales EMG, el cual es muy pesado e impensable colocar en un sistema embebido, el programa permite ganar la característica de portabilidad y de implementación, por ejemplo, en un RaspBerry Pi®.

Se comprobó que el programa corrió en varias plataformas Windows y Linux sin necesidad de instalar las librerías usadas en cada computador, comprobando la característica de multiplataforma de Java que era un requerimiento planteado para el *software*.

Este *software* puede ser implementado en trabajos posteriores como el diseño de señales de control para sistemas robóticos tipo exoesqueletos, que requieren de un procesamiento muy robusto de señales electrofisiológicas como las EMG para la interacción con el usuario, que requieran posibilidades de procesamiento embebido.

## Referencias

- Baturone, I., & Gersnoviez, A. (2010). Diseño de Sistemas Neuro-Difusos para Control de Robots mediante XFUZZY 3. *Tecnología, aprendizaje y enseñanza de la electrónica*. Sevilla.
- Blanco, R. (2016). *Diseño de un sistema de predicción del rendimiento deportivo a partir de técnicas de inteligencia computacional en selecciones UMING*. Bogotá: Propuesta de Tesis, Maestría en Ingeniería Mecatrónica, Universidad Militar Nueva Granada.
- D. K., S. G., S. K., S. G., & B. G. (2015). Design of Equiripple Linear-Phase FIR Filter Using Computer Aided Method. *Communications and Signal Processing (ICCSP)*, (págs. 0001-0005). Melmaruvathur, India.
- Dieter, G. E. (2009). *Engineering Design*. Boston: McGraw-Hill.
- Figueroa-García, I., Aguilar-Leal, O., Hernández-Reynoso, A., Madrigal, J., Fuentes, R., Huegel, J., & García-Gonzalez, A. (2014). Platform for the Study of Virtual Task-Oriented Motion and its evaluation by EEG and EMG Biopotentials. *IEEE Engineering in Medicine and Biology Society, 36th Annual International Conference*. Chicago, USA.
- Hargrove, L. J., Englehart, K., & Hudgins, B. (2007). A Comparison of Surface and Intramuscular Myoelectric Signal Classification. *IEEE transactions on biomedical engineering*, vol. 54, no. 5.
- Kazerooni, H., & Strausser, K. A. (2011). The Development and Testing of a Human Machine Interface for a Mobile Medical Exoskeleton. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, USA.
- Momen, K., Krishnan, S., & Chau, T. (2007). Real-Time Classification of Forearm Electromyographic Signals Corresponding to User-Selected Intentional Movements for Multifunction Prosthesis Control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(4), 535-542.
- Princeton University. (16 de 11 de 2016). *Introduction To Programming in Java*. Obtenido de <http://introcs.cs.princeton.edu/java/97data/>
- R. B., A. C., & M. P. (2016). Diseño de un sistema difuso para el reconocimiento de la actividad muscular. *IJIAS*, 19(2).
- Romo, H. A., Realpe, J. C., & Jojoa, P. E. (2007). *Análisis de Señales EMG Superficiales y su Aplicación en Control de Prótesis de Mano*. 14(1).
- Saikia, A., Mazumdar, S., Sahai, N., Paul, S., & Bhatia, D. (2016). Comparative Study and Feature Extraction of the Muscle Activity Patterns in Healthy Subjects. *3rd International Conference on Signal Processing and Integrated Networks*. Noida, India.
- Universidad Militar Nueva Granada. (2016). Manual de Usuario de Software BioMePro. Bogotá.