

## NOTA TÉCNICA

## FREEMOTIONLIBS

### Una Biblioteca para el Análisis del Movimiento

Darwin Rocha<sup>1</sup>, Ana Aguilera<sup>1</sup>

(Recibido Marzo 2013, Aceptado Junio 2013)

<sup>1</sup>Centro de Análisis, Modelado y Tratamiento de Datos, Facultad de Ciencias y Tecnología, FACYT, Universidad de Carabobo, Venezuela  
[dawinrocha@gmail.com](mailto:dawinrocha@gmail.com), [aaguilef@uc.edu.ve](mailto:aaguilef@uc.edu.ve)

---

**Resumen:** El análisis cuantitativo del movimiento humano, dentro de la biomecánica, es un área que revela un avance significativo en el estudio del movimiento, ofreciendo medidas precisas que apoyan el diagnóstico/detección en diversos campos como el deportivo, médico, industrial, entre otros. Para apoyar en esta tarea han surgido en el mercado y a nivel académico diferentes soluciones informáticas que ofrecen diferentes fortalezas y debilidades que condicionan su utilidad y aplicación. Un desarrollo inspirado en el sistema propietario Vicon© fue elaborado por el Dr. Franck Barbier, capaz de realizar cálculos sobre señales capturadas y almacenadas y la visualización en 3D del movimiento. Sin embargo este software, presentado en una versión standalone, tiene una serie de inconvenientes que pretendemos resolver en el presente trabajo. Para ello mediante una reingeniería del código y el establecimiento de patrones precisos de diseño/desarrollo/prueba se ha generado una nueva versión web, con facilidades para conexión a base de datos, entre otras.

**Palabras clave:** Biomecánica/ Análisis de Movimiento/ Visualización C3D.

---

## FREEMOTIONLIBS

### A Library for Motion Analysis

---

**Abstract:** Quantitative analysis of human movement in biomechanics, is an area that reveals a significant advance in the study of movement. It provides precise measurements that support the diagnosis/detection in various fields such as sports, medical, industrial, among others. To assist in this task, it has emerged in the market and at different academic level solutions that offer different strengths and weaknesses that determine their utility and their application. A development inspired by the Vicon© property-system was developed by Dr. Franck Barbier. It is capable of performing computations on signals captured and stored and the 3D visualization of the movement. However this software, presented in a standalone version, has a number of drawbacks that intend to solve in this paper. To achieve this goal, a re-engineering the code and establishing precise patterns of design/development/test have generated a new web version with database access and other functionalities.

---

**Keywords:** Biomechanics/ Movement Analysis/ C3D Visualization

#### I. INTRODUCCIÓN

La biomecánica es una ciencia multidisciplinaria, en donde confluyen principios y reglas de otras ciencias, como física, medicina, ergonomía, entre otras, y que realiza aportes significativos a áreas como: robótica, fisioterapia y computación, entre otras (Aggarwal & Cai, 1999). La biomecánica como ciencia, se encarga del estudio del movimiento de los cuerpos de los seres vivos y sus respectivas implicaciones, en el caso del ser humano estudia todas las estructuras de carácter mecánico del cuerpo, buscando resolver los problemas derivados de las diversas condiciones a las que puede verse sometido, así como los diferentes movimientos y las fuerzas que actúan sobre él (Villarroya, 1996).

El estudio del movimiento humano, implica un análisis

teórico del movimiento del individuo (generalmente, realizado por un experto), la captura del movimiento y la realización de cómputos sobre los datos ya capturados (Vaughan & O' Connor, 1999). Esto se hace con el fin de procesarlos y obtener indicadores que permitan hacer un estudio cuantitativo y cualitativo del movimiento. Cada especie animal tiene su propia forma de locomoción, siendo la bipedestación la correspondiente al hombre, el Dr. Václa Vöjta ofrece una buena descripción de las funcionalidades del cuerpo humano (Rodríguez, 2009).

Para la captura del movimiento se han creado sistemas de adquisición y procesamiento. Un ejemplo de ellos es el ofrecido por la empresa VICON© (vicon, 2012) que a través de sistemas de sensores y dispositivos, realiza la captura de una serie de datos para modelar digitalmente el

movimiento realizado. Generalmente, la información derivada en la captura es almacenada en un archivo con formato C3D (C3D, 2012), el cual se ha convertido en un estándar para la captura del movimiento humano. Cabe destacar que los datos obtenidos son de tipo antropomórficos o provenientes de los segmentos esqueléticos rígidos en base a su antropometría (estudio de la medición del cuerpo humano en términos de las dimensiones del hueso, músculo y tejido adiposo).

Para el análisis de los datos capturados, se han hecho investigaciones para procesarlos de forma correcta y se han creado implementaciones de software para agilizar este proceso. Sin embargo, estos productos de software, son de difícil acceso (sea por costos o por portabilidad) o con funcionalidades limitadas. El doctor Frank Barbier (Barbier, 2005) desarrolló un software standalone, que permite realizar una serie de cálculos sobre los datos del movimiento y tiene opciones de visualización del archivo C3D. Sin embargo este software presenta una serie de restricciones que limitan su uso, accesibilidad y utilidad. No fue diseñado para múltiples usuarios, lo que limita su accesibilidad; requiere la instalación de otros componentes privados de software para su operatividad y funciona sobre un solo sistema operativo, lo que compromete su portabilidad; carece de una interfaz intuitiva y necesita direcciones específicas para la localización de sus archivos de entrada, esto dificulta su uso; no cumple con las normas internacionales sobre el desarrollo de productos de software y su estructura dificulta en gran medida la revisión y actualización del software. MOKKA y su librería B-TK es un software desarrollado por el sr. Arnaud Barré (Barre, 2012), que permite visualizar la información del C3D, pero solo puede realizar un número muy reducido de cálculos. Visual3D™ (Visual3D, 2012) es un avanzado software de análisis clínico de datos biomecánicos de movimiento en 3D. Con Visual3D™ se puede dar la flexibilidad para analizar con precisión los movimientos 3D usando datos adquiridos, técnicas de análisis, juegos de marcadores y los sistemas para definir modelos biomecánicos y generar resultados de análisis, pero al ser un software propietario limita su accesibilidad. Hay otras soluciones menos especializadas que permiten realizar ciertos cálculos sobre los datos del movimiento, hechas en Matlab © o maple© pero su corto alcance y la dificultad que representa su uso las hacen poco viables.

El objetivo de este trabajo es desarrollar un software para el análisis y visualización de datos derivados del movimiento, que mejore la accesibilidad de la aplicación a través de un acceso remoto vía web o intranet y la accesibilidad de los datos a través del uso de bases de datos integradas en la aplicación y búsqueda guiada de archivos de interés, mediante una interfaz sencilla y amigable. Para ello se propone partiendo del software desarrollado por Barbier, la aplicación de una reingeniería del código que lo pase, en primer lugar, a una versión de C/C++ de distribución libre y en un segundo lugar, a una aplicación con una interfaz web. La operatividad de la

aplicación alojada en un servidor facilitará su acceso remoto a los diferentes utilizadores, con capacidades para efectuar funciones de cálculo, la visualización del movimiento 3D en un navegador web y el registro de datos integrados en una base de datos, facilitando su uso posterior.

## II. DESARROLLO

Para el desarrollo de este trabajo se fijaron las siguientes metas para el logro del objetivo general: a) Estudiar el funcionamiento de la versión más reciente del software, utilizando casos de prueba reales para establecer un patrón de seguimiento. b) Identificar las rutinas que debían ser implementadas, eliminadas u optimizadas, para facilitar la extracción de los segmentos requeridos en la creación de la biblioteca. c) Crear con los segmentos de códigos extraídos, los diferentes módulos de la biblioteca. d) Probar que el producto obtenido es de calidad y cumple cabalmente con los requisitos exigidos. e) Desarrollar una interfaz de uso amigable para que los usuarios tengan acceso a la biblioteca, que facilite la selección de las diferentes opciones del sistema como visualizar, cargar un lote, cargar un solo caso, entre otros. f) Probar el correcto funcionamiento de la interfaz de usuario para garantizar que el producto obtenido, es funcional y cumple cabalmente con los requisitos exigidos. A continuación se detallarán las actividades realizadas y los principales resultados obtenidos en la desarrollo de cada etapa para el cumplimiento de las metas.

### A. Estudio del funcionamiento de la versión más reciente del software

#### 1) La biblioteca de funciones

El principal aporte de este trabajo es el conjunto de funciones desarrolladas para el estudio del movimiento. Las mismas fueron traducidas a C/C++ para efectos de portabilidad, sin embargo la esencia de las mismas se conserva de la misma manera en el software mejorado. A continuación un resumen de las mismas.

Las funciones desarrolladas en este trabajo se utilizan para el estudio de la cinemática y la dinámica tridimensional de sistemas multicuerpos. Todas las entidades procesadas por el software son representadas como vectores. Las rotaciones tridimensionales se formalizan usando cuaterniones también llamados parámetros de Euler, con el fin de evitar la posible indeterminación relacionado con otros formalismos (por ejemplo, ángulos de Euler). Los cálculos cinemáticos y cinéticos se realizan utilizando un formalismo matricial.

Cinco entidades informáticas permiten almacenar los resultados de los cálculos:

- Los segmentos (orientación, velocidad angular, aceleración angular usando cuaterniones, masa acelerada, cantidad de movimiento lineal y angular).
- Puntos (posición, velocidad, aceleración, fuerzas y pares aplicados al punto).

- Las variables son entidades en función del tiempo que permiten almacenar todos los resultados de los cálculos más allá de las soportadas por los segmentos y puntos.
- Los valores que son entidades que almacenan un valor puntual. Por ejemplo, el resultado del análisis morfológico de una señal (el máximo, el mínimo, etc).
- Las Matrices permiten almacenar los resultados de los cálculos matriciales. Las matrices cambian los puntos de referencia identificados con las experimentaciones preliminares para recalculer los centros de rotación con un número reducido de marcadores.

Noventa y un funciones están disponibles actualmente para el usuario. Estas se pueden agrupar en categorías amplias:

- El cálculo vectorial.
- El cálculo de la matriz.
- La cinemática de sólidos.
- La cinética de sólidos.
- Análisis de la morfología de las señales.
- El filtrado y suavizado de señales.
- Exportación de datos en formatos utilizados directamente en Excel y Statistica.

Para facilitar el desarrollo, el software permite que los cálculos se puedan ejecutar por partes. Así una vez desarrollados y validados los tratamientos, es posible ejecutar los cálculos (por lotes) sobre todas las experimentaciones en estudio y generar automáticamente los archivos de salida para su análisis.

## 2) Plan de Reingeniería

Para la realización del estudio del sistema base se procedió a crear un plan de reingeniería con los siguientes pasos: a) Revisar la estructura del código, que permita determinar cómo está organizado el código y la ubicación de los datos de entrada. b) Revisar que

metodología de desarrollo se usó, esto consiste en ver cómo está diseñado el código. c) Hacer un seguimiento de caja blanca, para detectar cuáles son las rutinas y procedimientos a seguir y qué código se encuentra obsoleto, cuál hay que reescribir y cuál hay que implementar. d) Establecer las mejoras a realizar en base a lo actual, para que el nuevo diseño cumpla con estos parámetros.

## 3) Evaluación del Sistema Base

El software a evaluar, está en una versión standalone, y se encuentra desarrollado en un framework de C/C++ llamado LabWindows. Los Requisitos del software del Dr. Barbier para poder ser instalado son: a) **Software:** Sistema operativo Windows, LabWindows, Matlab. b) **Hardware:** Procesador intel dual core o similar, al menos RAM 512 MB, espacio en disco superior a 5 GB.

El software original está compuesto por dos módulos, uno cuya principal labor es la realización de evaluaciones cuantitativas sobre los datos capturados del movimiento y el otro es un módulo de visualización en 3D de estos datos.

La interfaz original fue creada con LabWindows y se compone de tres pantallas:

### a) Carga de los datos al sistema.

En esta interfaz de la Figura 1, se tiene un menú de opciones, que permiten subir al sistema, los diversos archivos de datos, todos preexistentes. El sistema tiene la limitación de no permitir la creación de los mismos. El sistema sólo admite 4 tipos de archivo el “.par” que contiene los datos de parametrización del sistema, el C3D donde se encuentran capturados de forma digital los datos del movimiento del individuo, el morfo que contiene los datos morfológicos y un .bat que permite la carga en lote.

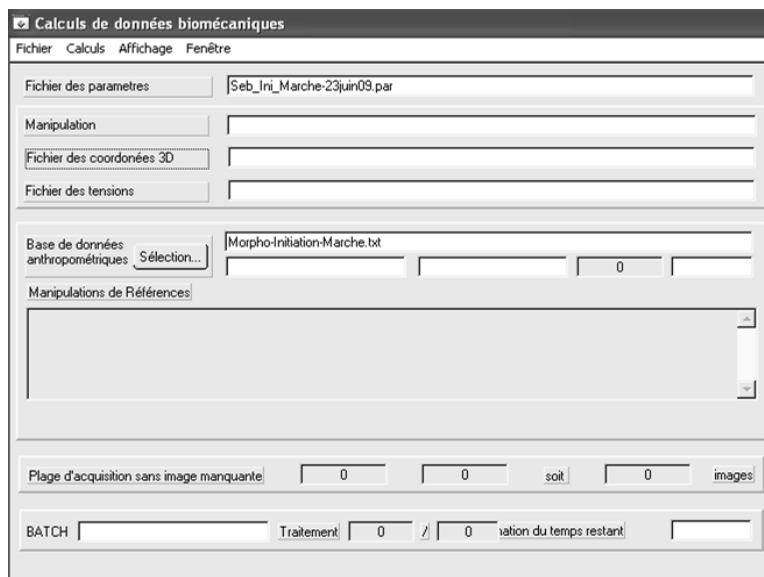
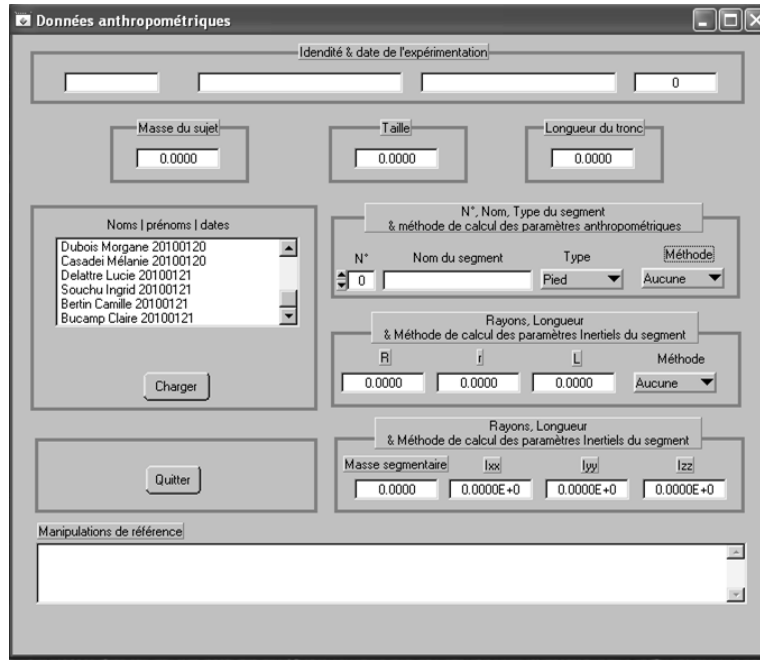


Figura 1. Interfaz Principal del Software original

**b) Carga de datos morfológicos**

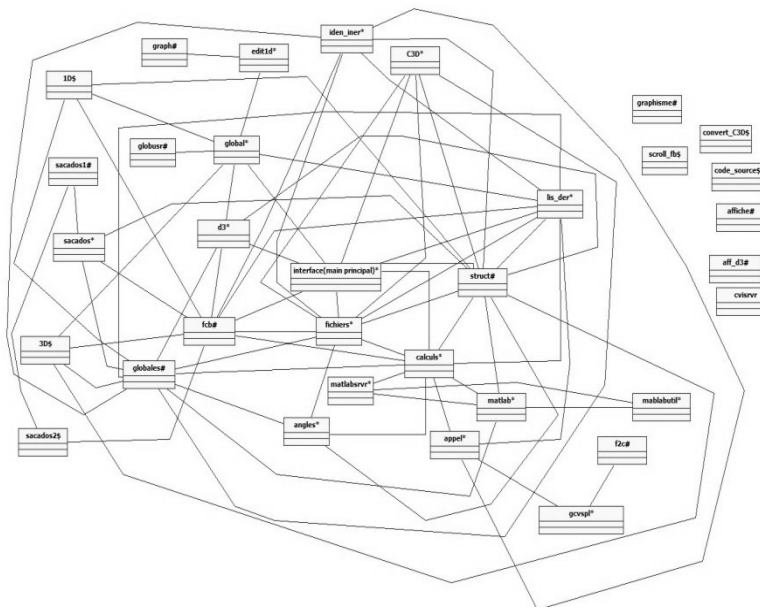
En esta interfaz de la Figura 2, se pueden observar los datos morfológicos del individuo, se puede seleccionar al individuo de una lista que se muestra, pero dicha lista no está en una base de datos sino en un archivo plano de texto. En la parte superior se puede observar, los datos de

identificación y edad del individuo, más abajo datos morfológicos, como peso, talla, longitud de tronco. Del lado derecho, se observa el número, nombre del segmento, tipo y método a aplicable. Así como largo, masa y marcadores que conforman el segmento.



**Figura 2. Interfaz Principal de los morfológicos del individuo**

Los datos de salida, son guardados en archivos con diversos formatos, una de las fallas que presenta este software es que ni los archivos de entrada ni los de salida se encuentran organizados de forma alguna, todos están en la misma carpeta juntos con los códigos del sistema, y a su vez, estos no tienen ninguna organización, siendo muy difícil establecer un patrón de seguimiento. En la Figura 3 es posible observar esta situación.

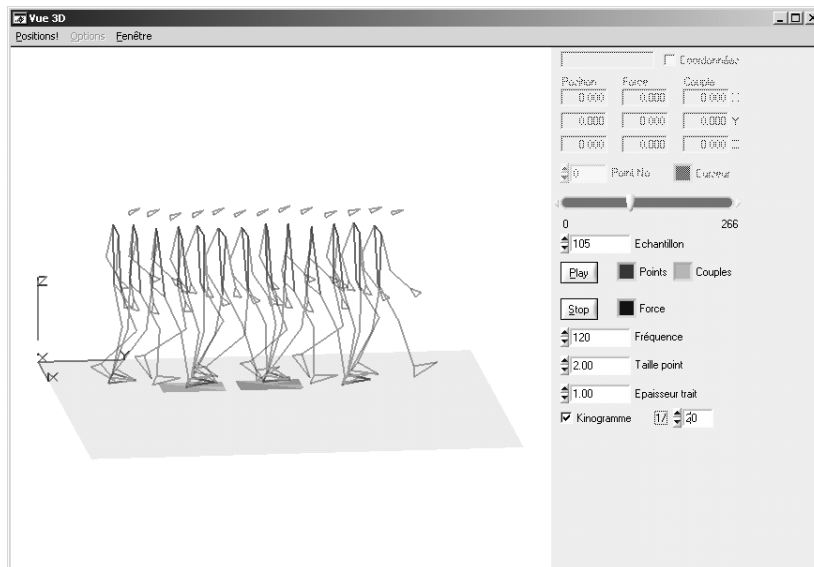


**Figura 3. Organización de los archivos en el sistema del Dr. Barbier**

Cada módulo es un archivo del sistema, y la unión entre ellos es la forma en que se incluyen, no tiene un direccionamiento puesto que la inclusión es mutua, los archivos que se encuentran conectados son 26, mientras que hay 7 que no tienen conexión con ningún otro archivo, estos últimos se conservaron porque están dentro de la carpeta del sistema, pero no se evidenció su utilidad práctica.

**c) Visualización del individuo realizando los movimientos en un plano 3D**

Esta interfaz permite mostrar de manera gráfica la experimentación en estudio. Muestra la coherencia de los puntos calculados y la definición de los segmentos, como se observa en la Figura 4.



**Figura 4. Interfaz gráfica del software del Doctor Babier**

**4) Flujo de Datos**

Al realizar el estudio, se evidenció el flujo de datos dentro del software, quedando establecidos de la siguiente forma:

*Paso 1:* Lectura del archivo de parámetros: aquí se lee del archivo con extensión .par toda la información que parametriza el sistema, esto es, las funciones a realizar, así como los segmentos a evaluar, el tipo de cálculos a realizar y los marcadores a considerar, por último los archivos de salida a generar.

*Paso 2:* Lectura del archivo C3D, aquí se lee toda la información capturada de forma biométrica, Para los efectos de este trabajo se toman los parámetros del archivo C3D por grupo (ver Tabla I).

Así el C3D se organiza en "grupos" (cada uno de los parámetros dentro de un archivo C3D pertenece a un grupo de parámetros particular). Cada grupo de parámetros relacionados tiene un nombre de grupo único y puede tener una descripción del grupo asociado a él. Un paso importante es sincronizar y verificar la información con la leída del .par. Esto consiste en tomar la información del C3D, sólo de aquellos marcadores indicados en el .par (si el C3D tuviese más marcadores estos se ignoran. Para lograr esto se lee el bloque de datos de vídeo, el cual contiene una estructura en bloques de 16

bytes para cada marcador en cada imagen capturada, el volumen total en bytes de información que contiene la sección de datos de video, se calcula de la siguiente forma mostrada en la ecuación (1), siendo N número de marcadores, M el número de imágenes capturadas y T el tamaño total de datos contiene la sección de datos de video:

$$T = N \times M \times 16 \text{ bytes. (1)}$$

**Tabla I. Especificación de los grupos y sus parámetros**

<b>Estructura de la Información Dentro Sección de Parámetros del C3D</b>	
<i>Grupos</i>	<i>Parámetros</i>
ANALOG	OFFSET, SCALE
FORCE_PLATFORM	USED, TYPE, CORNERS, ORIGIN, CHANNEL
EVENT	USED, CONTEXTS, LABELS, TIMES
POINT	LABELS

De cada marcador se guardan: 12 bytes que representan la posición x, y, z o el factor de escala para x, y, z y el cuarto grupo de bytes representa las 2 primeras cámaras que mide el marcador (1 bit para cada cámara, y los dos últimos residual promedio de la medición de punto/Factor escala) ver Tabla II.

Tabla II. Representación de un bloque de data de video

Posición en el espacio del marcador j en la imagen i (16 bytes)			
x (4 bytes)	y (4 bytes)	z (4 bytes)	Residual (4 bytes)

Para leer esta información, contenida en bloque de datos de video, se realiza el siguiente ciclo que se observa en la Figura 5.

Para fines de esta investigación los últimos 4 bits, leído en la variable FC se ignoran, en este segmento se debe alinear los puntos que viene de la lectura del archivos de parámetros.

*Paso 3:* Lectura del archivo que contiene las especificaciones morfológicas de la persona, como talla, largo de tórax, estatura, etc.

*Paso 4:* Cálculos. Se componen de tres sub pasos: a) Cálculos preliminares. b) Cálculos de puntos c) Cálculos posteriores. Estos cálculos permiten fijar la relación entre los puntos, su distancia en cada parte del ciclo de la marcha, el tipo de desplazamiento que tiene, entre otros.

El software ofrece la posibilidad de tratamiento por lote, esto es, se toman un conjunto de datos de entrada en un archivo, con el fin de repetir el proceso por cuántos casos tenga el lote. Aquí es importante señalar que este flujo de datos se encuentra repartido entre diversos archivos de códigos.

En la pantalla de visualización se tiene un plano en 3D, que permite observar los movimientos que realiza el individuo a partir de los marcadores colocados en su cuerpo, este plano tiene la capacidad de girar y rotar en 360 grados.

```

Int M; // número de imágenes
Int N; // número de marcadores
para i=0 hasta i= M
{
    para j=0 hasta j= N
    {
        leer (x);
        leer (y);
        leer (z);
        leer (FC);
    }
}
    
```

Figura 5. Algoritmo de lectura para un bloque de data de video

**B. Identificar las rutinas que debían ser implementadas, eliminadas u optimizadas, para facilitar la extracción de los segmentos requeridos en la creación de la biblioteca**

Para cada etapa del flujo de trabajo se identificó, cuáles rutinas pertenecían al framework usado, ya que debían ser re-implementadas puesto que no se dispone del código que ejecutan, como por ejemplo spline (que realiza un interpolación tipo spline), FTT (que realiza la transformada de Furier), ambas rutinas propias del framework. Se identificaron además cuáles no se utilizaban o eran innecesarias y cuáles debían ser optimizadas en un mejor proceso.

**Los procedimientos**

En esta parte se explica cómo se realizan los procedimientos y las funciones una vez leídos los datos de entradas. Los procedimientos son aquellos que realizan los diversos cálculos. Los cálculos se realizan en dos etapas: los preliminares y los posteriores, sin embargo corresponden a los mismos procedimientos, que son los usados por todo el sistema. Los procedimientos se invocan a través del método leído en el archivo de parámetros y en total son 91 métodos, que se componen de procedimientos y rutinas, que a su vez poseen subrutinas. En este trabajo se hace especial observación a las nativas del framework ya que éstas deberán ser re-implementadas. Mientras que el resto deberán ser sólo reestructuradas. La Tabla II presenta una descripción de la clasificación de las funciones identificadas en el sistema base.

**C. Crear con los segmentos de códigos extraídos, los diferentes módulos de la biblioteca**

Las rutinas y segmentos de códigos, que se tomaron del paso anterior, se usaron con el fin de crear los diferentes módulos del nuevo sistema. Este está compuesto de 4 módulos: a) lectura: encargado de leer los datos de entrada del sistema por sus diferentes medios (selección de archivo, creación de archivo, archivo subido por el usuario); b) procesamiento: encargado de realizar los cálculos solicitados por el usuario; c) escritura: encargo de escribir los archivos de salida y entregarlos al usuario y por el último; d) visualización: encargado de la presentación gráfica de los datos del individuo en el plano 3D.

Para los efectos del desarrollo y codificación del nuevo código, se hará una reestructuración con un enfoque orientado a objetos y en software de libre distribución. En el primer caso, se tomará ventajas de la orientación a objeto como lo es el encapsulamiento de los datos y métodos necesarios, el re-uso, la escalabilidad, la correcta modularización del software, entre otros y por otro lado, la consideración de libre distribución repercutirá favorablemente en los costos de desarrollo y mantenimiento.

En la recodificación, se crearon los objetos de los módulos, se cambiaron las variables de globales a locales, ya permite mayor control de los datos y su flujo. Se cambió de un sistema semi-estructurado, a un sistema totalmente orientado a objetos, con una programación clara y legible para futuras posibles modificaciones. En esta sección se realizaron cambios con fines pragmáticos, las lecturas de datos de entrada se encapsularon en objetos de C++. Esto hace que la lectura sea realmente independiente de los otros datos de entrada y sus métodos puedan ser tomados para otros fines.

**1) Estructuración de Archivos**

Para este trabajo se optó por la organización propuesta por el modelo-vista-controlador (MVC) para la estructuración del código, ya que permite separar la interfaz de usuario de los métodos y procedimientos que se realizan. Adicionalmente se colocaron archivos por carpetas para mejorar su ubicación y localización.

No se estableció ninguna jerarquía entre los archivos de códigos, sólo se distinguen en grandes grupos y estos en subgrupos (ver Tabla III), el caso particular de la interfaz se sub-divide en dos (MVC y graficación) porque los archivos de graficación requieren estar en una carpeta específica fuera del modelo MVC, para hacer un llamado correcto al motor gráfico.

**2) Visualización del archivo C3D**

Para realizar la visualización del archivo C3D, es necesario realizar una serie de pasos que se enumeran a continuación:

Tabla III. Clasificación de funciones usadas en el sistema base

Grupos de Funciones	Descripción
<b>Funciones de Entrada</b>	Son las encargadas de hacer lectura de los datos de entrada, bien sea por pantalla (nombre de los archivos) o las funciones que leen los archivos propiamente
<b>Funciones de Salida</b>	Son aquellas que escriben los archivos de salida, no realizan ningún procedimiento más que escribir en los diversos archivos
<b>Funciones Nativas de C/C++</b>	Son aquellas propias del lenguaje C/C++ que se utilizan para crear ciertos datos y realizar ciertos cálculos
<b>Funciones Creadas</b>	Son aquellas que realizan cómputos, y fueron creadas por el Dr. Barbier, creador del software base.
<b>Funciones Nativas del LabWindows</b>	Son aquellas que el framework ofrece para la realización de cálculos

**a) Lectura del C3D**

La lectura del C3D se realiza tal como se describe en las secciones previas sólo que se hace especial énfasis en los points:labels y en la data analógica, ya que en estas dos secciones del C3D es donde se encuentra los datos que nos permitirán graficar.

**b) Escritura del archivo del XML**

Como valor agregado a este desarrollo se estableció un archivo XML para pasar los datos entre su lectura y su graficación. La utilización de XML permite la creación de etiquetas entre las cuales se puede colocar la información necesaria. El algoritmo utilizado para este paso se detalla a continuación en la Figura 6.

```

Int N; // número de puntos
Int M; // número de imágenes que contiene los puntos
Para i=0 hasta N
{
    Escribir ("<etiqueta_inicio>");
    Para j=0 hasta M
    {
        Escribir valor [i][j];
    }
    Escribir ("<etiqueta_fin>");
}
    
```

Figura 6. Sección del algoritmo de escritura del XML

La “<etiqueta\_inicio>” debe tener la siguiente forma “<nombre del punto>” ejemplo “<nombre1>” y “<etiqueta\_fin>” debe tener la siguiente forma “</nombre del punto>” ejemplo “</nombre1>”, la matriz valor es de tamaño N x M, y cada celda de la matriz contiene el valor analógico o de posición 3D para el punto representado en la fila. El resultado de la ejecución del algoritmo se presenta en la Figura 7.

```

<Front>
<X>191.903 192.334 192.745 ....</X>
<Y>-302.527 -292.554 -202.59 ... </Y>
<Z> 1695.04 1695.80 1696.54 .... </Z>
</Front>
    
```

Figura 7. Archivo XML creado para la visualización de puntos

En esta sección de data de video se establecen dos etiquetas con el nombre del marcador, por ejemplo <Front> para el inicio y </Front> para el cierre, dentro de estas dos etiquetas se colocan las etiquetas para las coordenadas: <X>, <Y> y <Z>, con su respectivos cierres </X>, </Y> y </ Z>, dentro de cada coordenada se coloca, separados por un espacio simple, los valores que toma ese marcador en esa coordenada en cada fotograma, en el ejemplo de la Figura 7 se colocan por razones de espacio solo los valores de los tres primeros fotogramas, pero se colocan tantos valores como fotogramas existan.

**c) Creación del archivo JS**

Al tener listo el XML, para proceder a graficar se hace necesario crear un archivo en JavaScript 3, que será el encargado de generar la conexión con el motor gráfico del cliente y a la vez será el encargado de proyectar esta información en un plano tridimensional del navegador.

**d) Creación de los gráficos**

Este paso es solo de conexión, consiste en tomar todos los archivos creados añadirlos como cabeceras en el archivo base e invocar al motor gráfico. Una resultado de esto es la representación gráfica generada en un navegador como se observa en la Figura 8, donde se puede observar un plano en 3 dimensiones y la representación de una persona (puntos blancos) realizando un movimiento.

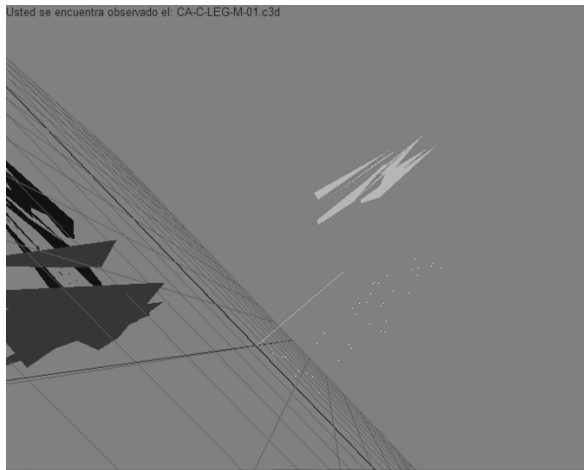


Figura 8. Visualización del C3D en el navegador

**D. Probar que el producto obtenido es de calidad y cumple cabalmente con los requisitos exigidos**

Una vez creada la biblioteca en C/C++, se aplicó la norma ISO-9126, que establece la calidad del software, obteniendo una puntuación 20 de 21, para más detalles sobre estas pruebas ver (Rocha, 2013)

**E. Desarrollar una interfaz de uso amigable para que los usuarios tengan acceso a la biblioteca, que facilite la selección de las diferentes opciones del sistema como visualizar, cargar un lote, cargar un solo caso, entre otros**

Se creó una interfaz web en PHP 5.3 y HTML 5, que permitiera al usuario cargar los diferentes datos de entrada, así como su selección si éstos ya existiesen, y la selección de las diferentes opciones del sistema.

**La Interfaz de Usuario**

La nueva interfaz de usuario se creó usando los siguientes lenguajes:

- Carga de datos: php 5.3+, HTML y JavaScript.

- Visualización del archivo C3D: php 5.3+, html5, XML, WebGL y javascript3.

Esto facilita el desarrollo de una interfaz Web sencilla y fácil de usar por el usuario. Parte de las mejoras incluyen la presentación de las opciones con despliegue de menú. Así el uso del sistema es más amigable y son más accesibles las opciones. La nueva interfaz de usuario ofrece otras funcionalidades no presentes en la versión original y que se detallan a continuación:

- Autorización de usuarios. Con el fin que cada usuario tenga su propia sección a la hora de usar el software y afin de gestionar la seguridad y restringir el acceso a la aplicación. Esto facilitará la adaptación futura al manejo de perfiles de utilizador, mantenimiento de sesiones y otras ventajas.
- Selección del archivo C3D, cuando éste ya hubiese sido usado con anterioridad.
- Selección del archivo de parámetros: Se incorporó la opción de subir un archivo de parámetros al sistema, seleccionar uno de los que ya fueron subidos al sistema o crear un archivo por medio del sistema. Este último una vez creado pasará a la lista de archivos existentes en el sistema observar las Figuras 9 y 10.

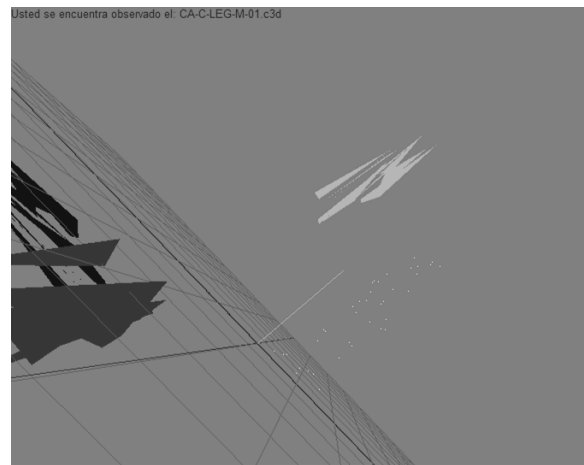


Figura 9. Menú de Opciones, luego del acceso

- Selección del archivo de datos morfológicos: aquí hubo una mejora al sistema existente, ya que anteriormente sólo permitía tomar aquellos archivos guardados en una dirección predefinida. En la nueva opción se puede: a) subir uno al sistema y esta información queda reflejada de forma automática en la base de datos, ya que esto permite tener mayor control de los datos y hacer un buen resguardo; b) seleccionar uno ya existente en el sistema; c) crear uno por medio del sistema que al momento de ser creado se añade al sistema y a la base de datos o seleccionar uno de los datos existentes en la base de datos, que crea un archivo que queda añadido al sistema.
- Carga en lote: al igual que en los casos anteriores ahora el usuario podrá cargar un archivo de lote (.bat) al



sistema, seleccionar uno del sistema o crear uno, aquí es importante decir que hubo un ligero cambio, ya no se necesita especificar la ubicación de los archivos que componen el lote, el nuevo sistema se encarga de eso ahora.

- Visualización del C3D: anteriormente para visualizar un C3D, se requería que el archivo de parámetros estuviese, en la nueva versión, no es necesario ya que el sistema se encarga de leer y hacer las configuraciones necesarias.

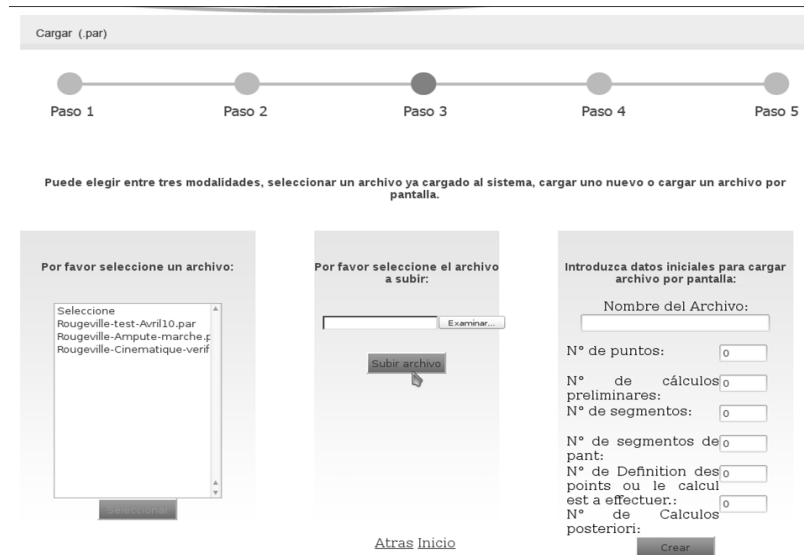


Figura 10. Ventana para cargar el archivo de parámetros

### Mejoras realizadas al sistema existente

A continuación se detallan algunas de las mejoras más significativas incorporadas en la nueva versión del código. Es importante señalar que las funcionalidades del sistema anterior fueron conservadas.

- Cambios en la estructura del código. Se realizó una recodificación de la mayor parte del sistema, se suprimieron las variables globales, y se llevaron a locales, con el objetivo de poder encapsular las rutinas de lectura de datos de entrada. Anteriormente esto se hacía de forma secuencial, lo cual imponía que cambios en la forma de entrada o la estructura de datos implicaban cambios en gran cantidad de segmentos de código.
- Creación de un objeto para cada archivo de entrada. Con el objetivo de poder separar la carga para los tres archivos de entrada y poder aprovechar todas las ventajas del paradigma orientado a objetos. Así, por ejemplo las modificaciones en una forma de entradas no afectadas a las otras.
- Implementación de una base de datos. En el software base, los datos de los pacientes objeto de estudio, se guardan en un archivo de texto, acarreado problemas de inconsistencia, dificultad de uso y actualización, entre otras. Se procedió a incorporar un sistema de base de datos, para aprovechar al máximo estos sistemas. Esto sienta las bases para futuros trabajos de análisis, dado que se podrán reutilizar los datos de manera sencilla en una estructuración integrada.

- Incorporación de la biblioteca de funciones a un sistema web. Esto incrementa la accesibilidad del sistema. Efectivamente, al colocar esta biblioteca de análisis biomecánico, en un entorno web, los usuarios podrán tener acceso al uso de esta valiosa herramienta, sin complicaciones de instalación de software extra o con características propietarias. Cabe destacar en este punto que los resultados son entregados en un archivo comprimido para que el experto puede tenerlos disponibles en cada momento o transpórtalos a donde se requiera su uso.

### F. Probar el correcto funcionamiento de la interfaz de usuario para garantizar que el producto obtenido, es funcional y cumple cabalmente con los requisitos exigidos

Una vez realizada la interfaz web y la biblioteca C/C++, se procedió a su integración mediante la sentencia "system" de PHP, la cual permite ejecutar sentencias de la Shell del sistema operativo. Esto facilitó la verificación de la correcta funcionalidad del sistema web, asegurando que fuese fácil e intuitivo de usar, así como tener un producto de calidad.

### III. RESULTADOS Y APLICACIONES

Se creó una herramienta, amigable y fácil de usar, que encontrándose en la web, permite al investigador de biomecánica realizar procesos de visualización del C3D,

así como la realización de cálculos para dar soporte a los estudios que se encuentre realizando.

Dentro de los usos que pudiera tener la herramienta creada, la visualización del C3D, que es de por sí un importante aporte, permite recrear gráficamente los movimientos, objetos del estudio, que pudiera realizar un atleta mientras entrena o un paciente con dificultad motora, una ama de casa mientras carga algún objeto, entre otros.

Las aplicaciones de esta herramienta se pueden catalogar en tres grupos:

- Ergonómicos: si deseamos hacer el estudio sobre una ama de casa que debe cargar ciertos objetos, podremos, observar como el objeto pudiera inclinar a la persona en una u otra dirección, al hacer los cálculos se tendrá un análisis cuantitativo de este estudio y permitirá crear objetos más anatómicos o que faciliten su manejo.
- Deportivos: al analizar el movimiento de un atleta, visualmente se podrá fijar donde está realizando mal el ejercicio pero cuantitativamente, se tendrá indicadores que permitan ver el grado de rotación de las partes de su cuerpos, así como si el movimiento se efectúa dentro de los indicadores establecidos.
- Médicos: al utilizar esta herramienta como apoyo al diagnóstico médico, se puede cuantificar si un determinado paciente, sufre alguna anomalía motora. Al observar los indicadores, el experto podrá establecer fijar el tipo de patología.

Al tener toda la información requerida en una base de datos, no se hará necesario que el movimiento se repita constantemente sino que bastará con consultarlo al sistema, también se puede fijar un ciclo de evoluciones de los indicadores.

#### Casos de prueba

Con el objetivo de evaluar la funcionalidad del sistema, se estudiaron un conjunto de datos de pacientes con problemas de lumbalgia. Estos datos fueron recopilados de otro trabajo en la clínica Rougeville, Francia sobre un grupo de registros de 50 casos recopilados 2011.

La lumbalgia se define como aquel dolor común de duración variable, circunscrito a la parte baja de la espalda o zona lumbar. Esta dolencia está perjudicando cada vez a mayor número de personas, afectando su calidad de vida, sin importar edad ni sexo. Las causas que originan esta dolencia se desconocen, los estudios arrojan que pueden provenir desde estados emocionales hasta físicos, y que un tratamiento general no puede aplicarse, sino de manera individual para cada paciente. En el diagnóstico y tratamiento de esta patología intervienen diferentes especialistas como traumatólogos, neurocirujanos, entrenadores de educación física, radiólogos, psicólogos, entre otros, que generan una cierta cantidad de información útil referente al caso clínico. De los datos derivados del análisis de movimiento se han empleado técnicas de análisis y visualización con

FREEMotionlibs Las evaluaciones funcionales y de usuarios realizadas muestran resultados satisfactorios, siendo posible observar los diferentes movimientos que realizaban los pacientes y obtener de estos indicadores que facilitan la toma de decisiones.

#### IV. CONCLUSIONES

1. El estudio de biomecánica es de vital importancia en algunas áreas de la ciencia. El producto implementado en el presente trabajo constituye un notable aporte, ya que a pesar de la existencia de otros desarrollos en esta área, este trabajo es pionero al llevarlo a un entorno web.
2. El uso de WebGL, JavaScript 3 y HTML5, representa un adelanto en la graficación tridimensional del archivo C3D dentro del navegador web, unido a una representación en XML del C3D. Esto es un aporte significativo dentro de la biomecánica ya que en esta investigación se establecen los parámetros básicos.
3. Al conocimiento de los autores no se evidenció alguna otra investigación que estableciera como crear un archivo XML para la representación del archivo C3D.
4. La integración de los lenguajes C/C++, JS, PHP, WebGL y XML en una sola aplicación representa un antecedente para futuros desarrollos, ya que el hecho que una aplicación esté desarrollada en varios lenguajes, no es reciente, pero el uso de lenguajes tan diversos obliga a pensar en cómo mejorar la comunicación entre los lenguajes y/o a la creación de estándares que permitan dicha comunicación.
5. Se ha alcanzado una aplicación que permite tener acceso multiusuario que ofrece la misma funcionalidad que su versión standalone, pero ofreciendo las ventajas implícitas de estar en la web, tener respaldo en una base de datos, tener una forma clara y legible de llevar la marcha del trabajo lo cual permite añadir más funcionalidades o suprimir alguna sin afectar el resto del código.
6. Un gran contribución de esta investigación está en el hecho que sienta un precedente en cómo llevar un software totalmente hecho en C/C++, a una versión web, hacer un análisis de la rutinas que generan la interfaz de usuario, de las que realizan cálculos, las que generan resultados y las que hacen la integración de todo el sistema, y llevar cada rutina a un lenguaje distinto que se adaptara a los requerimientos de un sistema web, así como mantener su funcionalidad al mismo tiempo que se agregaban nuevas funciones y aumentar la capacidad de interacción con el usuario.
7. En futuros trabajos, se podrá realizar minerías de datos, para comprender patrones de diagnósticos de los datos recopilados, también se pueden explorar otras formas de procesamiento como por ejemplo usando javascript para pasar carga del servidor a la máquina del cliente u otras formas de interactuar con el usuario.

## V. REFERENCIAS

1. Aggarwal J. K. and Cai Q., Human motion analysis: Computer Vision and Image Understanding Vol. 73, No. 3, March, pp. 428–440, 1999.
2. Villarroya A. (1996), “Conceptos Básicos En Biomecánica”, Escuela Universitaria de Ciencias de la Salud. Zaragoza, España. VOL. IV, N° 7, paginas 117-121.
3. Vaughan C., Davis B., O’ Connor J. (1999), “Dynamics of Human Gait (2nd edition)”, Kiboho Publishers, Cape Town, South Africa. ISBN: 0-620-23560-8.
4. Rodriguez, P. (23 de Enero de 2009). Blog Archivador. Recuperado el 21 de Noviembre de 2011, de <http://www.elergonomista.com/aitor05.htm>
5. Sitio Web oficial de VICON Standard (s.f), Recuperado el 5 de diciembre de 2012, de <http://www.vicon.com/standard/>
6. Como se encuentra hecho el archivo C3D y utilidad (s.f.) Recuperado el 02 de Noviembre de 2012, de [http://www.c3d.org/pdf/c3dformat\\_ug.pdf](http://www.c3d.org/pdf/c3dformat_ug.pdf).
7. Barbier, F. (2005) “Proposition d’une méthodologie itérative conduisant à l’élaboration desystèmes d’aide à la prise de décision dans le domaine de l’analyse du mouvement ”, Habilitación a la dirección de investigación, Faculté des Sciences et des Métiers du Sport Université de Valenciennes, Valenciennes, France.
8. Barre, A. (3 de marzo de 2012). Blog Archivador. Recuperado el 15 de Noviembre de 2012, de <https://code.google.com/p/b-tk/>
9. Sitio Web oficial de visual3D (s.f), Recuperado el 5 de diciembre de 2012, de <http://www.c-motion.com/support/education/>
10. Rocha, D. Freemotionlibs: una biblioteca en C/C++ para el estudio del movimiento humano. Trabajo Especial de Grado, Licenciatura en Computación, Facultad de Ciencias y Tecnología, Universidad de Carabobo, Venezuela, Enero 2013.
11. Hall, S. J. (1995) Basis Biomechanics. 2° edition, Mosby-Year Book, Inc. St. Louis. USA.
12. Hamill, J. and Knutzen, K. M. (1995) BiomechanicalBasis of Human Movement 1° edition. William & Wilkins. Human Kinetics. USA.
13. Lander, J. Working with Motion Capture File FormatsGame Developer (1998).
14. Larrazabal, G. (2003) UCSparseLib: User’s Manual. Alibrary to solve dense end sparse linear system. University of Carabobo, Sciences and technology faculty, department of computer sciences.
15. WINTER D. (2005) “Biomechanics and motor control of human movement”, 3era Edición, Editorial John Wiley and Sons, New Jersey, USA.

## RECONOCIMIENTOS

Este trabajo ha sido parcialmente financiado por el CDCH de la Universidad de Carabobo bajo el proyecto “A CAD development for the Clinical Gait Analysis using data mining”, 2013-2015.