

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA MECÁNICA**

### **ADAPTACIÓN DE UN SISTEMA DE CAPTURA DE MOVIMIENTO PARA EL LABORATORIO DE BIOMECÁNICA DE LA FACULTAD DE INGENIERÍA MECÁNICA DE LA ESCUELA POLITÉCNICA NACIONAL**

#### **TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN DISEÑO Y SIMULACIÓN**

**FREIRE BONILLA BYRON ANDRÉS**

byron.freire@epn.edu.ec


**DIRECTOR: Ing. CEVALLOS BARRAGÁN CARLOS ENRIQUE, Ph.D.**

carlos.cevallos02@epn.edu.ec

**Quito, enero 2023**

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por **BYRON ANDRÉS FREIRE BONILLA**, bajo mi supervisión.

A handwritten signature in blue ink, written in Arabic script, which reads 'Carlos Cevalles'. The signature is written over a horizontal line.

**Ing. Carlos Cevalles, Ph.D.**

DIRECTOR

## DECLARACIÓN

Yo, **BYRON ANDRÉS FREIRE BONILLA**, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.



Byron Andrés Freire Bonilla

## DEDICATORIA

*A todos*



## CONTENIDO

<b>CERTIFICACIÓN</b> . . . . .	<b>i</b>
<b>DECLARACIÓN</b> . . . . .	<b>ii</b>
<b>DEDICATORIA</b> . . . . .	<b>iii</b>
<b>AGRADECIMIENTO</b> . . . . .	<b>iv</b>
<b>ÍNDICE DE FIGURAS</b> . . . . .	<b>viii</b>
<b>ÍNDICE DE TABLAS</b> . . . . .	<b>x</b>
<b>RESUMEN</b> . . . . .	<b>xi</b>
<b>ABSTRACT</b> . . . . .	<b>xii</b>
<b>INTRODUCCIÓN</b> . . . . .	<b>1</b>
Objetivo General . . . . .	2
Objetivos Específicos . . . . .	2
<b>1. MARCO TEÓRICO</b> . . . . .	<b>3</b>
1.1. Situación actual del país . . . . .	3
1.2. Sistemas de captura de movimiento . . . . .	5
1.2.1. Sistemas no visuales . . . . .	6
1.2.1.1. Sistemas magnéticos . . . . .	6
1.2.1.2. Sistemas mecánicos . . . . .	7
1.2.2. Sistemas visuales . . . . .	8
1.2.2.1. Sistemas con marcadores . . . . .	8
1.2.2.2. Sistemas sin marcadores . . . . .	9
1.2.3. Rastreo robótico . . . . .	10
1.3. Métodos de análisis . . . . .	10
1.4. Fundamentos de la captura de movimiento . . . . .	11
1.4.1. Software a analizarse . . . . .	11
1.4.1.1. OpenMoCap . . . . .	11
1.4.1.2. RTMoCap . . . . .	12

1.4.1.3. OptiTrack Camera SDK . . . . .	13
1.4.1.4. Motion Analyst . . . . .	16
1.4.1.5. FreeMoCap . . . . .	16
1.4.1.6. DLTdv digitizing tool . . . . .	17
1.4.1.7. EasyWand . . . . .	18
1.4.1.8. OBS Studio . . . . .	19
1.4.1.9. OpenCap . . . . .	19
1.4.1.10. FFmpeg . . . . .	20
<b>2. METODOLOGÍA . . . . .</b>	<b>21</b>
2.1. Instrumentación . . . . .	21
2.1.1. Computadores . . . . .	21
2.1.2. Cámaras . . . . .	21
2.1.2.1. Infrarrojas . . . . .	21
2.1.2.2. Ópticas . . . . .	21
2.1.3. Material adicional . . . . .	22
2.2. Proceso de calibración . . . . .	23
2.2.1. OptiTrack Camera SDK . . . . .	24
2.2.2. Motion Analyst - OBS Studio - FFmpeg . . . . .	25
2.2.2.1. Grabación del gesto . . . . .	25
2.2.3. FreeMoCap . . . . .	26
2.2.4. DLTdv digitizing tool - OBS Studio . . . . .	27
2.2.4.1. Grabación de toma de calibración . . . . .	27
2.2.4.2. Extracción de parámetros de las cámaras . . . . .	27
2.2.5. OpenCap . . . . .	31
2.2.5.1. Instalar la aplicación . . . . .	31
2.2.5.2. Configuración . . . . .	31
2.3. Protocolo para toma de datos . . . . .	35
2.3.1. Motion Analyst - OBS Studio - FFmpeg . . . . .	35
2.3.1.1. Separación de archivos individuales . . . . .	35
2.3.2. FreeMoCap . . . . .	35
2.3.3. DLTdv digitizing tool - OBS Studio . . . . .	36
2.3.3.1. Grabación del gesto . . . . .	36

2.3.4. OpenCap . . . . .	36
2.4. Plan de recolección de datos . . . . .	37
2.4.1. Motion Analyst - OBS Studio - FFmpeg . . . . .	37
2.4.1.1. Análisis . . . . .	37
2.4.2. FreeMoCap . . . . .	37
2.4.3. DLTdv digitizing tool - OBS Studio . . . . .	38
2.4.3.1. Análisis . . . . .	38
2.4.4. OpenCap . . . . .	42
<b>3. RESULTADOS Y DISCUSIÓN . . . . .</b>	<b>44</b>
3.1. Análisis . . . . .	44
3.1.1. Motion Analyst . . . . .	44
3.1.2. FreeMoCap . . . . .	44
3.1.3. DLTdv digitizing tool - OBS Studio . . . . .	45
3.1.4. OpenCap . . . . .	46
3.2. Resultados . . . . .	46
3.3. Discusión y validación . . . . .	49
<b>4. CONCLUSIONES Y RECOMENDACIONES . . . . .</b>	<b>54</b>
4.1. Conclusiones . . . . .	54
4.2. Recomendaciones . . . . .	54
<b>BIBLIOGRAFÍA . . . . .</b>	<b>55</b>
<b>ANEXOS . . . . .</b>	<b>59</b>
I. Listado de software de captura de movimiento . . . . .	59
II. Configuración de OBS Studio . . . . .	62
III. Instalación de FFmpeg . . . . .	69
IV. Análisis con Motion Analyst . . . . .	71
V. Instalación de FreeMoCap . . . . .	74
VI. Instalación de DLTdv8a . . . . .	75
VII. Instalación de easyWand . . . . .	78
VII. Scripts de calibración de tablero de ajedrez . . . . .	79
IX. Uso de Camera Calibrator . . . . .	83



## ÍNDICE DE FIGURAS

Figura 1.1. Rastreo de movimiento humano mediante tecnologías de sensores . . .	6
Figura 1.2. OpenMoCap - Repositorio GitHub. . . . .	11
Figura 1.3. RTMoCap. . . . .	12
Figura 1.4. OptiTrack – Camera SDK . . . . .	13
Figura 1.5. Motion Analyst – SimTK. . . . .	16
Figura 1.6. FreeMoCap – Repositorio GitHub. . . . .	16
Figura 1.7. DLTdv digitizing tool. . . . .	17
Figura 1.8. OBS Studio. . . . .	19
Figura 1.9. FFmpeg. . . . .	20
Figura 2.1. Interfaz gráfica – OptiTrack Camera SDK. . . . .	24
Figura 2.2. FreeMoCap – Verificación de cámaras disponibles. . . . .	26
Figura 2.3. FreeMoCap – Calibración de cámaras disponibles. . . . .	26
Figura 2.4. Código para insertar nombre del archivo de video. . . . .	28
Figura 2.5. Nombre de archivos con datos del video. . . . .	28
Figura 2.6. Código para insertar nombre de los archivos de video. . . . .	28
Figura 2.7. Código para insertar valores de los archivos de video. . . . .	29
Figura 2.8. Interfaz gráfica EasyWand. . . . .	29
Figura 2.9. Ingreso de parámetros de fondo. . . . .	30
Figura 2.10. Ingreso de parámetros de cámaras. . . . .	30
Figura 2.11. Pantalla de ingreso - OpenCap. . . . .	32
Figura 2.12. Configuración de cámaras – OpenCap. . . . .	32
Figura 2.13. Calibración de cartilla – OpenCap. . . . .	34
Figura 2.14. Verificación de calibración de cartilla – OpenCap. . . . .	34
Figura 2.15. FreeMoCap – Verificación de cartilla de calibración. . . . .	35
Figura 2.16. Ingreso de datos y toma de pose neutral - OpenCap . . . . .	37
Figura 2.17. FreeMoCap – Inicio de grabación. . . . .	38
Figura 2.18. Interfaz de DLTdv8. . . . .	38
Figura 2.19. Menú de selección de número de videos. . . . .	39
Figura 2.20. Calibración con DLT – DLTdv8. . . . .	39
Figura 2.21. Organización de interfaz – DLTdv8. . . . .	39

Figura 2.22. Inclusión de puntos de rastreo – DLTdv8. . . . .	40
Figura 2.23. Configuración de rastreo – DLTdv8. . . . .	40
Figura 2.24. Exportación de puntos de rastreo – DLTdv8. . . . .	41
Figura 2.25. Generación de archivo de rastreo – DLTdv8. . . . .	41
Figura 2.26. Selección de tipo de formato – DLTdv8. . . . .	41
Figura 2.27. Creación de intervalos de confianza – DLTdv8. . . . .	41
Figura 2.28. Pantalla de inicio de grabación – OpenCap. . . . .	42
Figura 2.29. Generación de simulación – OpenCap. . . . .	43
Figura 3.1. Proceso de análisis para Motion Analyst – OBS Studio – FFmpeg. . . . .	44
Figura 3.2. Proceso de análisis para FreeMoCap. . . . .	44
Figura 3.3. Proceso de análisis para DLTdv – OBS Studio. . . . .	45
Figura 3.4. Proceso de análisis para OpenCap. . . . .	46
Figura 3.5. Etapas de trabajo. . . . .	49
Figura 3.6. Grabación de gestos – OpenCap. . . . .	51
Figura 3.7. Carga de modelo – OpenSim . . . . .	52
Figura 3.8. Carga de movimiento – OpenSim. . . . .	52
Figura 3.9. Elongación músculo recto femoral – OpenSim. . . . .	53
Figura 3.10. Análisis – OpenCap. . . . .	53

## ÍNDICE DE TABLAS

Tabla 1.1. Tecnologías y técnicas de MoCap. . . . .	3
Tabla 1.2. Comparación de desempeño – Rastreo de movimiento. . . . .	6
Tabla 1.3. Ventajas y desventajas – Sistemas magnéticos. . . . .	7
Tabla 1.4. Ventajas y desventajas – Sistemas mecánicos. . . . .	8
Tabla 1.5. Ventajas y desventajas – Sistemas con marcadores. . . . .	9
Tabla 1.6. Propiedades de OptiTrack Camera SDK. . . . .	13
Tabla 2.1. Características de los equipos empleados. . . . .	21
Tabla 2.2. Características cámaras infrarrojas. . . . .	21
Tabla 2.3. Características cámaras ópticas. . . . .	22
Tabla 2.4. Material adicional. . . . .	22
Tabla 3.1. Detalles OptiTrack Camera SDK. . . . .	46
Tabla 3.2. Detalles Motion Analyst – OBS Studio – FFmpeg. . . . .	47
Tabla 3.3. Detalles FreeMoCap. . . . .	47
Tabla 3.4. Detalles DLTdv digitizing tool – OBS Studio. . . . .	47
Tabla 3.5. Detalles OpenCap. . . . .	47
Tabla 3.6. Capacidades de los programas utilizados para laboratorio. . . . .	48
Tabla 3.7. Comparación de capacidades de los programas utilizados. . . . .	48

## RESUMEN

Se ha propuesto definir una adaptación de un sistema de captura de movimiento para el Laboratorio de Biomecánica de la Facultad de Ingeniería Mecánica con uso de software libre.

Se evaluó un total de 10 programas y la combinación de 3 de estos como apoyo para 2 de los mismos. Se presenta tres etapas para uso, con un proceso de calibración, protocolo para toma de datos y un plan de recolección de datos. Se limitó el análisis a cuatro programas debido a que 3, presentaron problemas de compilación.

Las capturas de movimiento se realizaron con cámaras web caseras, fijadas en soportes adecuados con una distancia hacia el sujeto de prueba de no más de 5 metros. Se probó el software en tres equipos (dos fijos y uno portátil). El análisis comparativo final recomienda el uso de dos de los programas, uno de los cuales trabajará en combinación con software libre.

**Palabras clave:** Biomecánica, captura de movimiento, software libre

## **ABSTRACT**

It has been proposed to define an adaptation of a motion capture system for the Biomechanics Laboratory of the Faculty of Mechanical Engineering using free Open source software.

A total of 10 programs were evaluated and the combination of 3 of these as support for 2 of them. Three stages are presented for use, with a calibration process, data collection protocol and a data collection plan. The analysis was limited to four programs because three of them presented compilation problems.

Motion captures were performed with homemade webcams, fixed on suitable supports with a distance to the test subject of no more than 5 meters. The software was tested on three computers (two fixed and one portable). The final comparative analysis recommends the use of two of the programs, one of which will work in combination with open source software.

**Keywords:** Biomechanics, motion capture, Opensource software.

# **ADAPTACIÓN DE UN SISTEMA DE CAPTURA DE MOVIMIENTO PARA EL LABORATORIO DE BIOMECAÁNICA DE LA FACULTAD DE INGENIERÍA MECÁNICA DE LA ESCUELA POLITÉCNICA NACIONAL**

## **INTRODUCCIÓN**

La captura y el análisis automático del movimiento humano es un área de investigación muy activa debido tanto al número de aplicaciones potenciales como a su complejidad inherente. Esta contiene una serie de problemas difíciles, como inferir la pose y el movimiento de un objeto 3D altamente articulado y auto oclusivo a partir de imágenes. La complejidad presentada, hace que el área de investigación sea un reto desde un punto de vista puramente académico. Considerando el punto de vista de las aplicaciones, los métodos basados en la visión por ordenador suelen ser la única solución no invasiva, lo que los hace muy atractivos (Moeslund, Hilton, y Krüger, 2006).

Los sistemas de captura de movimiento (MoCaps) se utilizan en varias aplicaciones, como el rango de movimiento (ROM) (Gómez-García, Godino-Llorente, y Castellanos-Domínguez, 2014), el control postural (Huang y Brown, 2013), el análisis de la marcha (Wang, 2006) y la aplicación deportiva (Lamoth y van Heuvelen, 2012). Se ha utilizado también para abordar diversas cuestiones relacionadas con la rehabilitación de accidentes cerebrovasculares (Hondori, Khademi, Dodakian, Cramer, y Lopes, 2013); (Pridmore, Hilton, Green, Eastgate, y Cobb, 2004); (Burke, Morrow, McNeill, McDonough, y Charles, 2008).

Dentro de la Facultad de Ingeniería Mecánica de la Escuela Politécnica Nacional, se llevaron a cabo algunos proyectos enfocados en el área de biomecánica y rehabilitación (PIJ-15-08, PIMI-15-04). Además, existen algunas tesis de investigación (Maza y A., 2017) (Moreno Maza, 2017), (Molina y J., 2017) (Zagal Molina, 2017), (Cruz y A., 2018) (Ruiz Cruz, 2018), (Freire, 2019) que contribuyen a los primeros acercamientos para proponer estudios a largo plazo sobre sujetos de prueba y el establecimiento de protocolos de evaluación adecuados que posteriormente puedan ser validados.

El seguimiento del movimiento humano es un requisito común para muchas aplicaciones del mundo real, como videovigilancia (detección de movimiento para localización de

intrusos), juegos (inclusión del movimiento de los jugadores en software para ejecución de gestos que permitan el control de un personaje), aplicaciones culturales y médicas (por ejemplo, creación de escenas que involucren la participación del usuario con entornos interactivos, análisis de marcha, análisis de gestos en el rostro de usuarios). Se puede detectar y rastrear un objeto simple utilizando varias características de imagen, como regiones de color (marcadores), bordes, contornos o texturas.(Wang, Cheng, y Zhao, 2010).

El Laboratorio de Biomecánica de la Facultad de Ingeniería Mecánica, actualmente maneja el procesamiento de captura de movimiento a través del software Kinescan/IBV (Instituto de Biomecánica de Valencia, 2021) el cual ha permitido la ejecución de varias tesis (Zambrano, 2020), (Pérez, 2020), que haciendo uso del software antes mencionado, han contribuido con información de análisis ergonómico con un sistema de cámaras fijas.

Dado este contexto, se propone la adaptación y generación de un sistema de fotogrametría móvil que permita la obtención de datos de marcadores (posiciones).

## **Objetivo General**

Adaptar un sistema de captura de movimiento para el Laboratorio de Biomecánica de la Facultad de Ingeniería Mecánica de la Escuela Politécnica Nacional.

## **Objetivos Específicos**

- Comparar distintos softwares libres de captura de movimiento para seleccionar el más adecuado.
- Generar un proceso de calibración y captura de información para las cámaras.
- Recuperar datos (archivos de texto) que puedan ser usados en análisis biomecánicos.

# 1. MARCO TEÓRICO

## 1.1. Situación actual del país

La biomecánica es un campo interdisciplinar que estudia los modelos, fenómenos y leyes relevantes para el movimiento y el equilibrio de los seres vivos, así como las estructuras mecánicas presentes en los organismos vivos. Gran parte del trabajo que se realiza se centra en el cuerpo humano, la investigación médica o la robótica; se nutre de los conocimientos adquiridos en una serie de ciencias biomédicas y aplica los conocimientos de las disciplinas de la mecánica, la ingeniería, la anatomía, la fisiología., para estudiar el comportamiento del cuerpo humano (Moeslund et al., 2006).

La captura de movimiento (MoCap) según Menolotto, Komaris, Tedesco, O'Flynn, y Walsh (2020): es el proceso de seguimiento y recodificación digital de los movimientos de objetos o seres vivos en el espacio. Las diferentes tecnologías y técnicas para capturar el movimiento, se muestran en la Tabla 1.1.

Tabla 1.1. Tecnologías y técnicas de MoCap.

<b>Tecnología</b>	<b>Descripción</b>
Sistemas basados en cámaras infrarrojas	Se utilizan para triangular la ubicación de cuerpos rígidos retroreflectantes adheridos al objeto o sujeto
Cámaras sensibles a la profundidad	Proyectan luz hacia un objeto, pueden estimar la profundidad en función del tiempo transcurrido desde la emisión de la luz hasta la detección de la luz retro dispersada.
Sistemas basados en sensores inerciales (IMU), campos electromagnéticos y potenciómetros	Siguen los movimientos relativos de estructuras articuladas
Sistemas híbridos	Combinan distintas tecnologías MoCap para mejorar la precisión y reducir las oclusiones de la cámara

Fuente: Menolotto et al. (2020)

La investigación también se ha centrado en el manejo y el procesamiento de conjuntos de datos de alta dimensión con una amplia gama de técnicas de análisis, como el aprendizaje automático (Komaris et al., 2019), filtros de Kalman (Jin, Zhao, Jin, Yu, y Li, 2014), agrupación jerárquica (Komaris et al., 2020) y otras.

Se han realizado reseñas y encuestas disponibles sobre el uso de varios sistemas MoCap en relación con diferentes aplicaciones como análisis en robótica (Field, Stirling, y Naghdy, 2011), análisis de movimiento de los atletas (Fang y Xuesong, 2015), reconocimiento



de acción humana (Chen, Jafari, y Kehtarnavaz, 2017), 2017), movimiento humano, seguimiento y estimación y reconocimiento de poses (Moeslund et al., 2006). Otro estudio ha revisado las tecnologías de seguimiento que se utilizan para capturar el movimiento humano para rehabilitación (Zhou y Hu, 2008).

Se han recopilado 5 trabajos de titulación para presentar un panorama de la investigación en sistemas de captura de movimiento en nuestro país, a continuación, una descripción breve de los mismos.

Un primer trabajo, desarrollado por Moya y Miguel (2018) muestra en su resumen, que describió la técnica de la sentadilla a través de la observación del gesto motor en los fisicoculturistas amateurs del gimnasio de la Pontificia Universidad Católica del Ecuador (PUCE). El tipo de estudio realizado fue descriptivo, transversal, observacional, con una muestra constituida por 17 deportistas que cumplieron con los criterios de inclusión. Para este estudio se realizó un análisis de movimiento mediante video con el programa KINOVEA (v.0.80.15), y se recabaron datos por medio de un cuestionario acerca de características demográficas, de entrenamiento, del dolor y si presentaban o no lesiones actuales.

Dos trabajos realizados en la Universidad Técnica del Norte, (Pozo y N., 2018) y (Anrrango y A., 2019) muestran aplicaciones de Microsoft Kinect, en el primer caso, para desarrollar un sistema de captura de movimiento para asistir el entrenamiento de los miembros del club de taekwondo de la universidad, monitoreando la ejecución de las posturas básicas, utilizando la tecnología de captura de movimiento del sensor Kinect para obtener las coordenadas de las articulaciones del cuerpo humano en el plano. En el segundo caso, se desarrolló una aplicación en un lenguaje de programación de código abierto mostrando un sistema de monitoreo de los diferentes movimientos corporales a través de los cálculos de los ángulos de flexión de las extremidades del cuerpo. La aplicación está basada en el uso del mismo sensor y genera un esqueleto (skeleton), el que será mostrado en pantalla y reproducirá los movimientos realizados por el usuario que se posicione frente al sistema.

En la Universidad de las Américas, el trabajo de García (2020) además del uso del sensor Kinect para la captura de movimiento, utilizó varios métodos digitales para la recreación 3D de las danzantes, entre ellos fotogrametría, modelado y escultura digital.

Finalmente, en (Yansaguano y B., 2020) se describe el diseño e implementación de un sistema electrónico de corrección del gesto deportivo en base a la biomecánica en deportistas de marcha atlética mediante machine learning, deep learning y aprendizaje automático profundo, con bibliotecas y modelos que están dirigidos a este campo y ayudan a la implementación del prototipo de seguimiento de los practicantes de marcha atlética mediante la sensorización y detección.

En general, se aprecia que los trabajos cumplen con el objetivo de hacer una captura de movimiento con el uso de sensores comerciales y analizar el mismo para diferentes aplicaciones, sin embargo, no se evidencia que se disponga de un laboratorio personalizado con programas especializados además de que los sistemas empleados se basan en dispositivos que trabajan en el rango visible sin recurrir a cámaras infrarrojas y sensores IMU; los datos que se informan no muestran que se realicen calibraciones para el uso del software de video.

## **1.2. Sistemas de captura de movimiento**

Existen numerosos sistemas de captura de movimiento (MoCap), todos ellos capaces de realizar un análisis de las posiciones, fuerzas, velocidades e impulsos de los movimientos de un sujeto tomando como referencia las variables obtenidas en determinados rangos de tiempo. De los diferentes sistemas y tecnologías existentes en el mercado, los más conocidos son los sistemas ópticos -que utilizan deflectores o marcadores colocados en puntos estratégicos del cuerpo- y los sistemas inerciales(Estevez-García et al., 2015).

Es esencial estudiar la estructura y las características de los sensores individuales que se emplean para el sistema particular de captura de movimiento. En general, un sistema de seguimiento puede ser no visual, visual (por ejemplo, basado en marcadores y sin marcadores) o una combinación de ambos, como se puede ver en la Figura 1.1

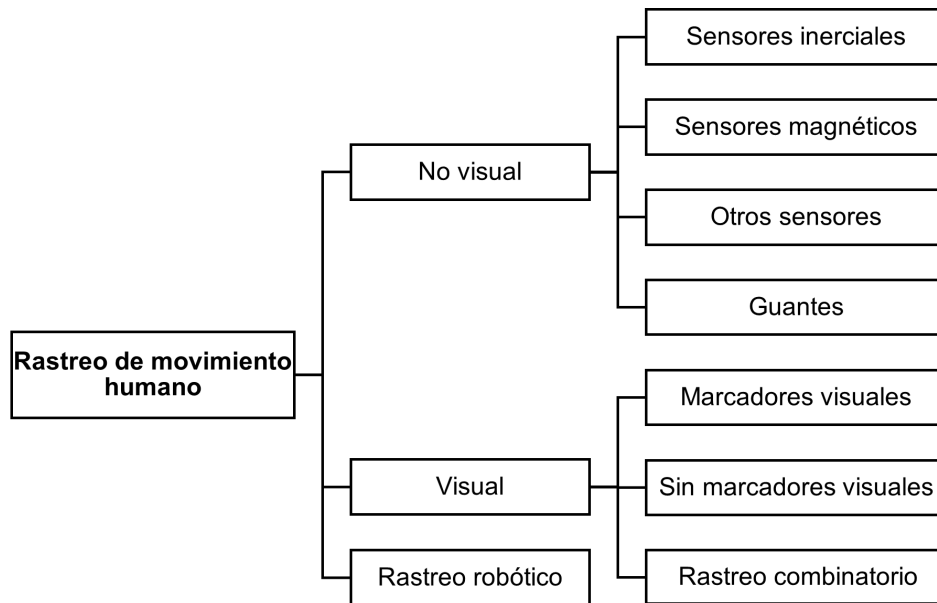


Figura 1.1. Rastreo de movimiento humano mediante tecnologías de sensores  
Fuente: Adaptado de Zhou y Hu (2008)

Según Zhou y Hu (2008), el rendimiento de los sistemas basados en las técnicas de la Figura 1.1 se describe en la Tabla 1.2

Tabla 1.2. Comparación de desempeño – Rastreo de movimiento.

Sistema	Precisión	Compactación	Cómputos	Costo	Inconvenientes
Inercial	Alta	Alta	Eficiente	Bajo	Desviación
Magnético	Media	Alta	Eficiente	Bajo	Materiales ferromagnéticos
Ultrasonido	Media	Baja	Eficiente	Bajo	Oclusión
Guante	Alta	Alta	Eficiente	Medio	Postura parcial
Con marcadores	Alta	Baja	Ineficiente	Medio	Oclusión
Sin marcadores	Alta	Alta	Ineficiente	Bajo	Oclusión
Combinatorio	Alta	Baja	Ineficiente	Alto	Multidisciplinario
Robot	Alta	Baja	Ineficiente	Alto	Movimiento limitado

Fuente: Adaptado de Zhou y Hu (2008)

### 1.2.1. Sistemas no visuales

Los sensores empleados en estos sistemas se adhieren al cuerpo del sujeto para recoger información sobre el movimiento; se suelen clasificar en mecánicos, inerciales, acústicos, de radio, de microondas y magnéticos. Algunos de ellos tienen gran sensibilidad que pueden detectar pequeñas amplitudes, como los movimientos de los dedos de los pies o de las manos.

#### 1.2.1.1. Sistemas magnéticos

Los sistemas MoCap magnéticos (electromagnéticos) se denominan a veces rastreadores magnéticos, estos se derivan de los sensores colocados en el casco de los pilotos de aviones militares para seguir la posición y orientación de la cabeza del piloto para la

pantalla montada en el casco. Con un MoCap magnético, se colocan de 12 a 20 sensores de seguimiento en un sujeto de captura para medir la relación espacial con un transmisor magnético. Los sensores de seguimiento emiten sus traslaciones y orientaciones. (Kitagawa y Windsor, 2008)

Por lo tanto, no es necesario el post procesamiento para calcular las rotaciones. Este hecho permite utilizar los sistemas magnéticos para aplicaciones en tiempo real. Los sensores de seguimiento no son obstruidos por los sujetos de captura ni por los accesorios de materiales no metálicos, lo que supone una ventaja sobre los sistemas ópticos. Sin embargo, son propensos a las interferencias magnéticas y eléctricas causadas por los objetos metálicos y la electrónica de los entornos. Las interferencias pueden dar lugar a una salida distorsionada. Las estructuras de los edificios con metales de alta conductividad no son adecuadas como espacios de captura para los sistemas magnéticos. El cableado y las baterías de los sensores de seguimiento pueden limitar los movimientos de los sujetos de captura. Además, las baterías de los sensores de seguimiento deben recargarse cada pocas horas. En la Tabla 1.3 se presentan las ventajas y desventajas de estos sistemas.

Tabla 1.3. Ventajas y desventajas – Sistemas magnéticos.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>- Los datos ópticos son precisos.</li> <li>- La posición y la orientación están disponibles sin post procesamiento.</li> <li>- La retroalimentación en tiempo real permite aplicaciones instantáneas.</li> <li>- Los sensores de seguimiento no son ocultados por objetos no metálicos.</li> <li>- Se pueden capturar simultáneamente varios sujetos con múltiples configuraciones.</li> <li>- Los sistemas magnéticos son menos costosos que los sistemas ópticos.</li> </ul>	<ul style="list-style-type: none"> <li>- Los sensores de seguimiento son propensos a las interferencias magnéticas y eléctricas.</li> <li>- El cableado y las baterías de los sensores de seguimiento pueden limitar los movimientos de los sujetos de captura.</li> <li>- Los sensores magnéticos tienen una frecuencia de muestreo menor que la mayoría de los sistemas ópticos.</li> <li>- Los datos magnéticos tienden a tener ruido.</li> <li>- Las configuraciones de los sensores de seguimiento son difíciles de cambiar.</li> <li>- Los volúmenes de captura de los sistemas magnéticos son normalmente más pequeños que los sistemas ópticos.</li> </ul>

Fuente: Adaptado de (Kitagawa y Windsor, 2008)

### 1.2.1.2. Sistemas mecánicos

Los sistemas MoCap mecánicos (exoesqueléticos) miden directamente los ángulos de las articulaciones de un sujeto que lleva un dispositivo articulado que consta de varillas rectas y potenciómetros. Las barras rectas están unidas a potenciómetros en las articulaciones

del cuerpo, diseñadas para medir los ángulos de las articulaciones a medida que el sujeto se mueve. El dispositivo tiene el aspecto de un exoesqueleto. Otros tipos de sistemas mecánicos son los guantes (Kitagawa y Windsor, 2008).

Los sistemas mecánicos inalámbricos proporcionan grandes volúmenes de captura. Una desventaja notable de los sistemas mecánicos es que no miden muy bien la traslación global. Hacen un mal trabajo cuando los pies se levantan del suelo. Si un sujeto salta, los datos normalmente no seguirán el salto y los datos se quedarán en el suelo. Si un personaje sube las escaleras, los datos nunca se elevarán en el aire, sino que parecerá que camina en su sitio. A menudo se añaden sensores magnéticos a los sistemas mecánicos para corregir este problema. Se resume las ventajas y desventajas de estos sistemas en la Tabla 1.4.

Tabla 1.4. Ventajas y desventajas – Sistemas mecánicos.

<b>Ventajas</b>	<b>Desventajas</b>
<ul style="list-style-type: none"> <li>- Los datos ópticos son precisos.</li> <li>- Tiempo real.</li> <li>- Relativamente barato.</li> <li>- Sin oclusión.</li> <li>- Sin interferencias magnéticas o eléctricas.</li> <li>- Altamente portátil.</li> <li>- Gran rango de captura.</li> </ul>	<ul style="list-style-type: none"> <li>- No hay traducciones globales.</li> <li>- Restricciones al movimiento del sujeto de captura.</li> <li>- Rompible.</li> <li>- Configuración fija de los sensores.</li> <li>- Baja frecuencia de muestreo.</li> </ul>

Fuente: Adaptado de (Kitagawa y Windsor, 2008)

## **1.2.2. Sistemas visuales**

Los sensores ópticos se aplican normalmente para mejorar la precisión en la estimación de la posición. Los sistemas de seguimiento pueden clasificarse como de marcadores visuales o sin ellos, según si los estos están unidos a las partes del cuerpo o no.

### **1.2.2.1. Sistemas con marcadores**

El seguimiento basado en marcadores visuales es una técnica en la que las cámaras se aplican para seguir los movimientos humanos, con marcadores colocados en el cuerpo humano.

Un sistema óptico típico consta de 4 a 32 cámaras y un ordenador que controla las cámaras. En la mayoría de los sistemas ópticos, los sujetos de captura llevan marcadores, que pueden ser reflectantes (pasivos) o emisores de luz (activos). Los marcadores pasivos

están hechos de materiales reflectantes y sus formas son esféricas, semiesféricas o circulares. Las formas y tamaños de los marcadores dependen de las resoluciones de la cámara y de los sujetos de captura (por ejemplo, se utilizan marcadores más pequeños para las capturas faciales y de manos). Los marcadores pasivos se adhieren directamente a la piel del sujeto de captura o se fijan con velcro a un traje de MoCap. En la Tabla 1.4 se aprecian las ventajas y desventajas de estos sistemas.

Tabla 1.5. Ventajas y desventajas – Sistemas con marcadores.

<b>Ventajas</b>	<b>Desventajas</b>
<ul style="list-style-type: none"> <li>- Los datos ópticos son precisos.</li> <li>- La tasa de captura es alta.</li> <li>- Se pueden capturar múltiples sujetos simultáneamente.</li> <li>- Se puede utilizar un gran número de marcadores.</li> <li>- Las configuraciones de los marcadores pueden cambiarse fácilmente, en función de los objetivos del proyecto.</li> <li>- Los sujetos de captura de los sistemas ópticos pueden moverse libremente en un volumen de captura.</li> <li>- El volumen de captura puede ser mayor que el de la mayoría de los otros sistemas.</li> <li>- Se pueden generar datos esqueléticos.</li> </ul>	<ul style="list-style-type: none"> <li>- Se requiere un amplio post-procesamiento.</li> <li>- Datos de rotación deben calcularse a partir de los datos de posición en el post procesamiento.</li> <li>- Los marcadores pueden ser ocultados por los sujetos de captura o los accesorios, lo que resulta en la pérdida de datos.</li> <li>- La iluminación debe ser controlada para la mayoría de los sistemas ópticos, especialmente los pasivos.</li> <li>- La información visual en tiempo real se limita a las figuras de palo.</li> <li>- El hardware suele ser más caro que otros tipos de equipos de MoCap.</li> </ul>

Fuente: Adaptado de (Kitagawa y Windsor, 2008)

Los sistemas de seguimiento basados en marcadores visuales, por ejemplo, VICON u OptiTrack, se utilizan a menudo como “estándar de oro” en el análisis del movimiento humano debido a su precisa información de posición (los errores se sitúan en torno a 1 [mm]). Esta característica de precisión motiva de forma optimista las aplicaciones populares de los sistemas de sistemas de seguimiento basados en marcadores visuales en medicina (Zhou y Hu, 2008).

### **1.2.2.2. Sistemas sin marcadores**

Los sistemas de seguimiento visual sin marcadores sólo utilizan sensores ópticos para medir los movimientos del cuerpo humano. Esta aplicación está motivada por los defectos de usar sistemas basados en marcadores visuales, de forma general:

1. Identificación de puntos de referencia óseos estándar.
2. Tejido blando que recubre los puntos de referencia óseos pueden moverse, dando lugar a datos ruidosos.

3. El propio marcador puede tambalearse debido a su propia inercia.
4. Los marcadores pueden incluso quedar completamente desviados o desubicados de la posición marcada al inicio.

Una pequeña desventaja es que esta técnica requiere un cálculo intensivo para llevar a cabo la localización y la reducción de errores en 3D, además de la minimización de la latencia de los datos.

### **1.2.3. Rastreo robótico**

Los sistemas de seguimiento asistidos por robots, son un subconjunto de robots terapéuticos que presentan plataformas valiosas para la rehabilitación neurológica de miembros humanos tras un accidente cerebrovascular (Hillman, 2003) (Speich y Rosen, 2004). La representación de la posición/orientación de las extremidades está incluida y se requiere necesariamente para guiar su movimiento. Estos sistemas incorporan tecnologías de sensores individuales para llevar a cabo estrategias de “sentido-medida-retroalimentación” (Zhou y Hu, 2008).

## **1.3. Métodos de análisis**

Se instalará las diferentes alternativas de software libre de captura de movimiento en un computador junto con el SDK OptiTrack® para evaluar las capacidades de personalización y de captura de datos disponibles con el fin de seleccionar la más adecuada para el laboratorio.

Una vez definido el software adecuado, se definirá el proceso experimental de adaptación del mismo para el sistema de video-análisis en el laboratorio de análisis de movimiento de la Facultad de Ingeniería Mecánica de la Escuela Politécnica Nacional. Este proceso consiste en: generación de calibración en un entorno cerrado (laboratorio) del equipo utilizando marcadores reflectivos o posiciones de referencia adecuadas; definición del volumen de captura en el cual se analiza el fenómeno (área de estudio) y finalmente, generación de datos en función de los parámetros anteriores.

Concluido el proceso anterior, se explorará la posibilidad de utilizar el sistema de captura de movimiento (cámaras y software) en exteriores, con la finalidad de dotar movilidad al mismo para análisis de fenómenos biomecánicos.

## 1.4. Fundamentos de la captura de movimiento

Según (Estevez-García et al., 2015), la captura de movimiento, o simplemente MoCap, es una tecnología que permite capturar movimientos en un entorno digital para su posterior análisis. Esta tecnología se basa en una técnica en la que las propiedades geométricas de los objetos y/o seres vivos se establecen a partir de la imagen capturada. En los últimos años, la captura de movimiento se ha implantado ampliamente en muchos sectores, pero llamó la atención de la mayoría de la gente a raíz de su creciente presencia en la industria cinematográfica. Además de esta industria, los sistemas de captura de movimiento se utilizan ahora ampliamente en otros ámbitos, por ejemplo, en los laboratorios más avanzados de I+D médica y, en particular, en el campo de la biomecánica (Corazza, Gambaretto, Munderman, y Andriacchi, 2010).

### 1.4.1. Software a analizarse

#### 1.4.1.1. OpenMoCap

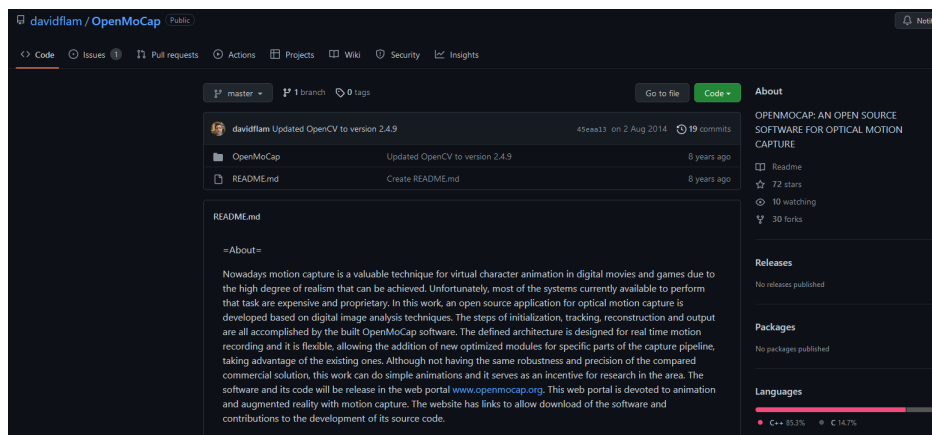


Figura 1.2. OpenMoCap - Repositorio GitHub.  
Fuente: (Flam, 2022)

OpenMoCap desarrolla una aplicación de código abierto para la captura óptica de movimiento basada en técnicas de análisis de imágenes digitales. Los pasos de inicialización, seguimiento, reconstrucción y salida son realizados por el software desarrollado. La arquitectura definida está diseñada para la grabación de movimiento en tiempo real y es flexible, permitiendo la adición de nuevos módulos optimizados para partes específicas de captura, aprovechando los ya existentes. Aunque no tiene la misma robustez y precisión de una solución comercial, OpenMoCap puede realizar animaciones sencillas y sirve de incentivo para la investigación en el área. El software y su código



se publicaron en el portal web <https://github.com/davidflam/OpenMoCap>. Este portal web está dedicado a la animación y la realidad aumentada con captura de movimiento. El sitio web tiene enlaces que permiten descargar el software y contribuir al desarrollo de su código fuente.

### 1.4.1.2. RTMoCap

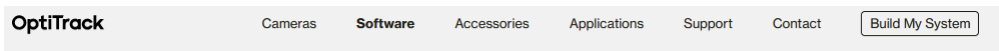
#### RTMoCap

<b>Menu</b>	<b>RealTime Motion Capture Toolbox (RTMocap)</b>
<b>RealTime Motion Capture Toolbox (RTMocap)</b>	<i>(This site is still under construction and most of its content will change in the next weeks. Please send your inquiries to <a href="mailto:rtmocaptoolbox@gmail.com">rtmocaptoolbox@gmail.com</a>)</i>
<b>General Principles</b>	
<b>Install the Toolbox</b>	
<b>Hands-on Tutorial : offline processing</b>	
<b>Hands-on Tutorial : online experiment</b>	
<b>Four Key examples using RTMocap</b>	
Example 1 : Reading motor intention in a reach to grasp task	
Example 2 : Introducing timing uncertainty within Sequential Motor Actions	
Example 3 : Reinforcement learning	
Example 4 : Introducing the social aspects of motor control	
	<p>We present here a toolbox for real-time motion capture of biological movements that runs under the cross platform MATLAB environment (The Mathworks, Inc.). Available functions include (1) the setting of reference positions, areas and trajectories of interest, and (2) the detection of events to use as trigger of external reinforcers (e.g., light, sound, odours).</p> <p>Through fast on-line communication between hardware controller and RTMocap, automatic trial selection is possible in function of either pre-set or adaptive criterion. Rapid pre-processing of signals is also proposed, which includes artefact rejection, filtering, epoch selection, and averaging.</p> <p>4 typical key examples are developed (1) to propose clear understanding of the importance of real-time control for 3D motion in cognitive sciences and (2) to present users with simple lines of codes that can be set as starting points for customized experiments using simple MATLAB syntax. RTMocap is freely available (<a href="http://sites.google.com/site/RTMocap/">http://sites.google.com/site/RTMocap/</a>) under the GNU public license for non-commercial use and open source development, together with sample data and extensive documentation.</p> <p>Download links : <a href="http://github.com/DLewkowicz/RTMocap">http://github.com/DLewkowicz/RTMocap</a> (download .zip file)</p>

Figura 1.3. RTMoCap.  
Fuente: (Lewkowicz, 2022)

RTMoCap es una caja de herramientas para la captura de movimientos biológicos en tiempo real que se ejecuta bajo el entorno multiplataforma MATLAB. Las funciones disponibles incluyen (1) el establecimiento de posiciones de referencia, áreas y trayectorias de interés, y (2) la detección de eventos para utilizarlos como disparadores de reforzadores externos (por ejemplo, luz, sonido, olores). Gracias a la rápida comunicación en línea entre el controlador de hardware y el RTMoCap, es posible la selección automática de ensayos en función de un criterio preestablecido o adaptativo. También se propone un preprocesamiento rápido de las señales, que incluye el rechazo de artefactos, el filtrado, la selección de épocas y el promediado. RTMoCap está disponible gratuitamente bajo la licencia pública GNU para uso no comercial y desarrollo de código abierto, junto con datos de muestra y una extensa documentación (RTMoCap, 2022)

### 1.4.1.3. OptiTrack Camera SDK



## Camera SDK



Integrate OptiTrack cameras with tracking and vision applications. Full camera control and sample application source code included.

Figura 1.4. OptiTrack – Camera SDK  
Fuente: (OptiTrack, 2022)

Camera SDK es un paquete de desarrollo gratuito que permite a los usuarios controlar y realizar capturas desde las cámaras NaturalPoint para crear aplicaciones de seguimiento y visión personalizadas. El SDK incluye el control total de la cámara, el acceso a los datos de los objetos en 2D, la sincronización de la cámara y el seguimiento vectorial integral: un completo surtido de herramientas para crear aplicaciones rentables y potentes desde cero o integrarlas con el software existente. Todos los modelos actuales de cámaras OptiTrack, TrackIR y SmartNav son compatibles, además de muchos modelos descatalogados. El OptiHub y la Llave de Hardware también son compatibles para la sincronización y el desarrollo de licencias (OptiTrack, 2022).

Algunas de las propiedades de este SDK que indica el fabricante se muestran en la Tabla 1.6

Tabla 1.6. Propiedades de OptiTrack Camera SDK.

Propiedades	Descripción
Integración con C++	Una interfaz C++ para el desarrollo sustituye a la interfaz COM del anterior SDK de OptiTrack para una mayor facilidad de integración con una variedad de lenguajes y marcos de software.

Propiedades	Descripción
Control de la cámara	<p data-bbox="675 293 1262 488">Amplio acceso a las funciones, métodos, devoluciones de llamada y propiedades de la cámara para un control completo de su sistema de visión, incluyendo:</p> <ul data-bbox="675 510 1270 1245" style="list-style-type: none"> <li data-bbox="675 510 1270 600">- Modo de vídeo (objeto, segmento, precisión, escala de grises sin procesar, MJPEG).</li> <li data-bbox="675 622 1206 712">- Velocidad de fotogramas con control de decimación de fotogramas.</li> <li data-bbox="675 734 1206 824">- Exposición con control automático de la exposición.</li> <li data-bbox="675 846 1145 880">- Control automático de la ganancia.</li> <li data-bbox="675 902 1114 936">- Enmascaramiento por hardware.</li> <li data-bbox="675 958 1098 992">- Ventana de tamaño de imagen.</li> <li data-bbox="675 1014 794 1048">- Umbral.</li> <li data-bbox="675 1070 850 1104">- Iluminación.</li> <li data-bbox="675 1126 978 1160">- Conmutador de filtros.</li> <li data-bbox="675 1182 994 1216">- Modo de alta potencia .</li> <li data-bbox="675 1238 1042 1272">- Control del LED de estado.</li> </ul>
Datos de objetos	<p data-bbox="675 1267 1270 1413">Se pueden extraer los datos de los objetos 2D de hasta 512 objetos en la vista de la cámara para su seguimiento y filtrado.</p> <p data-bbox="675 1435 1110 1469">La información disponible incluye:</p> <ul data-bbox="675 1491 1302 1789" style="list-style-type: none"> <li data-bbox="675 1491 1254 1525">- Ubicación (coordenadas X, Y ponderadas).</li> <li data-bbox="675 1547 759 1581">- Área.</li> <li data-bbox="675 1603 895 1637">- Anchura/Altura.</li> <li data-bbox="675 1659 831 1693">- Redondez.</li> <li data-bbox="675 1715 1302 1789">- Posibilidad de eliminar los efectos de distorsión de la lente.</li> </ul>

Propiedades	Descripción
Seguimiento de vectores (6 grados de libertad)	El SDK de la cámara incluye el seguimiento vectorial básico tanto para el TrackClip como para el TrackClip PRO, proporcionando una salida 6DoF. Vector es una tecnología propia desarrollada por NaturalPoint para el seguimiento 6DoF utilizando tres marcadores y una sola cámara.
Sincronización multicámara	Se pueden sincronizar varias cámaras para que disparen sus obturadores y expongan los fotogramas al mismo tiempo. En lugar de obtener fotogramas de cámaras individuales, se pueden crear grupos para obtener fotogramas sincronizados y confirmar que las cámaras del grupo están correctamente sincronizadas.
Aplicaciones de ejemplo con código fuente	Se incluye el código fuente y la documentación de las aplicaciones de ejemplo para demostrar las mejores prácticas y proporcionar puntos de partida para utilizar el SDK. Las aplicaciones de ejemplo incluyen: <ul style="list-style-type: none"> <li>- Lista de cámaras.</li> <li>- Sincronización de fotogramas.</li> <li>- Procesamiento de imágenes.</li> <li>- Visor MJPEG.</li> <li>- Modo de segmentos.</li> <li>- Seguimiento de vectores.</li> </ul>

Fuente:(OptiTrack, 2022)

#### 1.4.1.4. Motion Analyst

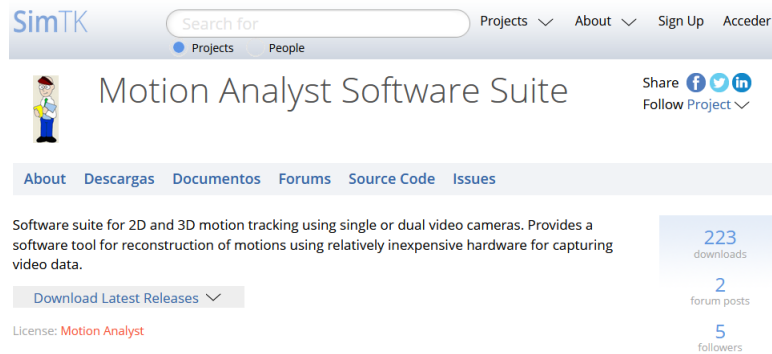


Figura 1.5. Motion Analyst – SimTK.  
Fuente: (Colbrum, 2022)

Este proyecto es un conjunto de herramientas de análisis de movimiento que utilizan imágenes de cámaras de vídeo comunes para medir movimientos en 2D y 3D. Las ubicaciones de los marcadores en el espacio 2D pueden seguirse en el tiempo utilizando MotionAnalyst2D. Cuando se está interesado en la reconstrucción en 3D, es necesario completar el análisis en 2D utilizando dos cámaras que capturan simultáneamente las imágenes. Al combinar los dos resultados 2D con los datos de calibración de la orientación de la cámara, se pueden reconstruir las ubicaciones 3D de esos marcadores originales utilizando MotionAnalyst3D.

#### 1.4.1.5. FreeMoCap

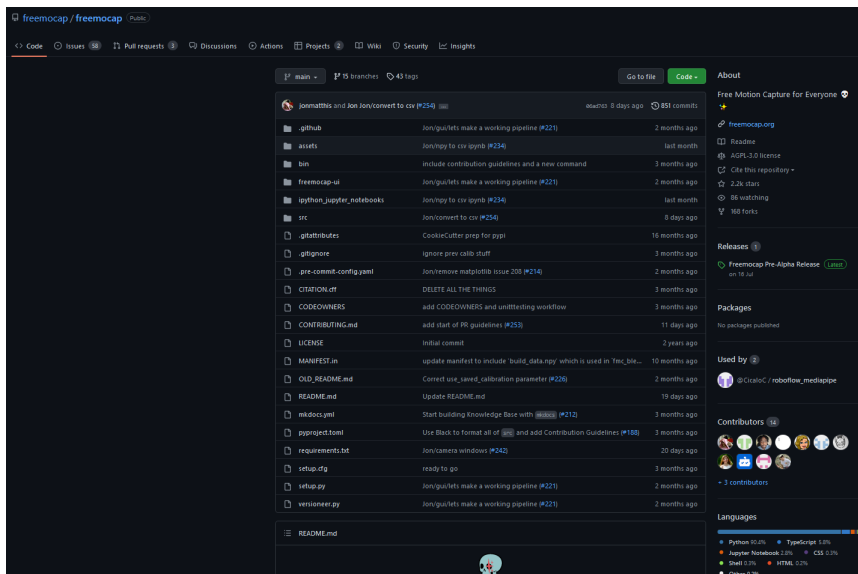


Figura 1.6. FreeMoCap – Repositorio GitHub.  
Fuente: (Matthis y Idehen, 2022)

Es un sistema y plataforma de captura de movimiento de código libre y abierto, de coste mínimo y de grado de investigación, para investigación científica descentralizada, educación y formación.

El proyecto Free Motion Capture (FreeMoCap) tiene como objetivo proporcionar gratuitamente a todo el mundo un software de captura de movimiento sin marcadores de nivel de investigación. Posee un marco de trabajo fácil de usar que conecta una serie de herramientas de código abierto de vanguardia de las comunidades de visión por ordenador y aprendizaje automático para registrar con precisión el movimiento 3D de todo el cuerpo de seres humanos, animales, robots y otros objetos.

Utilizan una filosofía de desarrollo de "diseño universal", con el objetivo de crear un sistema que sirva a las necesidades de un investigador científico profesional y que, al mismo tiempo, sea intuitivo para un niño de 13 años sin formación técnica y sin asistencia externa.

#### 1.4.1.6. DLTdv digitizing tool

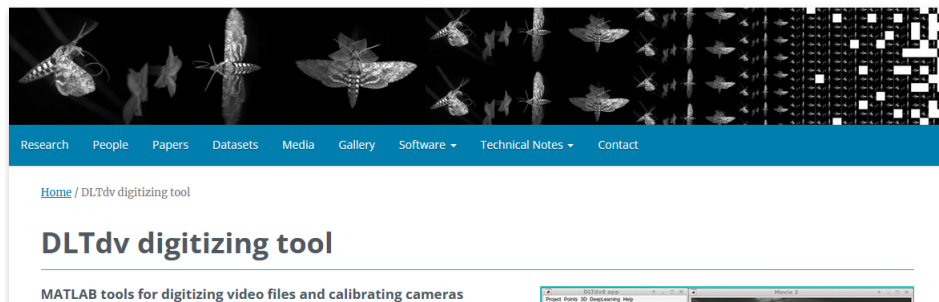


Figura 1.7. DLTdv digitizing tool.  
Fuente: (Hedrick, 2022)

DLTdv8 es un programa de análisis de vídeo que posee una herramienta de Deep Learning junto las capacidades para digitalizar o anotar vídeos en 2D, o 3D a través de una calibración estereoscópica de transformación lineal directa. Está disponible como una aplicación MATLAB, una aplicación binaria empaquetada para Mac, Windows o Linux que se ejecuta con o sin licencia MATLAB, o como código fuente.

Características:

- Lee directamente de archivos de película AVI, mp4 y MOV.
- Seguimiento automático configurable con y sin marcadores, incluyendo Deep Learning

- para el seguimiento de un solo punto, multipunto y multicámara.
- Seguimiento en 3D, con información para el usuario que muestra la línea epipolar y la ubicación del punto reprojectado.
  - Desplazamientos de sincronización del flujo de vídeo especificados por el usuario.
  - Proyectos con vídeos de diferentes frecuencias de cuadro.
  - Muestra hasta 9 flujos de vídeo.
  - Aumenta o disminuye el zoom en cualquier grado utilizando la rueda del ratón y los controles del teclado.
  - Requiere MATLAB r2019b o superiores; la caja de herramientas de análisis de imágenes es necesaria para la identificación del centroide del marcador y la caja de herramientas de aprendizaje profundo es necesaria para el seguimiento basado en el aprendizaje profundo
  - Las versiones binarias empaquetadas tienen acceso completo a todas las funciones, incluido el aprendizaje profundo, y pueden instalarse incluso si se tiene MATLAB instalado.

#### **1.4.1.7. EasyWand**

Es un programa de calibración de video, entre sus características se tiene:

- Proporciona una cómoda interfaz gráfica para calibrar 2 o más cámaras mediante un bastón de calibración.
- También puede incorporar puntos de escena no emparejados, como paisajes de fondo o animales.
- Utiliza SBA (sparse bundle adjustment) para una calibración rápida y precisa.
- Admite la alineación con un sistema de referencia mundial a partir de ejes situados en la vista de la cámara o con la aceleración gravitatoria.
- Especificación de diferentes niveles de optimización que van desde ninguno hasta distancia focal, punto principal y distorsión de lente radial + tangencial.
- Produce resultados de calibración como parámetros intrínsecos + extrínsecos de la cámara y como coeficientes DLT para su uso con DLTdv.
- Probado con MATLAB 2011a y posteriores, no requiere cajas de herramientas.
- Empaquetado con easyCamera, una herramienta de MATLAB para planificar las grabaciones de la cámara.
- Con licencia GPLv3 (Therriault, Fuller, y Jackson, 2014).

#### 1.4.1.8. OBS Studio

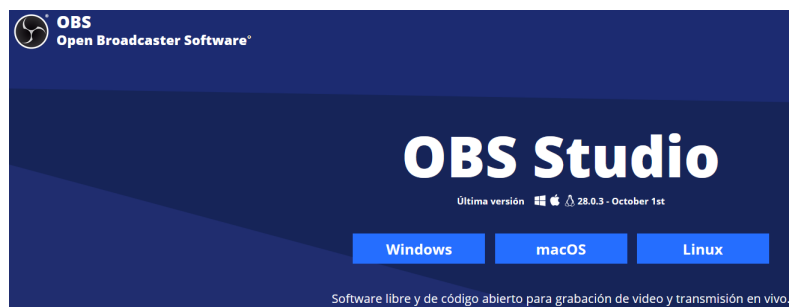


Figura 1.8. OBS Studio.

Fuente: (Studio, 2022)

OBS Studio es un software diseñado para capturar, componer, codificar, grabar y transmitir contenidos de vídeo de forma eficiente. Se distribuye bajo la Licencia Pública General de GNU v2 (o cualquier versión posterior).

#### Source record

Plugin para OBS Studio para añadir un filtro que permite grabar una fuente de forma separada.

#### 1.4.1.9. OpenCap

OpenCap es un paquete de software para estimar la dinámica del movimiento humano en 3D a partir de vídeos de teléfonos inteligentes, combina la visión por ordenador, el aprendizaje profundo y la simulación musculoesquelética para cuantificar la dinámica del movimiento humano a partir de vídeos de smartphones.

OpenCap se compone de una aplicación iOS, una aplicación web y una en la nube. Para recoger datos, los usuarios abren una aplicación en dos o más dispositivos iOS y los emparejan con la aplicación web OpenCap. La aplicación web permite a los usuarios grabar vídeos simultáneamente en los dispositivos iOS y visualizar la cinemática tridimensional (3D) resultante. En la nube, los puntos clave 2D se extraen de los vídeos de las diferentes posiciones de los dispositivos mediante algoritmos de estimación de la postura de código abierto.

Los vídeos se sincronizan en el tiempo mediante correlaciones cruzadas de las velocidades de los puntos clave, y los puntos clave 3D se calculan triangulando estos puntos clave 2D sincronizados. Estos puntos clave 3D se convierten en un conjunto



de marcadores anatómicos 3D más completo utilizando una red neuronal recurrente (LSTM) entrenada con datos de captura de movimiento. A continuación, se calcula la cinemática 3D a partir de las trayectorias de los marcadores utilizando cinemática inversa y un modelo musculoesquelético con restricciones biomecánicas. Por último, las medidas cinéticas se estiman mediante simulaciones dinámicas impulsadas por los músculos que siguen la cinemática 3D.

#### 1.4.1.10. FFmpeg

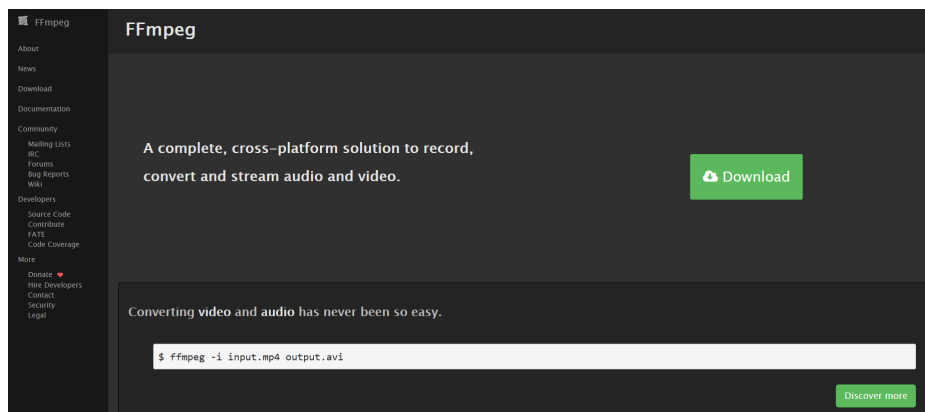


Figura 1.9. FFmpeg.  
Fuente: (FFmpeg, 2022)

El proyecto FFmpeg trata de ofrecer la mejor solución técnicamente posible tanto a los desarrolladores de aplicaciones como a los usuarios finales. Para lograrlo, combina las mejores opciones de software libre disponibles. Se utiliza el código generado para mantener bajas las dependencias de otras librerías y maximizar el mismo al compartirlo entre las partes de FFmpeg.

FFmpeg es el principal marco multimedia, capaz de decodificar, codificar, transcodificar, muxar, demuxar, transmitir, filtrar y reproducir prácticamente todo lo que los humanos y las máquinas han creado. Es compatible con los formatos antiguos hasta los más modernos. No importa si fueron diseñados por algún comité de estándares, la comunidad o una corporación. Es altamente portable: FFmpeg compila, ejecuta y pasa la infraestructura de pruebas FATE en Linux, Mac OS X, Microsoft Windows, los BSD, Solaris, etc. bajo una amplia variedad de entornos de compilación, arquitecturas de máquinas y configuraciones.

## 2. METODOLOGÍA

Se verificará en cada uno de los programas el proceso de calibración, toma de datos, así como la forma de recolectar los mismos. Se presentará la instrumentación general con la que se cuenta para hacer el proceso de captura de movimiento y el uso de la misma de mostrarse compatibilidad con el software.

### 2.1. Instrumentación

#### 2.1.1. Computadores

Se ha trabajado con 3 equipos, de los cuales el equipo 1 es parte del Laboratorio de Biomecánica de la EPN y los 2 equipos restantes pertenecen al autor de esta investigación. En la Tabla 2.1 se detalla las características de los equipos en los que se probó el software.

Tabla 2.1. Características de los equipos empleados.

Características	Equipo 1	Equipo 2	Equipo 3
Tipo	PC	PC	Laptop
Sistema operativo	Windows 10 (21H1)	Windows 10 (21H1)	Windows 10 (21H1)
Procesador	Intel i5 8400	Intel i7 4770	AMD A6-3400M
Memoria RAM	8 gb	28 gb	8 gb
Tarjeta de video	NVIDIA GeForce GT 710	NVIDIA GeForce GTX 1050 Ti	AMD Radeon HD 6520G
VRAM	984 mb	4 gb	495 mb

#### 2.1.2. Cámaras

##### 2.1.2.1. Infrarrojas

Las características de las cámaras que posee el Laboratorio de Biomecánica se muestran en la Tabla 2.2.

Tabla 2.2. Características cámaras infrarrojas.

Elemento	Descripción
<b>Cámara OptiTrack</b>	<b>Cantidad:</b> 9
	<b>Resolución:</b> 1280 x 1024
	<b>Frames:</b> 30 - 240 FPS
	<b>Formatos:</b> RAW (Escala de grises)
	MJPEG (Escala de grises)

Fuente: (OptiTrack, 2022)

##### 2.1.2.2. Ópticas

Las características de las cámaras ópticas usadas se muestran en la Tabla 2.3.

Tabla 2.3. Características cámaras ópticas.

Elemento	Descripción
WebCam HP Truevision HD	<b>Cantidad:</b> 3
	<b>Megapixels:</b> 0.92
	<b>Resolución máxima:</b> 1280 x 720 (0.922 MP)
	<b>Resolución mínima:</b> 176 x 144 (0.025 MP)
	<b>Estándares soportados:</b> QCIF (176 x 144), QVGA (320 x 240), Video CD PAL (352 x 288), nHD (640 x 360), VGA (640 x 480), HD (1280 x 720)
	<b>Frames:</b> 30 FPS

Fuente: *Probar cámara* (2022)

### 2.1.3. Material adicional

En la Tabla 2.4 se detalla los materiales adicionales empleados para las diferentes etapas de la captura de movimiento.

Tabla 2.4. Material adicional.

Elemento	Descripción
Marcador casero	<b>Cantidad:</b> 8
	<b>Material:</b> Espuma flex
	<b>Díámetro:</b> 2.35 [cm]
Cartilla de calibración 1	<b>Cantidad:</b> 1
	<b>Material:</b> Sintra
	<b>Dimensiones:</b> 29.5 x 21 [cm]
	<b>Contenido:</b>
	- Tablero de ajedrez (4 x 5) Tamaño del cuadrado 35 [mm] - Tablero ChAruco (4 x 7)
Cartilla de calibración 2	<b>Cantidad:</b> 1
	<b>Material:</b> Sintra
	<b>Dimensiones:</b> 1.80 x 0.70 [m]
	<b>Contenido:</b>
	- Tablero de ajedrez (8 x 9) Tamaño del cuadrado 8.9 [cm] - Tablero ChAruco (5 x 7)

<b>Elemento</b>	<b>Descripción</b>
<b>Trípode</b>	<b>Cantidad:</b> 2
	<b>Material:</b> Aluminio
	<b>Altura:</b> 1.50 m
<b>Máscara</b>	<b>Cantidad:</b> 1
	<b>Material:</b> Tela
	<b>Tipo:</b> Pasamontañas
<b>Extensión USB</b>	<b>Cantidad:</b> 3
	<b>Dimensión:</b> 5 [m]
	<b>Tipo:</b> USB 2.0
<b>Cinta doblefaz para ropa</b>	<b>Cantidad:</b> 1
	<b>Dimensión:</b> 2.7 [m]
<b>Dispositivos iOS</b>	<b>Cantidad:</b> 2
	<b>Tipo:</b> 1 iPhone 11, 1 iPhone SE 2020
	<b>Año:</b> Modelos superiores a 2018

Se ha dividido en las siguientes secciones de esta investigación (proceso de calibración, protocolo de toma de datos y plan de recolección de datos) los procesos que se hallaron en cada programa con el fin de evidenciarlos.

## **2.2. Proceso de calibración**

No se considerará en esta sección (ni en posteriores) al software OpenMoCap y a RTMoCap, puesto que, en el caso del primero, no se ha podido realizar una compilación del programa en base a la información de la que se dispone en el repositorio; para el segundo, se ha actualizado algunas líneas del código MATLAB, a pesar de eso, los archivos facilitados para los procesos de calibración y grabación no se pueden ejecutar pues se carece del archivo de vinculación con el sistema de cámaras.

Se describe el proceso desarrollado para poner en marcha al software usando los equipos y material disponible.

### 2.2.1. OptiTrack Camera SDK

Se procede a la instalación del programa demostrativo de video en los equipos 1 y 3, obteniendo como resultado la interfaz gráfica que se puede ver en la Figura 2.1.

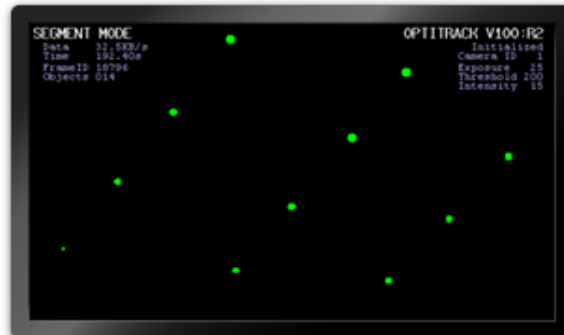


Figura 2.1. Interfaz gráfica – OptiTrack Camera SDK.  
Fuente: (OptiTrack, 2022)

La interfaz gráfica permite realizar la selección de una cámara infrarroja a la vez, así como “calibrar” diferentes modos de iluminación para una mejor captura de marcadores reflectivos, sin embargo, las actividades de grabación de video y rastreo se tienen que realizar con el software propietario de la empresa que requiere de la compra de una licencia.

Al no poseer una licencia para la ejecución de software se procedió a usar FFmpeg y OBS Studio para poder hacer una detección de las cámaras optoelectrónicas con el fin de realizar el proceso de calibración. Con OBS Studio se realizó una búsqueda de cámaras en los equipos, obteniendo los siguientes resultados:

- Equipo 1: Detecta cámara virtual de OBS Studio y cámara óptica auxiliar.
- Equipo 3: Detecta cámara virtual de OBS Studio y cámara web óptica incorporada del equipo.

Con FFmpeg se ejecutó un comando de reconocimiento de fuentes:

```
ffmpeg -sources
```

Se consiguieron los siguientes resultados:

- Equipo 1: Detecta cámara virtual de OBS Studio, cámara óptica auxiliar, entradas de audio.
- Equipo 3: Detecta cámara virtual de OBS Studio y cámara web óptica incorporada del equipo y micrófono.

Debido a que no hubo una detección favorable de los equipos para realizar la calibración, se detuvo el proceso.

### **2.2.2. Motion Analyst - OBS Studio - FFmpeg**

El software Motion Analyst dispone de dos versiones libres de descarga, para rastreo de marcadores en 2D y 3D. Se utiliza para seguir el movimiento de los marcadores en las imágenes de vídeo, (no rastrea realmente los marcadores, sino rastrea los patrones de valores de píxeles de una imagen a otra). Las imágenes deben guardarse como una serie de archivos .tif o .jpg. Los nombres de los archivos deben tener la misma longitud y los últimos caracteres deben ser valores numéricos crecientes que correspondan a los números de fotogramas del vídeo. El seguimiento se realiza en escala de grises, por lo que todas las imágenes deben ser convertidas a este formato antes del seguimiento, por este motivo no se puede cargar un archivo de video, sino que se debe hacer un procesamiento del mismo (grabación, separación de archivos y análisis). Este proceso está auxiliado por 2 programas (OBS Studio, FFmpeg) que realizarán fases previas para que el programa pueda realizar su función de rastreo.

#### **2.2.2.1. Grabación del gesto**

1. Se instala el software OBS Studio en el equipo con el cual se va a realizar la captura.
2. Se instala el plug-in source record en OBS Studio.
3. Se monta 2 de las cámaras web en los trípodes y una en el monitor del equipo a usarse. Las cámaras deberán dirigirse hacia el área donde se vaya a ejecutar el gesto. Estas se pueden colocar a una distancia de 1 [m] entre sí.
4. Se conectan las cámaras a los puertos USB del equipo.
5. Se abre OBS Studio y se configura a las cámaras colocando el nombre adecuado, así como también controlando los parámetros de video (brillo, saturación) y de grabación (Configuración del plug-in de grabación, formato de salida). (Anexo II. Configuración de OBS Studio)
6. Se crea una carpeta para almacenar los videos.
7. Se procede a la grabación del gesto.
8. Se verifica la creación de los archivos de video para cada una de las cámaras en una carpeta.

### 2.2.3. FreeMoCap

1. Se monta 2 de las cámaras web en los trípodes y una en el monitor del equipo a usarse. Las cámaras deberán dirigirse hacia el área donde se vaya a ejecutar el gesto. Estas se pueden colocar a una distancia de 1 [m] entre sí.
2. Se conectan las cámaras a los puertos USB del equipo.
3. Se instala FreeMoCap (Anexo V. Instalación de FreeMoCap).
4. Dentro del entorno `freemocap-env` y en el directorio raíz, escribir lo siguiente:  
`ipython`
5. En el nuevo prompt escribir: `import freemocap`
6. Para acceder a la detección de las cámaras, escribir: `freemocap.RunMe()`
7. El programa procederá a verificar la presencia de los dispositivos y generará una ventana en la cual se puede seleccionar de forma individual o grupal las cámaras conectadas (Figura 2.2).

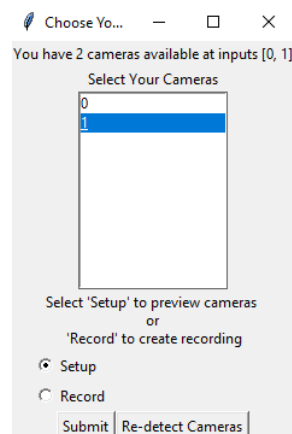


Figura 2.2. FreeMoCap – Verificación de cámaras disponibles.

8. Una vez señalado el/los dispositivos y seleccionando la opción *Setup*, se pulsa sobre el botón *Submit*, el cual llevará a una ventana de configuración que se aprecia en la Figura 2.3.

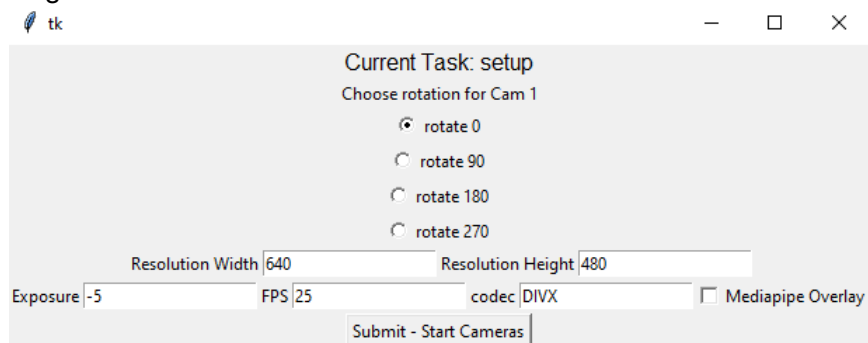


Figura 2.3. FreeMoCap – Calibración de cámaras disponibles.

#### **2.2.4. DLTdv digitizing tool - OBS Studio**

DLTdv digitizing tool, puede realizar el rastreo de movimiento de los marcadores en las imágenes de vídeo. El seguimiento se realiza en videos con escala de grises, por lo que todas las imágenes deben ser convertidas a escala de grises antes del seguimiento, aunque no existe inconveniente en trabajar con videos a color. Para el procesamiento del video se usará como apoyo el programa OBS Studio así como un script de MATLAB para poder verificar las características de las cámaras usadas, posterior a eso se hará uso de la herramienta de detección.

##### **2.2.4.1. Grabación de toma de calibración**

1. Se instala el software OBS Studio en el equipo con el cual se va a realizar la captura.
2. Se instala el plug-in source record en OBS Studio.
3. Se monta 2 de las cámaras web en los trípodes y una en el monitor del equipo a usarse. Las cámaras deberán observar hacia el área donde se vaya a ejecutar el gesto.
4. Se conectan las cámaras a los puertos USB del equipo.
5. Se abre OBS Studio y se configura a las cámaras colocando el nombre adecuado, así como también controlando los parámetros de video (brillo, saturación) y de grabación (Configuración del plug-in de grabación, formato de salida). (Anexo II. Configuración de OBS Studio )
6. Se crea una carpeta para almacenar los videos.
7. Se coloca la cartilla de calibración 2 (tablero de ajedrez) a una distancia de entre 3 a 5 [m] de las cámaras
8. Se procede a la grabación de la cartilla de calibración por menos de un minuto.
9. Se verifica la creación de los archivos de video para cada una de las cámaras en una carpeta.

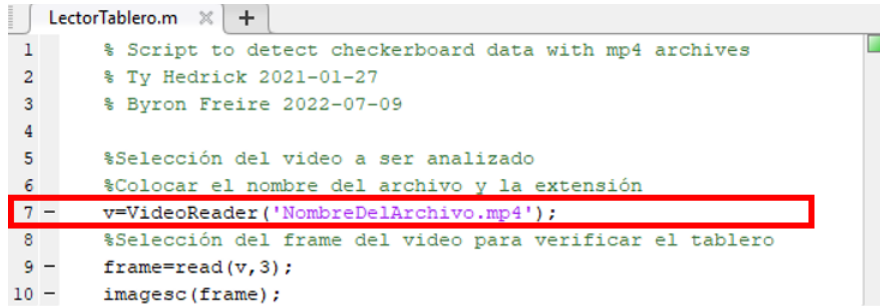
##### **2.2.4.2. Extracción de parámetros de las cámaras**

###### **a. Uso de script LectorTablero.m**

1. Se abre el archivo LectorTablero.m en MATLAB.
2. Se coloca el nombre del archivo de video (obtenido de la cámara que se encuentre de frente al tablero) junto con su extensión en la línea del código, como se muestra



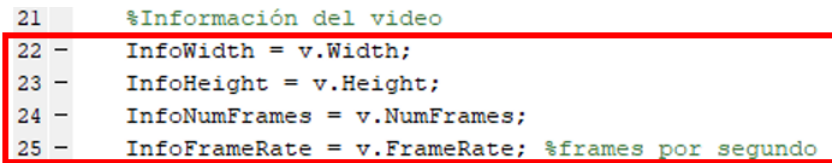
en la Figura 2.4.



```
LectorTablero.m x +
1 % Script to detect checkerboard data with mp4 archives
2 % Ty Hedrick 2021-01-27
3 % Byron Freire 2022-07-09
4
5 %Selección del video a ser analizado
6 %Colocar el nombre del archivo y la extensión
7 - v=VideoReader('NombreDelArchivo.mp4');
8 %Selección del frame del video para verificar el tablero
9 - frame=read(v,3);
10 - imagesc(frame);
```

Figura 2.4. Código para insertar nombre del archivo de video.

3. Se ejecuta el script pulsando *run*.
4. Verificar la generación de los archivos que se muestran en la Figura 2.5 con la información del video.

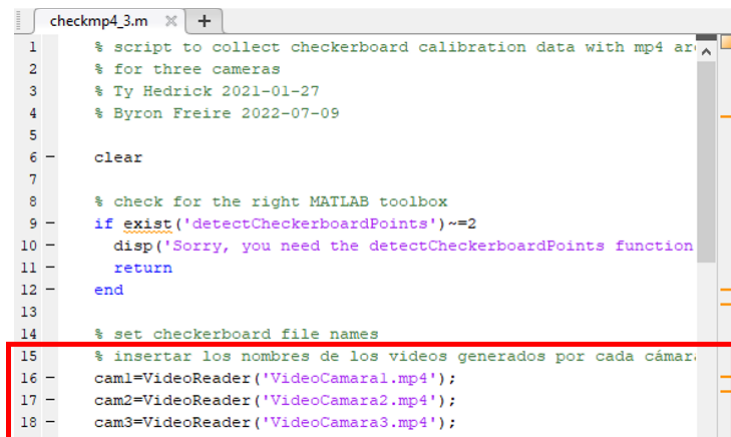


```
21 %Información del video
22 - InfoWidth = v.Width;
23 - InfoHeight = v.Height;
24 - InfoNumFrames = v.NumFrames;
25 - InfoFrameRate = v.FrameRate; %frames por segundo
```

Figura 2.5. Nombre de archivos con datos del video.

#### b. Uso de script checkmp4\_3.m

1. Se abre el archivo checkmp4\_3.m
2. Se coloca el nombre de los archivos de video (obtenidos de las cámaras usadas) junto con su extensión en la línea del código, como se muestra en la Figura 2.6.



```
checkmp4_3.m x +
1 % script to collect checkerboard calibration data with mp4 ar
2 % for three cameras
3 % Ty Hedrick 2021-01-27
4 % Byron Freire 2022-07-09
5
6 - clear
7
8 % check for the right MATLAB toolbox
9 - if exist('detectCheckerboardPoints')~=2
10 - disp('Sorry, you need the detectCheckerboardPoints function
11 - return
12 - end
13
14 % set checkerboard file names
15 % insertar los nombres de los videos generados por cada cámar
16 - cam1=VideoReader('VideoCamara1.mp4');
17 - cam2=VideoReader('VideoCamara2.mp4');
18 - cam3=VideoReader('VideoCamara3.mp4');
```

Figura 2.6. Código para insertar nombre de los archivos de video.

3. Colocar los valores de *expectedSize* (primer valor del WHOS del archivo LectorTablero.m) y *numFrames* (valor de *InfoNumFrames* del archivo LectorTablero.m) en las líneas respectivas, como se muestra en la Figura 2.7.

```

20 % expected checkerboard size
21 % Colocar el primer valor del WHOS (size) de LectorTablero.m
22 - expectedSize=56;
23
24 % number of video frames
25 % Colocar el valor de InfoNumFrames de LectorTablero.m
26 - numFrames=492;

```

Figura 2.7. Código para insertar valores de los archivos de video.

4. Se ejecuta el script pulsando *run*.
5. Se verifica que se hayan generado 2 archivos de extensión *.csv* los cuales poseen la información de fondo y del bastón de calibración con los cuales se extraerán los parámetros de las cámaras.

### c. Uso de easyWand

1. Se abre el archivo *easyWand5.m* y se lo ejecuta en MATLAB.
2. Se carga el archivo *.csv* del bastón de calibración en la sección *Load wand points* (ver Figura 2.8).

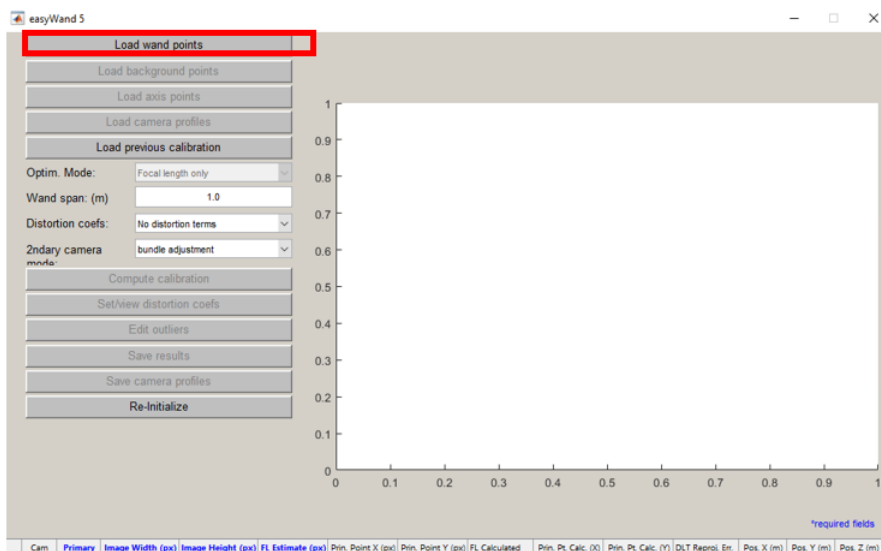


Figura 2.8. Interfaz gráfica EasyWand.

Una vez cargado ese archivo se habilitarán las opciones para la carga del archivo de fondo, puntos de los ejes y perfiles de cámara.

3. Se carga el archivo *.csv* de fondo en la opción *Load background points* (ver Figura 2.9)

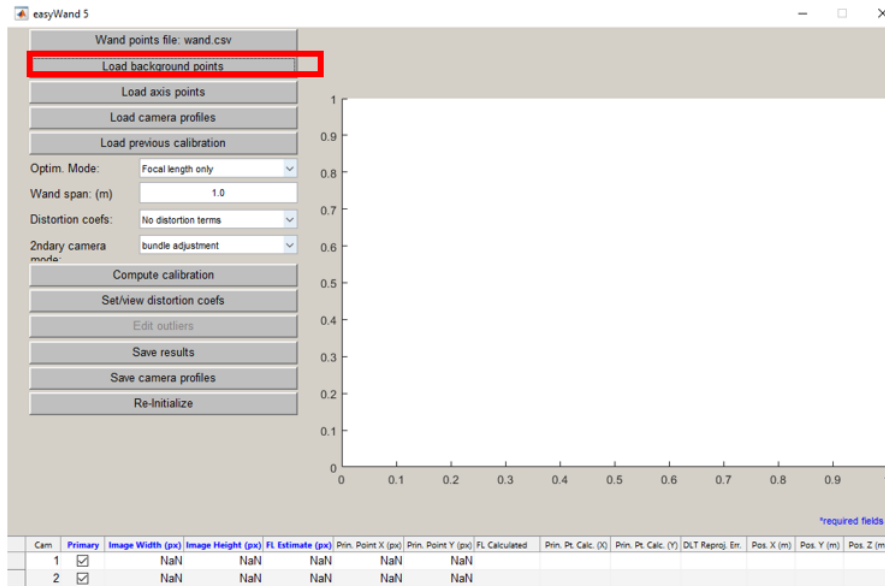


Figura 2.9. Ingreso de parámetros de fondo.

- Se ingresa los valores de *Image Width*, *Image Height* y *FL Estimate*. Los dos primeros valores se pueden obtener de los campos *InfoWidth* e *InfoHeight* del script LectorTablero.m (Figura 2.5), mientras que para el último se recomienda usar la app Camera Calibrator de MATLAB (Anexo IX Uso de Camera Calibrator) para conseguir un número aproximado (Figura 2.10).

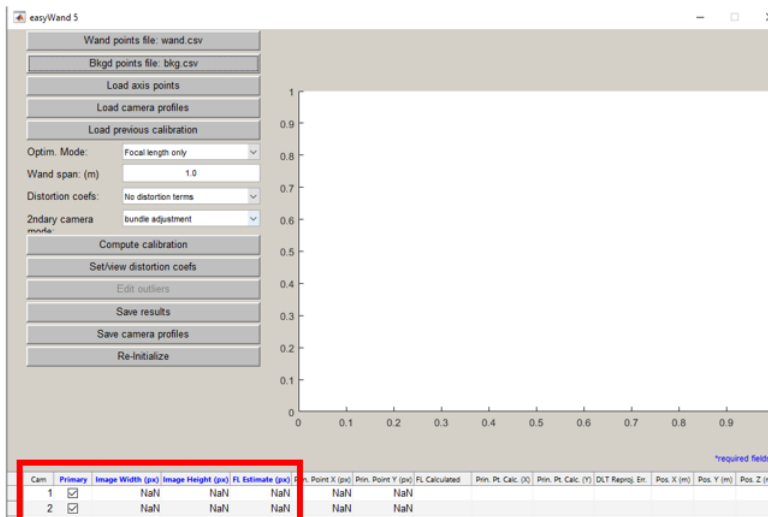


Figura 2.10. Ingreso de parámetros de cámaras.

- Una vez ingresados los valores en la opción *Optim. Mode* se selecciona una de las opciones (se recomienda *Focal Length only*).
- Se ingresa el valor de *Wand span* (determinado en nuestro caso por el largo de la diagonal principal del tablero de calibración, 0.81 [m])
- Se pulsa sobre la opción *Compute calibration* y en función de los valores hallados

para los parámetros de posición en X, Y y Z se reajusta el *Optim. Mode* o *Wand span* para obtener valores cercanos a las distancias reales.

8. Se pulsa en la opción *Save results* para almacenar el archivo de calibración en formato .csv. Este archivo se usará en el programa DLTdv8.

## **2.2.5. OpenCap**

### **2.2.5.1. Instalar la aplicación**

1. Se instala la aplicación *TestFlight* de la tienda de Apple (<https://apps.apple.com/es/app/testflight/id899247664>).
2. Instalar la aplicación OpenCap en los dispositivos a usarse (<https://testflight.apple.com/join/10LnKWBf>).

### **2.2.5.2. Configuración**

Previo a la configuración de las cámaras se debe tomar en cuenta las siguientes recomendaciones generales.

- Asegurarse de estar usando una conexión de internet estable. La carga de vídeos puede fallar con una conexión deficiente.
- No mover las cámaras una vez que se haya superado el paso de calibración. Si se mueve accidentalmente una cámara, iniciar una nueva sesión.
- La cartilla de calibración 1 (tablero de ajedrez) sólo se utiliza para calibrar las cámaras, se puede retirar después de completar con éxito el paso de calibración.

#### **a. Configuración de la cámara**

##### ***Recomendaciones***

- Asegurarse de estar conectado a una conexión de internet estable. La carga de vídeos puede fallar con una conexión deficiente.
- La cámara de la parte posterior del dispositivo iOS es la que se utiliza para la grabación.
- Se debe colocar las cámaras de tal manera que se eviten oclusiones de segmentos.
- Evitar que haya personas en el fondo donde se va a realizar la grabación del gesto.
- Evitar que el participante salga y vuelva a entrar en el volumen de captura. Es necesario que al menos dos cámaras vean todos los segmentos del cuerpo en todo momento durante el movimiento. Dos cámaras son suficientes para muchas tareas, pero las

tareas en las que los segmentos se ocluyen pueden beneficiarse de una cámara frontal adicional.

- Para una configuración de dos cámaras, las cámaras a 30-45° de la línea que mira hacia delante del participante suelen ser las mejores. Las vistas sagitales puras no funcionan bien debido a la oclusión de las extremidades.
- Se recomienda observar al participante realizar la tarea de interés en cada cámara antes de la calibración, para asegurarse de que permanece dentro del campo de visión.

### **Procedimiento**

1. Ingresar a la aplicación por medio de la página web (<https://app.opencap.ai/login>) e ingresar los datos para poder comenzar con el uso. Ver Figura 2.11.

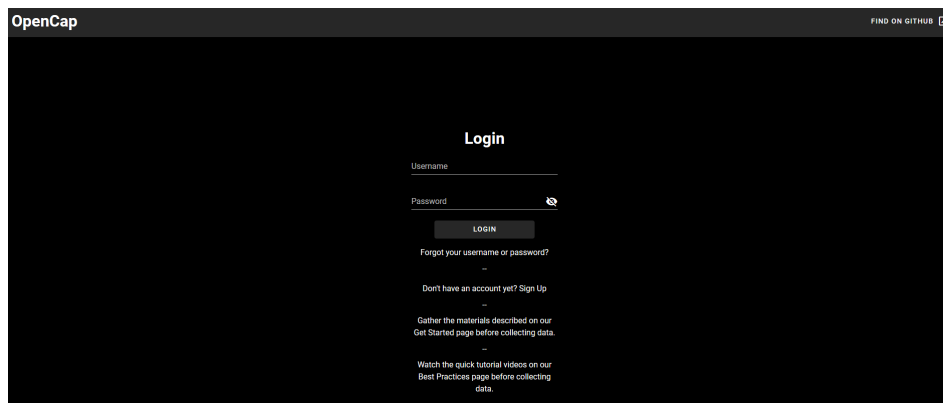


Figura 2.11. Pantalla de ingreso - OpenCap.

2. Seguir las indicaciones que se dan en el paso 1 (*Step 1*) de la nueva pantalla (ver Figura 2.12).

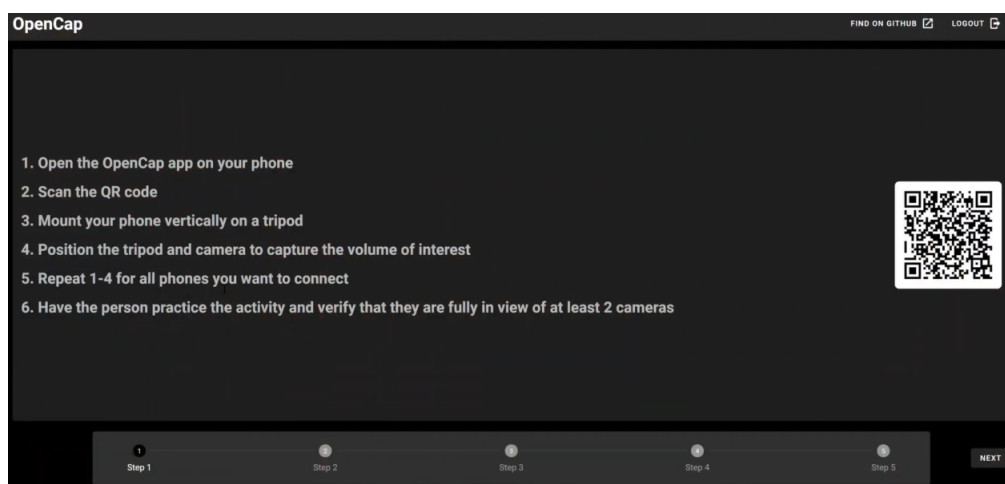


Figura 2.12. Configuración de cámaras – OpenCap.

3. Una vez concluido, seguir al paso 2.

## **b. Calibración de las cámaras**

### ***Recomendaciones***

#### **Entorno**

- Grabar en un entorno bien iluminado.
- Si se graba en el exterior y hay sol directo en el tablero de control, es posible que se requiera oscurecer un poco el tablero de antes de calibrar.

#### **Dimensiones del tablero de control**

- Utilizar un tablero de ajedrez en papel A4 como mínimo, con cuadrados de al menos 35 [mm]. Se recomienda imprimir este tablero.
- Asegurarse de que el tablero tiene un borde blanco en todos sus lados.
- Comprobar las dimensiones de los cuadrados del tablero después de imprimirlo.
- El número de columnas y filas que debe introducirse en la aplicación web es el número de esquinas negras, no el número de cuadrados en las columnas o filas.

#### **Posición de la cartilla de calibración 1**

- El tablero debe ser visible por todas las cámaras.
- El tablero debe estar perpendicular al suelo, con su borde largo paralelo al suelo.
- La línea que apunta al tablero debe bisecar el arco de la cámara.
- El tablero debe estar a menos de 5 [m] de las cámaras.
- La línea que sale del tablero no debe apuntar directamente a una de las cámaras. Mover el tablero hacia arriba, abajo, derecha o izquierda para evitarlo.
- El tablero debe estar en el centro del volumen de captura deseado, lo más cerca posible de las cámaras, siendo visible por todas ellas.

#### ***Procedimiento***

1. En el paso 2 (*Step 2*). Leer con atención las recomendaciones con respecto a la ubicación del tablero (*Place a checkerboard in the scene*).
2. Ingresar los detalles del tablero en la sección *Provide checkerboard details*.
3. Una vez colocado el tablero e ingresado los datos del mismo, pulsar en la opción *calibrate* que se encuentra en la parte inferior derecha de la pantalla (ver Figura 2.13).

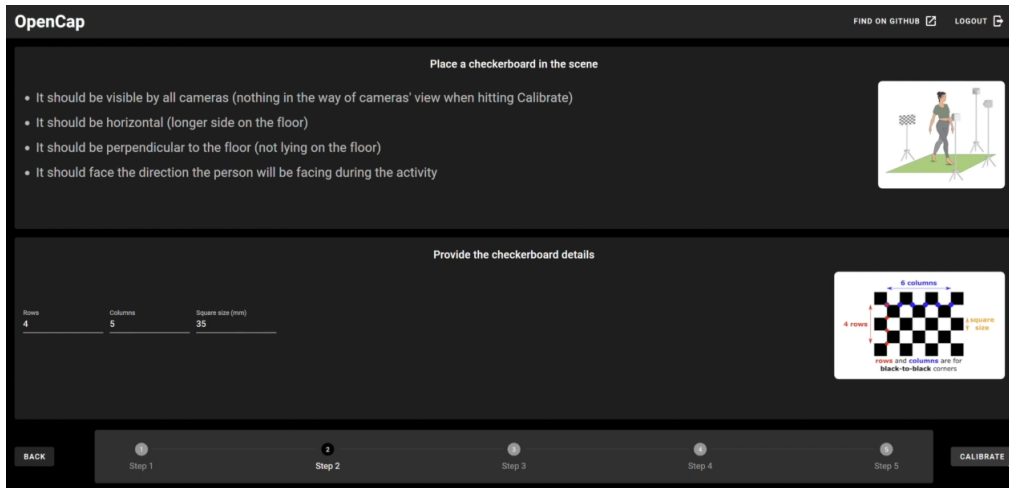


Figura 2.13. Calibración de cartilla – OpenCap.

4. En la siguiente pantalla (*Step 3*) se podrá verificar si la toma de calibración es correcta. Se muestran modelos recomendados de calibración en el lado derecho de la pantalla. Si la toma de calibración coincide con las recomendaciones se pulsa en el ícono de visto verde, caso contrario se pulsa en el ícono de la X y se vuelve al paso 2 (*Step 2*) para volver a hacer la toma de calibración. Finalmente, al obtener la toma de calibración adecuada se pulsa en el botón *confirm* que se encuentra en la parte inferior derecha de la pantalla (ver Figura 2.14) para seguir con el paso 4 (*Step 4*)

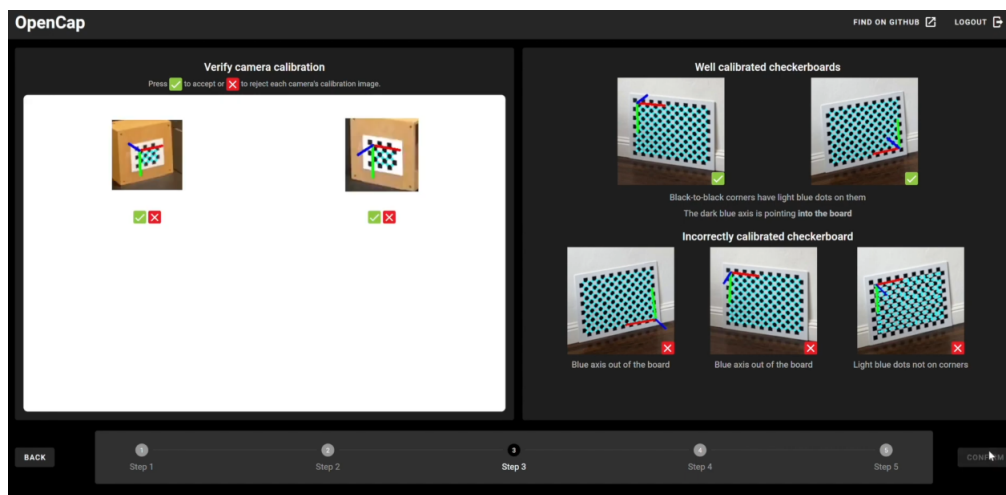


Figura 2.14. Verificación de calibración de cartilla – OpenCap.

## 2.3. Protocolo para toma de datos

### 2.3.1. Motion Analyst - OBS Studio - FFmpeg

#### 2.3.1.1. Separación de archivos individuales

1. Se instala el software OBS Studio en el equipo con el cual se va a realizar la captura.
2. Se instala FFmpeg. (Anexo III. Instalación de FFmpeg)
3. Se ingresa a la consola de Windows (cmd) en modo administrador.
4. Se busca y abre la ruta de la carpeta en la cual se encuentran los videos.
5. Se escribe el siguiente comando:`ffmpeg -i xxxx.mp4 -qscale:v 2 frame%04d.jpg`.  
Las letras xxxx se reemplazan con un nombre para los archivos
6. Se busca la carpeta en el explorador de archivos y se verifica que se hayan creado con la numeración adecuada.

#### 2.3.2. FreeMoCap

1. Se selecciona las características de rotación, ancho y alto de resolución, exposición, FPS, codec y el uso de mediapipe para todas las cámaras y se pulsa en el botón *Submit – Start Cameras*. Posteriormente se mostrará el video de las cámaras por separado con el nombre que el equipo les ha asignado, así como si se detecta o no las esquinas de la cartilla de calibración ChAruco (ver Figura 2.15).



Figura 2.15. FreeMoCap – Verificación de cartilla de calibración.

2. Colocar la cartilla de calibración ChAruco en una posición fija, a una distancia tal, que las cámaras empleadas puedan reconocer adecuadamente el patrón con los 24 puntos de detección. Se recomienda que la distancia no supere los 5 [m] desde la posición de las cámaras.
3. Una vez que las cámaras reconozcan el patrón de calibración y este mantenga la detección del tablero ChAruco (verificar que los 24 puntos se mantengan detectados) se puede pasar al siguiente paso. El usuario puede usar tomas a color o blanco y negro.



### **2.3.3. DLTdv digitizing tool - OBS Studio**

#### **2.3.3.1. Grabación del gesto**

1. Se sigue los pasos de la sección 2.2.4.1 hasta el número 6.
2. El paciente se coloca en una posición visible para todas las cámaras.
3. Se procede a la grabación del gesto por el tiempo que se considere necesario.
4. Se verifica la creación de los archivos de video para cada una de las cámaras en una carpeta.

### **2.3.4. OpenCap**

#### **a. Toma de datos**

##### ***Recomendaciones***

- Es útil utilizar ropa ajustada que contraste con un fondo sencillo y zapatos que contrasten con el suelo y con los pantalones.
- Comenzar a grabar al menos 1 [s] antes de que el participante comience el movimiento.
- En el caso de la marcha en caminadora o de los movimientos bastante estáticos (por ejemplo, la evaluación del balanceo postural), hacer que el participante levante una mano en el aire por encima de los hombros y la vuelva a bajar en algún momento del ensayo. Esto facilita la sincronización de la cámara.
- Los puntos de color junto a los nombres de los ensayos indican el estado del ensayo: verde (procesado con éxito), naranja (en proceso), rojo (error en el procesamiento). Es posible que se desee volver a recoger los ensayos con un punto rojo. Descargar los datos sólo cuando todos los puntos se hayan vuelto verdes (o rojos). Haga clic en los puntos rojos para obtener información sobre lo que ha fallado.
- Se puede recopilar datos de otro participante sin volver a calibrar las cámaras pulsando "Nueva sesión misma configuración"

##### ***Procedimiento***

1. En la ventana del paso 4 (*Step 4*) ingresar los datos del sujeto de estudio en el espacio *provide subject's details* (Figura 2.16); el campo de nombre se puede llenar con un identificador para que exista confidencialidad (por ejemplo, sujeto 01, paciente 01).

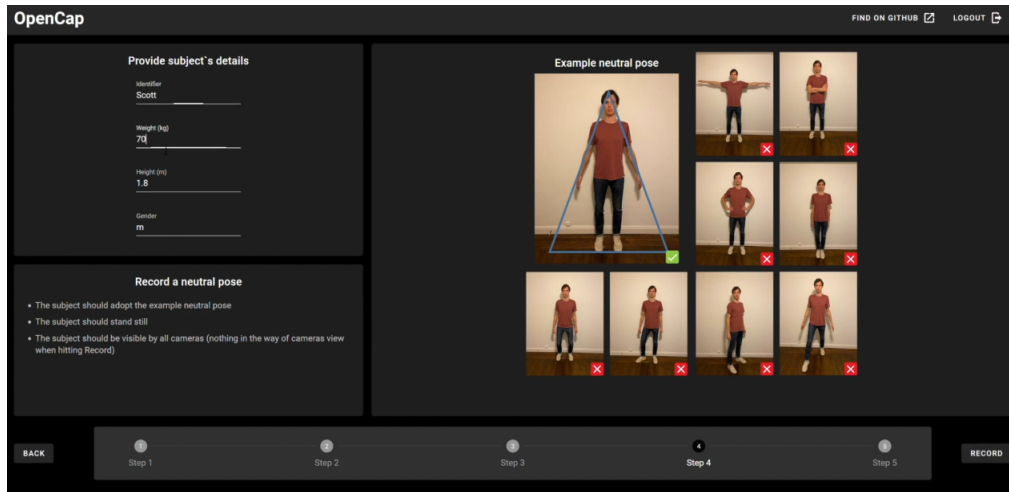


Figura 2.16. Ingreso de datos y toma de pose neutral - OpenCap

2. A continuación, se procederá a grabar una postura neutral. En la parte lateral derecha se puede apreciar ejemplos que sirven como guía. Una vez que el sujeto esté listo para la grabación se pulsa en el botón *record*.
3. El sujeto se deberá mantener estático por al menos 10 [s]. Después de esto se detendrá la grabación y la aplicación le informará sobre el éxito o falla de la captura.
4. En caso de falla se debe repetir el procedimiento.

## 2.4. Plan de recolección de datos

### 2.4.1. Motion Analyst - OBS Studio - FFmpeg

#### 2.4.1.1. Análisis

1. Se descarga el archivo MotionAnalyst\_Version 1.6.zip.
2. Se crea una carpeta para descomprimir los contenidos del archivo .zip.
3. Se abre el archivo MotionAnalyst.m en la carpeta donde estaba alojado.
4. Se ejecuta el archivo dentro de MATLAB.
5. Se selecciona la opción Track 2D Motion.
6. Inicia el análisis. (Anexo IV. Análisis con Motion Analyst)

#### 2.4.2. FreeMoCap

1. Se pulsa *ESC*, y en la siguiente ventana se verifica el nombre de la sesión de grabación y la ruta de la misma (Figura 2.17).

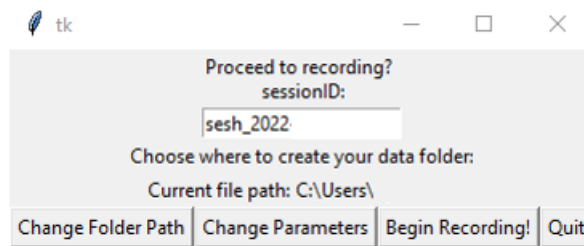


Figura 2.17. FreeMoCap – Inicio de grabación.

- Si no se desea trabajar en la ruta por defecto, se pulsa en el botón *Change Folder Path* para cambiarla.
- Pulsando sobre *Change Parameters*, se puede reconfigurar a las cámaras.
- El botón, *Begin Recording!* permite el inicio de grabación de la sesión.
- Con *Quit* se cierra la ventana y se vuelve a la consola Anaconda.

2. Pulsar *Quit* para salir de la sesión.

En el caso de FreeMoCap, no hay la necesidad de usar software auxiliar para calibración o grabación de los videos, ya que este posee herramientas propias.

### 2.4.3. DLTdv digitizing tool - OBS Studio

#### 2.4.3.1. Análisis

1. Se pulsa sobre el ícono creado en el escritorio para ejecutar el programa.
2. En la interfaz que se presenta se pulsa en la opción *Project* y después en *New* (Figura 2.18).

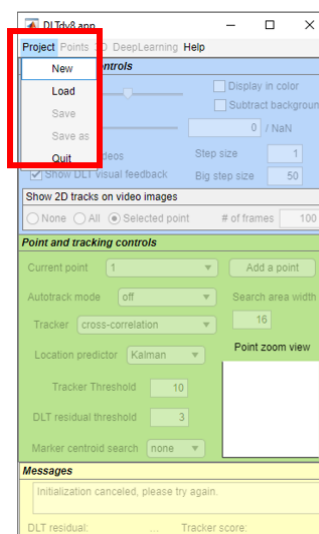


Figura 2.18. Interfaz de DLTdv8.

3. En el siguiente menú (Figura 2.19) se selecciona la cantidad de videos que se tiene (un video por cada cámara con la que se hizo la grabación).

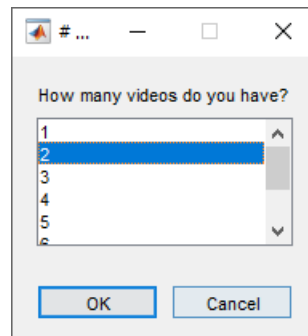


Figura 2.19. Menú de selección de número de videos.

Una vez escogida la cantidad de videos, el programa abrirá una ventana para que el usuario indique la ubicación de los mismos y estos puedan ser cargados.

4. Aparecerá un nuevo menú en el que se preguntará si los videos han sido calibrados mediante el método DLT (Figura 2.20).

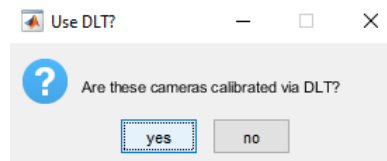


Figura 2.20. Calibración con DLT – DLTdv8.

Se debe pulsar en *yes* y seleccionar el archivo .csv que se generó anteriormente al usar *easyWand*.

5. Se reorganiza el espacio de trabajo con los videos grabados, la interfaz de DLTdv8 y la sección de avance de tiempo como se muestra en la Figura 2.21.

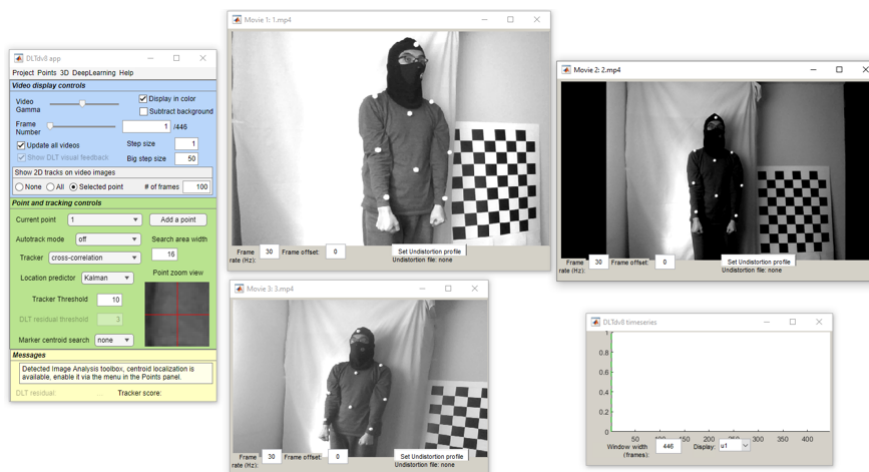


Figura 2.21. Organización de interfaz – DLTdv8.

6. Pulsar sobre la barra de título del video en el cual se va a incluir el primer punto

de rastreo y seleccionar este en al menos 2 videos para que el programa pueda triangular la posición del mismo punto en el tercer video como se aprecia en la Figura 2.22.

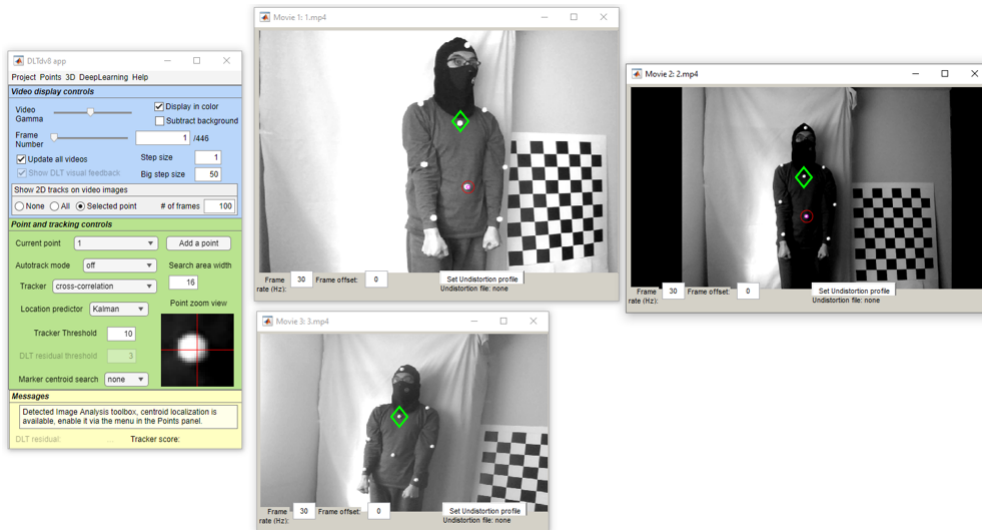


Figura 2.22. Inclusión de puntos de rastreo – DLTdv8.

7. En la sección *Point and tracking controls* pulsar en la opción *add point* para añadir otro punto de rastreo y repetir el paso 6.
8. Una vez creados los puntos de rastreo para los videos, en la sección *Point and tracking controls* seleccionar la opción *multitrack* para hacer seguimiento a múltiples puntos (Figura 2.23).

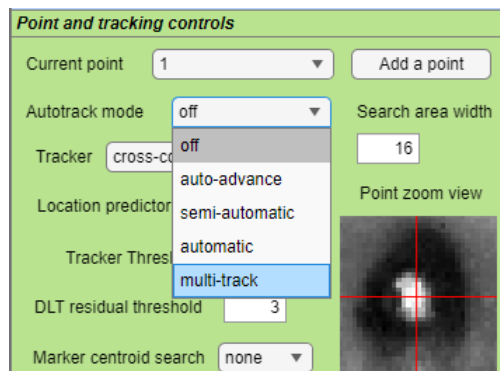


Figura 2.23. Configuración de rastreo – DLTdv8.

9. Pulsar sobre la barra de título del primer video cargado y después presionar la tecla *f* para que el video avance automáticamente y se haga el rastreo de los puntos seleccionados.
10. Para exportar los puntos, pulsar en la opción *export* y seleccionar la carpeta en la cual se almacenarán los datos como se ve en la Figura 2.24.

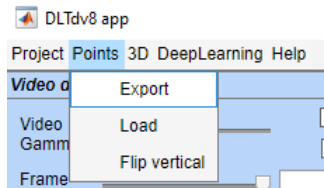


Figura 2.24. Exportación de puntos de rastreo – DLTdv8.

11. En el siguiente menú mantener el prefijo sugerido por el programa para almacenar la información y pulsar en *OK* (Figura 2.25).

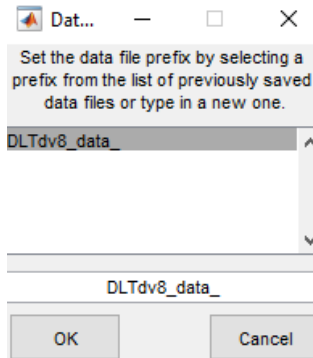


Figura 2.25. Generación de archivo de rastreo – DLTdv8.

12. Seleccionar la opción *flat* en el siguiente menú (Figura 2.26).



Figura 2.26. Selección de tipo de formato – DLTdv8.

13. Si desea puede crear los intervalos de confianza pulsando *yes* caso contrario se pulsa en la opción *no* del siguiente menú (Figura 2.27).

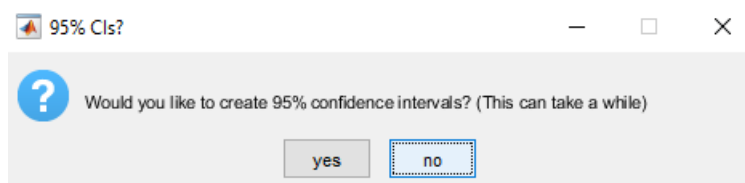


Figura 2.27. Creación de intervalos de confianza – DLTdv8.

14. Finalmente, se crearán 4 archivos de datos de extensión *.csv* con la información de rastreo:
  - DLTDVdv8\_data\_offsets.csv.
  - DLTDVdv8\_data\_xypts.csv.
  - DLTDVdv8\_data\_xyzpts.csv.

– DLTDVdv8\_data\_xyzres.csv.

Los cuáles servirán para exportación a otros programas y validarlos.

#### 2.4.4. OpenCap

1. El siguiente paso consiste en grabar el gesto que realizará el sujeto. En la nueva pantalla se coloca el nombre de la prueba a realizarse en la sección *trial name* y se comienza la grabación pulsando sobre el botón *start recording*, como se ve en la Figura 2.28.



Figura 2.28. Pantalla de inicio de grabación – OpenCap.

2. Terminado el gesto, se detiene la grabación y el software comenzará a procesar los datos del mismo en la nube para generar los datos de exportación, como se ve en la Figura 2.29.

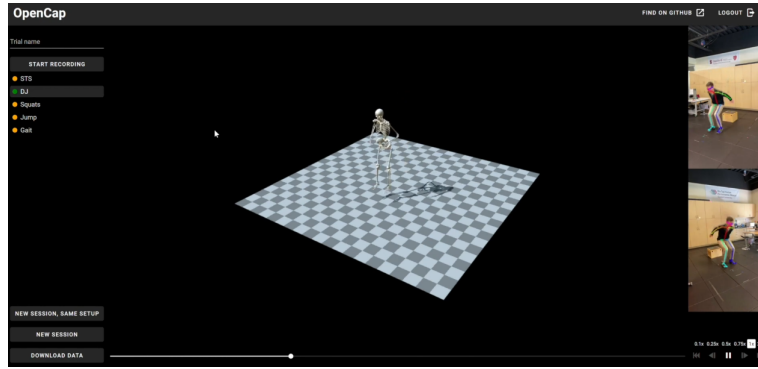


Figura 2.29. Generación de simulación – OpenCap.

3. Se descarga los datos una vez que se coloca un punto verde junto al nombre del ensayo realizado.



### 3. RESULTADOS Y DISCUSIÓN

Una vez concluida la etapa de pruebas con los distintos programas y determinar las etapas de calibración, toma y recolección de datos se establece las características de análisis de cada uno.

Se probó los programas en 3 equipos, tanto el equipo 1 y 2 fueron útiles para la realización de pruebas de campo gracias a las características que presentan, lo que no sucedió con el equipo 3 el cual no permitió cumplir totalmente las etapas.

#### 3.1. Análisis

##### 3.1.1. Motion Analyst

En la Figura 3.1 se muestra el esquema de trabajo para Motion Analyst.



Figura 3.1. Proceso de análisis para Motion Analyst – OBS Studio – FFmpeg.

El proceso de análisis consta del uso auxiliar de dos programas (OBS Studio y FFmpeg). Se utiliza los mismos ya que Motion Analyst no posee una interface propia para hacer calibración, toma y recolección de datos propia. El procesamiento del video por medio de FFmpeg proporciona los datos iniciales por medio del cual se realiza el rastreo 2D de los puntos que el usuario ha seleccionado. Los datos con los que trabajará el programa dependerán de la calidad de videos con los que inicialmente se hicieron las tomas.

##### 3.1.2. FreeMoCap

Para FreeMoCap se tiene el esquema de la Figura 3.2.

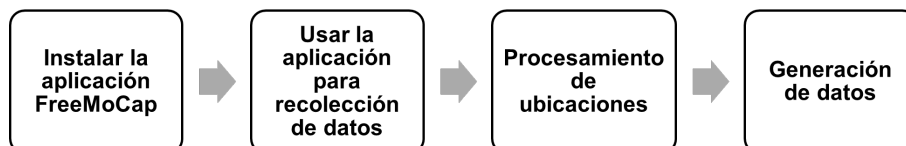


Figura 3.2. Proceso de análisis para FreeMoCap.

A diferencia de Motion Analyst y DLTdv, FreeMoCap no requiere de software adicional para el proceso en general. La instalación es sencilla por medio de las instrucciones que facilita el desarrollador. Se recomienda que la cartilla de calibración que se use

para el proceso de calibración sea visible para las cámaras de trabajo, así como vigilar las características de iluminación del sitio donde se realiza la captura de gesto. Existen inconvenientes cuando se trabaja con cámaras que estén muy próximas entre sí, por lo que se debe mantener una distancia de al menos 1 metro entre ellas. En el caso de que el programa se cierre repentinamente se debe volver a cargar la interfase para repetir el procedimiento. Se puede incorporar al análisis el uso de procesamiento mediante librerías de inteligencia artificial y se está trabajando en la mejora de la interfaz gráfica para la configuración general. No se requiere de una toma estática para el paciente.

La recuperación de los datos dependerá de la incorporación de la librería de inteligencia artificial DeepLabCut (<https://www.mackenziemathislab.org/deeplabcut>) ya que la librería OpenPose (<https://github.com/CMU-Perceptual-Computing-Lab/openpose>) que se usa en la versión estudiada no genera datos que puedan ser exportados.

### 3.1.3. DLTdv digitizing tool - OBS Studio

La forma de procesamiento de DLTdv se aprecia en la Figura 3.3.

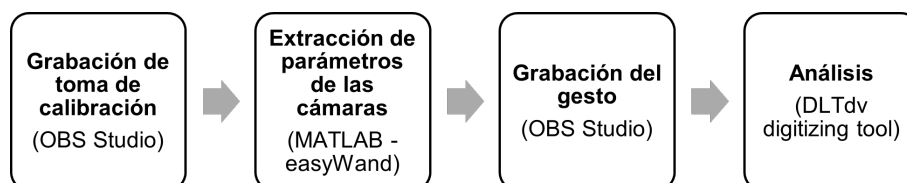


Figura 3.3. Proceso de análisis para DLTdv – OBS Studio.

De la misma manera que con Motion Analyst, se necesita de dos programas por separado para la gestión del proceso, además de que uno de estos funciona dentro de un software privativo (MATLAB). A pesar de este inconveniente se puede realizar un análisis con la herramienta. El requerimiento de los parámetros de las cámaras es uno de los pasos que presenta dificultad, ya que se debe realizar mediciones propias de la distancia de las cámaras a la cartilla de calibración, así como también de la altura de estas, para poder comprobar posteriormente si se ha calculado adecuadamente los valores para los ejes  $x$ ,  $y$  y  $z$ . Estos valores deben revisarse para que al instante de generar el archivo de calibración de cámaras que se usará en el DLTdv no existan problemas en la ubicación y rastreo automático de los puntos. Para la investigación se utilizó una configuración “frontal” de las cámaras, es decir, estas miraban de frente al paciente y no se ubicaron en otros planos.

Una de las ventajas que presenta la herramienta es la de análisis con red neuronal interna, la cual permite el entrenamiento por medio de múltiples videos además de que no se requiere que el paciente use marcadores lo cual ahorra tiempo en la preparación de la grabación.

### 3.1.4. OpenCap

El proceso de análisis para OpenCap se muestra en la Figura 3.4.

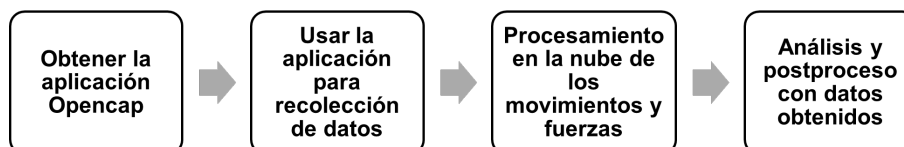


Figura 3.4. Proceso de análisis para OpenCap.

OpenCap se diferencia de los demás programas al funcionar de manera online mediante el registro a una plataforma que será la encargada de guiar al usuario para el proceso de captura de movimiento y análisis. Se incluye en cada etapa recomendaciones para que la captura sea adecuada y no necesita la inclusión de marcadores en el paciente para el seguimiento del gesto. El resultado obtenido se puede exportar a OpenSim para un estudio detallado.

## 3.2. Resultados

Los resultados obtenidos de la revisión, se muestran en la Tabla 3.1, Tabla 3.2, Tabla 3.3, Tabla 3.4 y Tabla 3.5, en el caso de trabajar en combinación con otros programas, las características se refieren al primero que se menciona.

Tabla 3.1. Detalles OptiTrack Camera SDK.

Software	Características
<b>OptiTrack Camera SDK</b>	<ul style="list-style-type: none"> <li>- Descarga libre.</li> <li>- No permite una calibración de un volumen de control.</li> <li>- No permite grabación de video.</li> <li>- No permite rastreo de marcadores.</li> <li>- No reconoce cámaras ópticas.</li> <li>- Permite observar una cámara a la vez, usando el programa demostrativo.</li> </ul>

Tabla 3.2. Detalles Motion Analyst – OBS Studio – FFmpeg.

<b>Software</b>	<b>Características</b>
<b>Motion Analyst OBS Studio FFmpeg</b>	<ul style="list-style-type: none"> <li>- Descarga libre, requiere de MATLAB para la ejecución de los archivos.</li> <li>- Trabaja con al menos 2 cámaras.</li> <li>- Requiere de programas auxiliares para la adquisición de video (OBS Studio) y de los trames que lo componen (FFmpeg).</li> <li>- Trabaja con imágenes en blanco y negro.</li> <li>- Permite rastreo de puntos en 2D.</li> <li>- Permite una calibración en 3D, pero no especifica la construcción del calibrador.</li> <li>- Puede definir las ubicaciones de los marcadores para múltiples formatos y tamaños de imagen, pero está limitado a rastrear sólo imágenes .jpg, tif o .tiff.</li> </ul>

Tabla 3.3. Detalles FreeMoCap.

<b>Software</b>	<b>Características</b>
<b>FreeMoCap</b>	<ul style="list-style-type: none"> <li>- Descarga libre.</li> <li>- Código disponible en un repositorio para uso particular.</li> <li>- Trabaja con al menos 3 cámaras.</li> <li>- No necesita programas auxiliares para la grabación y calibración de videos.</li> <li>- No reconoce cámaras optoelectrónicas.</li> <li>- Puede generar archivos de datos, pero al instalar los complementos se generan conflictos de compatibilidad.</li> </ul>

Tabla 3.4. Detalles DLTdv digitizing tool – OBS Studio.

<b>Software</b>	<b>Características</b>
<b>DLTdv digitizing tool OBS Studio</b>	<ul style="list-style-type: none"> <li>- Descarga libre</li> <li>- Trabaja con al menos 2 cámaras</li> <li>- No reconoce cámaras optoelectrónicas.</li> <li>- Posee una herramienta de red neuronal para entrenamiento.</li> <li>- No necesita que al paciente se le coloquen marcadores.</li> <li>- El tiempo de procesamiento depende del tamaño de video ingresado.</li> </ul>

Tabla 3.5. Detalles OpenCap.

<b>Software</b>	<b>Características</b>
<b>OpenCap</b>	<ul style="list-style-type: none"> <li>- Descarga libre.</li> <li>- Trabaja con al menos 2 cámaras.</li> <li>- Realiza la captura con dispositivos Apple®, con modelos que posean versiones superiores al año 2018.</li> <li>- Rápido y portable.</li> <li>- Seguro en el manejo de información digital.</li> <li>- Interpretación de movimiento y fuerzas preciso.</li> <li>- No reconoce cámaras optoelectrónicas.</li> </ul>

Para el uso en laboratorio se elaboró especificaciones las cuales se presentan en la Tabla 3.6.

Tabla 3.6. Capacidades de los programas utilizados para laboratorio.

Software	Tipo de cámaras requeridas	Mínimo número de cámaras	Requerimiento de marcadores en el paciente	Tiempo de procesamiento	Portabilidad
OpenMoCap	Optoelectrónicas	3	SI	_____	_____
RTMoCap	Optoelectrónicas	3	SI	_____	_____
OptiTrack (Camera SDK)	Optoelectrónicas	3	SI	_____	_____
Motion Analyst OBS Studio FFmpeg	Ópticas	3	NO	En función del tamaño del archivo de video	SI
FreeMoCap	Ópticas	2	NO	< 15 minutos	SI
DLTdv digitizing tool OBS Studio	Ópticas	3	NO	En función del tamaño del archivo de video	SI
OpenCap	Ópticas	2	NO	< 15 minutos	SI

Al conocer algunas de las características de las herramientas utilizadas y en función de los procesos que se han presentado, se muestra en la Tabla 3.7, un resumen de las etapas de análisis, así como el cumplimiento o no de las mismas.

Tabla 3.7. Comparación de capacidades de los programas utilizados.

Software	Calibración	Toma de datos	Recolección de datos
OpenMoCap	_____	_____	_____
RTMoCap	_____	_____	_____
OptiTrack (Camera SDK)	_____	_____	_____
Motion Analyst OBS Studio FFmpeg	X	X	_____
FreeMoCap	X	X	_____
DLTdv digitizing tool OBS Studio	X	X	X
OpenCap	X	X	X

En función a lo mostrado en la tabla anterior se recomienda el uso de dos programas: DLTdv digitizing tool (en combinación con OBS Studio) y OpenCap. Es recomendable que para el primero se realicen más ensayos de calibración para obtener mejores resultados en la calibración de cámaras, los mismos que servirán tanto para la exportación a otros programas como también para el entrenamiento de la red neuronal interna que posee, siendo esta una de las herramientas que brindará utilidad para estudios de múltiples videos. En el caso de OpenCap, se recomienda la adquisición de los dispositivos sugeridos por los creadores de la aplicación para poder realizar capturas de video con al menos 3 cámaras con el fin de evitar oclusiones y obtener mayor cantidad de datos para ser procesados en la nube, de esta manera, generar datos de exportación válidos para ser llevados a OpenSim y proceder a un estudio biomecánico.

OpenCap, ha puesto a disposición del público dos versiones para poder trabajar en modo offline:

1. Código fuente para calcular la cinemática de los videos (<https://github.com/stanfordnmb1/opencap-core>)
2. Código para el procesamiento de datos y generación de simulaciones dinámicas (<https://github.com/stanfordnmb1/opencap-processing>)

Sin embargo, estas aún requieren del procesamiento de video propio de los dispositivos mencionados anteriormente por lo que se puede realizar grabaciones y una vez terminadas, usarlas ya sea con el primer o segundo código proporcionado.

### 3.3. Discusión y validación

En la sección de metodología se han presentado 3 etapas de trabajo con los programas, las cuales se muestran en la Figura 3.5.

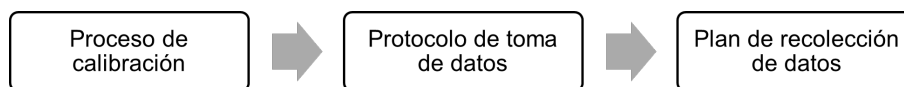


Figura 3.5. Etapas de trabajo.

La primera de las etapas tuvo que ser omitida en 3 de los programas investigados. Los problemas presentados fueron:

- OpenMoCap: Los archivos disponibles en el repositorio no se podían volver a compilar (el autor no indica la forma de instalación ni los programas que se pueden utilizar).
- RTMoCap: A pesar de poseer indicaciones y tener los archivos en el repositorio, falta un archivo que es el que proporciona la conexión con las cámaras, ya sean estas optoelectrónicas u ópticas.
- OptiTrack Camera SDK: Este kit de desarrollo presenta un programa demostrativo que funciona adecuadamente para calibración de parámetros de cámaras optoelectrónicas, una cámara a la vez, sin embargo, para poder usarlo por separado se debe combinar con software privativo de la misma empresa, ya que no realiza captura de video ni rastreo de marcadores.

Estos inconvenientes impidieron el desarrollo de las etapas siguientes en los programas mencionados. La primera etapa concluyó adecuadamente para 4 programas, dos de los cuales trabajaron en conjunto con hasta dos herramientas de software libre debido a las

limitaciones de trabajo en captura de video y en descomposición de frames.

La segunda etapa no presentó novedades en 4 de los programas restantes (véase la Tabla 3.7) Para la tercera etapa únicamente se pudo recuperar datos de dos de los cuatro programas restantes: DLTdv – OBS Studio y OpenCap. La primera combinación entrega datos útiles de coordenadas para ser exportados a otro programa de análisis, sin embargo, se debe revisar la calibración realizada por easyWand además de realizar un cambio de valores en las coordenadas, ya que las que se proporcionan, tienen intercambiados los ejes  $x$  e  $y$ .

OpenCap no presenta inconvenientes en las fases que propone, siendo este el programa que entrega mejor calidad en los resultados, así como también la exportación para poder validar datos. En la sección de análisis se presenta las etapas de proceso de cada uno de los programas usados, es de notar como, de forma general, estas no superan las 4 fases, las cuales están involucradas con las etapas planteadas al inicio de la investigación.

Finalmente, en la sección de resultados se aprecia tablas descriptivas en las que se presenta las características destacadas de las herramientas probadas. Hay que destacar la combinación de software libre que se hace para Motion Analyst y DLTdv, ya que se suple la carencia de métodos propios de adquisición y procesamiento de video con opciones libres que no perjudican los resultados de análisis propios de las herramientas, además de que estas permiten una instalación y configuración sencilla.

La Tabla 3.6, es de especial interés puesto que orienta sobre los requerimientos mínimos de los que se debe disponer en un laboratorio para poder realizar capturas de movimiento, así como la portabilidad del sistema que se vaya a seleccionar.

En la Tabla 3.7, se aprecia el resultado final del análisis. Se ha generado datos con OpenCap, los cuáles se han validado en OpenSim 4.4. A continuación un detalle de lo obtenido:

Se realizaron 3 grabaciones: una para pose neutral, otra para flexión y una para marcha como se puede visualizar en la Figura 3.6.

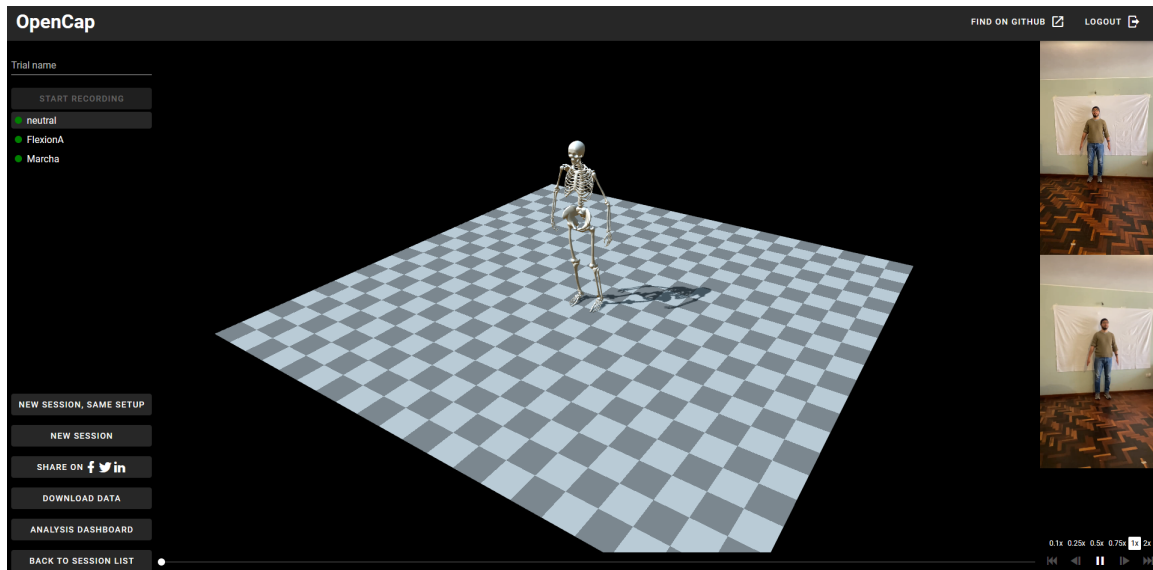


Figura 3.6. Grabación de gestos – OpenCap.

Estas grabaciones se hicieron respetando las consideraciones y recomendaciones proporcionadas por los creadores de la plataforma. Se trabajará con la grabación de marcha.

Para acceder y probar los datos se siguen los siguientes pasos:

1. Ingresar a la Plataforma OpenCap.
2. Se pulsa en la opción *download data* para obtener el archivo comprimido con el detalle del gesto que se ha grabado.
3. Descomprimir la carpeta.
4. Abrir OpenSim 4.4.
5. Cargar el modelo pulsando en el menú *File* y la opción *Open model*. Buscar la ubicación de la carpeta con los datos de la sesión. El archivo a cargar se encuentra en la ruta *OpenSimData/Model/* y posee la extensión *.osim*. Se muestra en la Figura 3.7 el resultado de la carga.



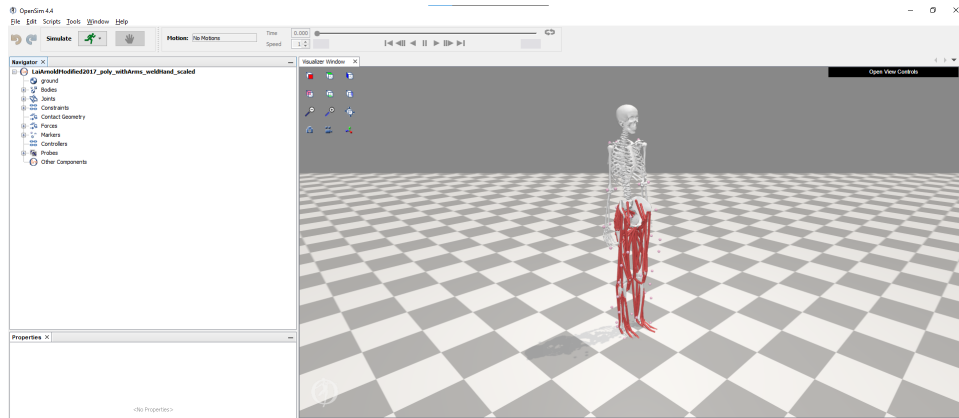


Figura 3.7. Carga de modelo – OpenSim

6. Cargar el movimiento pulsar en el menú *File* y después en la opción *Load Motion*. Buscar la ubicación de la carpeta con los datos de la sesión. El archivo a cargar se encuentra en la ruta *OpenSimData/Kinematics/* y posee la extensión *.mot*. El resultado de esta carga de datos se aprecia en la Figura 3.8

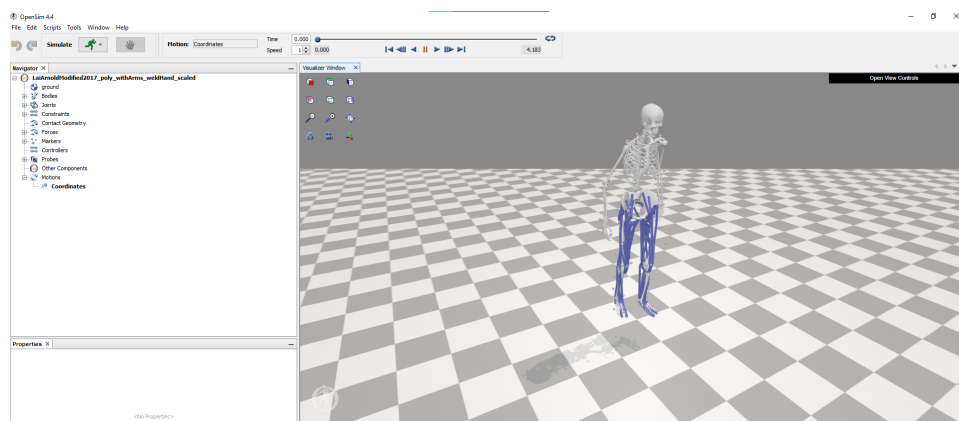


Figura 3.8. Carga de movimiento – OpenSim.

7. Pulsar en la opción *play* del reproductor para ver la ejecución del gesto. A partir de esta grabación se obtuvo una gráfica de elongación del músculo rectofemoral para la pierna izquierda del participante, como se ve en la Figura 3.9.

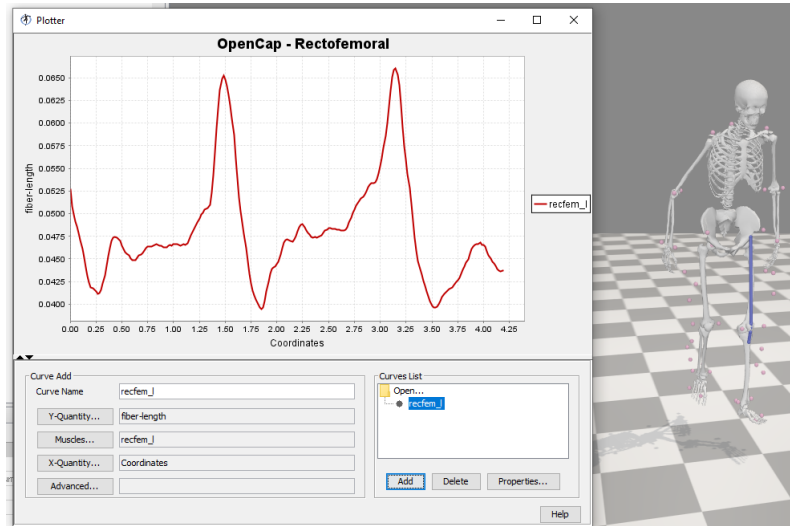


Figura 3.9. Elongación músculo recto femoral – OpenSim.

De esta manera, se puede obtener gráficas similares para diferentes músculos en función de la captura de movimiento realizada.

La plataforma OpenCap posee un tablero de análisis que también genera gráficas en función de los valores que el usuario desee (Figura 3.10). En el caso de no disponer de OpenSim se puede hacer uso de la misma para tener valoraciones preliminares.

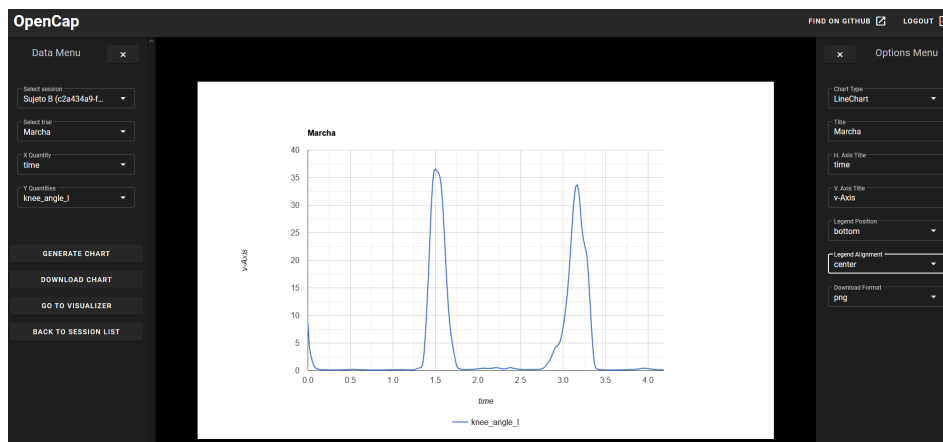


Figura 3.10. Análisis – OpenCap.

La validez de los resultados obtenidos en la plataforma OpenCap se han comentado en Uhrich et al. (2022).

## **4. CONCLUSIONES Y RECOMENDACIONES**

### **4.1. Conclusiones**

- Se adaptó un sistema de captura de movimiento para el Laboratorio de Biomecánica de la Facultad de Ingeniería Mecánica de la Escuela Politécnica Nacional el cual no depende exclusivamente de cámaras optoelectrónicas y de software de tipo privativo, sino de herramientas de software libre y cámaras de bajo costo.
- Se realizaron pruebas en 3 diferentes equipos, mostrando la aplicabilidad únicamente en dos de ellos gracias a las características de los procesadores y tarjetas de video.
- Se comparó distintos softwares libres de captura de movimiento seleccionando OpenCap como el más adecuado, por la versatilidad y portabilidad del mismo.
- Se evidencia el uso de DLTdv, sin embargo se deben hacer correcciones en la forma de ajuste de parámetros de cámaras con el fin de obtener calibraciones que sean útiles para generar datos exportables. Corrigiendo esto, el programa se presenta como una alternativa fuerte de análisis gracias al uso de la herramienta de Deep Learning que viene incorporada.
- Se generó un proceso de calibración y captura de información para las cámaras para al menos dos de los softwares analizados.
- Se recuperó datos que pueden ser usados en análisis biomecánicos con el programa OpenCap, estos se validaron en OpenSim 4.4.

### **4.2. Recomendaciones**

- Instalar software libre en los equipos del laboratorio para poner a prueba la capacidad de los mismos en el análisis y captura de movimiento.
- Hacer pruebas con más dispositivos de captura de video con el fin de obtener mejor calidad en los gestos que se han grabado.
- Probar una configuración de cámaras distinta a la sugerida en el trabajo para la adquisición de datos de video sin oclusiones.
- Configurar los dispositivos de captura de video en base del programa seleccionado para usar imágenes a color o blanco y negro.
- Crear una biblioteca de videos de gestos para poder disponer de la misma en el entrenamiento de Deep Learning para DLTdv8.
- Probar librerías de software de entrenamiento en captura de movimiento (DeepLabCut), en el Laboratorio con el fin de analizar movimientos en otros seres vivos.

## BIBLIOGRAFÍA

- Anrrango, E., y A., J. (2019). *Sistema de monitoreo del cuerpo humano mediante el uso de una cámara de profundidad*. Universidad Técnica del Norte.
- Burke, J., Morrow, P. J., McNeill, M., McDonough, S. M., y Charles, D. K. (2008). *Vision based games for upper-limb stroke rehabilitation*. Machine Vision and Image Processing Conference.
- Chen, C., Jafari, R., y Kehtarnavaz, N. (2017). A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools and Applications*, 76(3), 4405–4425.
- Colbrum, R. (2022). *Simtk*. ([https://simtk.org/projects/motion\\_analyst](https://simtk.org/projects/motion_analyst))
- Corazza, S., Gambaretto, E., Munderman, L., y Andriacchi, T. (2010). Automatic generation of a subject-specific model for accurate markerless motion capture and biomechanical applications. *IEEE Transactions on Biomedical Engineering*, 57(4), 806–812.
- Cruz, R., y A., C. (2018). *Fiabilidad de la técnica de video-análisis en el análisis cinemático de la flexo-extensión del cuello*. Escuela Politécnica Nacional.
- Estevez-García, R., Martín-Gutierrez, J., Mendoza, M., S., R. M., J., C.-M., P., S.-M., . . . Lodeiro-Santiago, M. (2015). Open data motion capture: MOCAP-ull database. *Procedia Computer Science*, 75, 316–326.
- Fang, H., y Xuesong, B. (2015). *Research and literature review on developing motion capture system for analyzing athletes action*. International Conference on Education Technology Management and Humanities Science (ETMHS 2015).
- FFmpeg. (2022, septiembre). *Fmpeg: A complete, cross-platform solution to record, convert and stream audio and video*. (<https://ffmpeg.org/>)
- Field, M., Stirling, D., y Naghdy, F. (2011). Human motion capture sensors and analysis in robotics. *Industrial Robot: An International Journal*, 38(2), 163–171.
- Flam, D. (2022). *Openmocap*. GitHub. (<https://github.com/davidflam/OpenMoCap>)
- Freire, B. (2019). *Evaluación de datos experimentales en un modelo opensim del comportamiento del movimiento de flexo extensión del cuello*. Escuela Politécnica Nacional.
- García, C. (2020). *Captura de movimiento del grupo afrodescendiente "Las Tres Marías*

*para la preservación cultural del patrimonio vivo del Ecuador*. Quito: Universidad de las Américas.




- Gómez-García, J. A., Godino-Llorente, J. I., y Castellanos-Domínguez, G. (2014). Non uniform embedding based on relevance analysis with reduced computational complexity: application to the detection of pathologies from biosignal. *Neurocomputing*, 132, 148–158.
- Hedrick, T. (2022). *Dltdv digitizing tool*. Hedrick Lab. (<https://biomech.web.unc.edu/dltdv/>)
- Hillman, M. (2003). *Rehabilitation robotics from past to present—a historical perspective*. Proceedings of the International Conference on Rehabilitation.
- Hondori, H. M., Khademi, M., Dodakian, L., Cramer, S. C., y Lopes, C. V. (2013). *A spatial augmented reality rehab system for post-stroke hand rehabilitation*. MMVR.
- Huang, M. H., y Brown, S. H. (2013). Age differences in the control of postural stability during reaching tasks. *Gait Amp; Posture*, 38, 837–842.
- Jin, M., Zhao, J., Jin, J., Yu, G., y Li, W. (2014). *The adaptive Kalman filter based on fuzzy logic for inertial motion capture system*.
- Kitagawa, M., y Windsor, B. (2008). *Mocap for artists : workflow and techniques for motion capture*. Burlington: Elsevier.
- Komaris, D. S., Govind, C., Clarke, J., Ewen, A., Jeldi, A., Murphy, A., y Riches, P. L. (2020). *Identifying car ingress movement strategies before and after total knee replacement*. Int. Biomech.
- Komaris, D. S., Perez-Valero, E., Jordan, L., Barton, J., Hennessy, L., y O’Flynn, O., B. (2019). *Predicting three-dimensional ground reaction forces in running by using artificial neural networks and lower body kinematics*. IEEE Access.
- Lamoth, C. J., y van Heuvelen, M. J. (2012). Sports activities are reflected in the local stability and regularity of body sway: older ice-skaters have better postural control than inactive elderly. *Gait Posture*, 35(3), 489–493.
- Lewcowicz, D. (2022). *Realtime motion capture toolbox (rtmocap)*. (<https://sites.google.com/site/rtmocap/>)
- Matthis, J., y Idehen, E. (2022). *Freemocap*. GitHub. (<https://github.com/freemocap/freemocapreadme>)
- Maza, M., y A., G. (2017). *Análisis del comportamiento del cuello en el efecto latigazo por medio de elementos finitos para obtener los puntos críticos de lesión por medio del análisis dinámico y de fuerzas generadas*. Escuela Politécnica Nacional.

- Menolotto, M., Komaris, D. S., Tedesco, S., O'Flynn, B., y Walsh, M. (2020). Motion capture technology in industrial applications: A systematic review. *Sensors*, 20(5687).
- Moeslund, T. B., Hilton, A., y Krüger, V. (2006). A survey of advances in vision-based human motion capture. *Computer Vision and Image Understanding*, 104, 90–126.
- Molina, Z., y J., J. (2017). *Análisis de video del comportamiento cinemático de la rodilla usando el método de eje finito helicoidal*. Quito: Escuela Politécnica Nacional.
- Moya, D., y Miguel, C. (2018). *Análisis biomecánico de la técnica de sentadilla en los fisicoculturistas de la Pontificia Universidad Católica del Ecuador (PUCE)*. Quito: Sede Quito en el año 2017.
- OptiTrack. (2022). *Optitrack camera sdk*. Autor. (<https://www.optitrack.com/software/camera-sdk/>)
- Pozo, S., y N., N. (2018). *Sistema de captura de movimiento para asistir el entrenamiento del club de taekwondo de la Universidad Técnica del Norte*. Universidad Técnica del Norte.
- Pridmore, T., Hilton, D., Green, J., Eastgate, R., y Cobb, S. (2004). *Mixed reality environments in stroke rehabilitation: interfaces across the real/virtual divide*. Proceedings of the 5th International Conference on Disability, Virtual Reality Associated Technology.
- Probar cámara. (2022, septiembre). (<https://es.webcamtests.com/>)
- Pérez, M. (2020). *Análisis postural del riesgo ergonómico inherente a la actividad del baño de neonatos desempeñada por enfermeras utilizando la metodología REBA a través de sensores de profundidad KINECT*. Quito: Escuela Politécnica Nacional.
- RTMoCap. (2022). *Real time motion capture toolbox rtmocap*. Autor. (<https://sites.google.com/site/rtmocap/home>)
- Speich, J., y Rosen, J. (2004). *Encyclopedia of Biomaterials and Biomedical Engineering : Medical robotics*. Marcel Dekker Inc.
- Studio, O. (2022). *OBS Studio*. Open Broadcaster Software. (<https://obsproject.com/es/download>)
- Theriault, D. H., Fuller, N. W., y Jackson, B. E. (2014). A protocol and calibration method for accurate multi-camera field videography. *Journal of Experimental Biology*, 217(11), 1843-1848.
- Uhlrich, S., Falisse, A., Kidzinski, L., Muccini, J., Ko, M., y Chaudhari, D. S., A. (2022). *Opencap:3d human movement dynamics from smartphone videos*. bioRxiv.

- Wang, L. (2006). *Abnormal walking gait analysis using silhouette-masked flow histograms ICPR 2006*. 18th International Conference on Pattern Recognition IEEE.
- Wang, L., Cheng, L., y Zhao, G. (2010). *Machine learning for human motion analysis: theory and practice*. Hershey PA: Medical Information Science Reference.
- Yansaguano, L., y B., F. (2020). *Sistema electrónico de corrección del gesto deportivo en base a la biomecánica en deportistas de marcha atlética*. Universidad Técnica de Ambato.
- Zambrano, F. (2020). *Estudio ergonómico en enfermeras que realizan la tarea de alimentación a infantes neonatos, mediante la validación, adquisición y procesamiento de datos capturados por sensores de profundidad*. Quito: Escuela Politécnica Nacional.
- Zhou, H., y Hu, H. (2008). Human motion tracking for rehabilitation: a survey. *Biomedical Signal Processing and Control*, 3(1), 1–18.

## ANEXOS

### I. Listado de software de captura de movimiento

Software	Enlace
OpenMocap	<a href="https://code.google.com/archive/p/openmocap/">https://code.google.com/archive/p/openmocap/</a> <a href="https://github.com/davidflam/OpenMoCap">https://github.com/davidflam/OpenMoCap</a> 
RTMoCap	<a href="https://sites.google.com/site/rtmocap/home">https://sites.google.com/site/rtmocap/home</a> 
OptiTrack (Camera SDK)	<a href="https://optitrack.com/software/camera-sdk/">https://optitrack.com/software/camera-sdk/</a> 



---

**Software****Enlace**

---

[https://simtk.org/projects/motion\\_analyst](https://simtk.org/projects/motion_analyst)

Motion Analysis



---

<https://freemocap.org/>

Freemocap



---

<http://motion.capture.free.fr/index.html>

Cheap Mocap



---

<https://biomech.web.unc.edu/dltdv/>

DLTdv digitizing tool



---

**Software****Enlace**

---

<https://anipose.readthedocs.io/en/latest/#>

Anipose



---

<https://github.com/CMU-Perceptual-Computing-Lab/openpose>

Openpose



---

<https://www.opencap.ai/>

OpenCap



---

<https://obsproject.com/es/download>

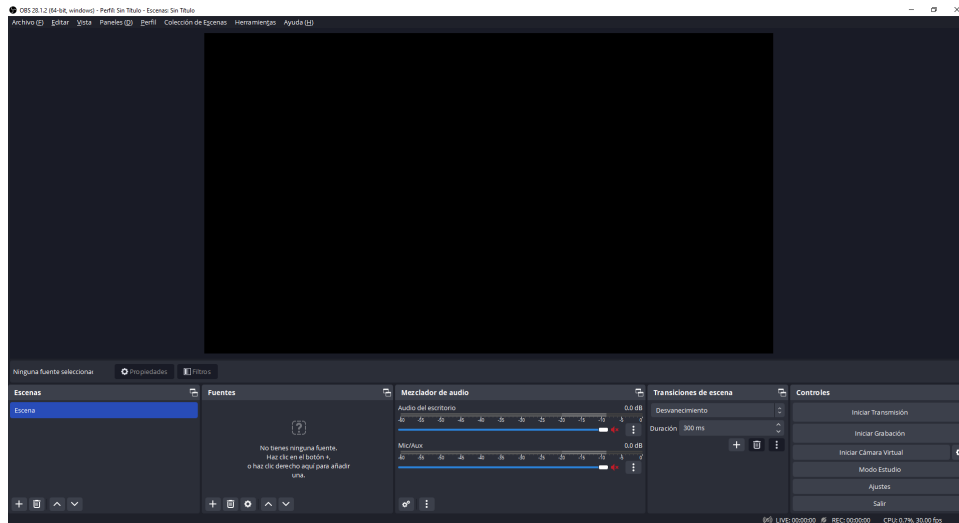
OBS Studio



## II. Configuración de OBS Studio

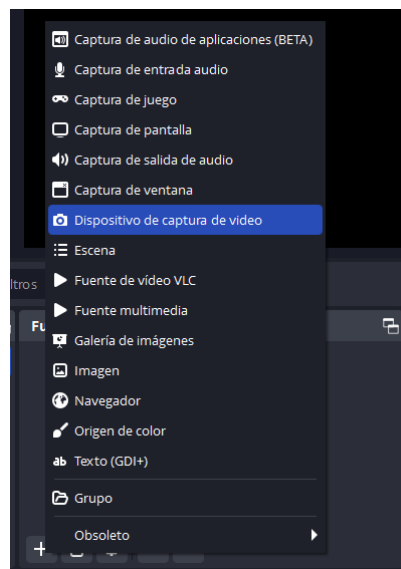
### a. Configuración de cámaras

1. Una vez que las cámaras web estén conectadas al equipo, pulsar en el ícono de OBS Studio para abrir la aplicación.
2. En la pantalla de inicio del programa, verificar que las entradas de sonido estén silenciadas para que de esta manera solo se grabe el video.



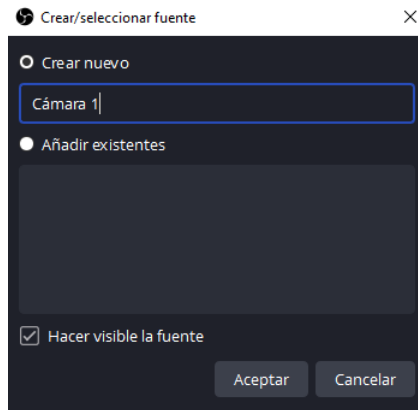
Pantalla principal – OBS Studio.

3. En la sección *fuentes* pulsar en el signo + y en el siguiente menú seleccionar la opción *Dispositivo de captura de video*.



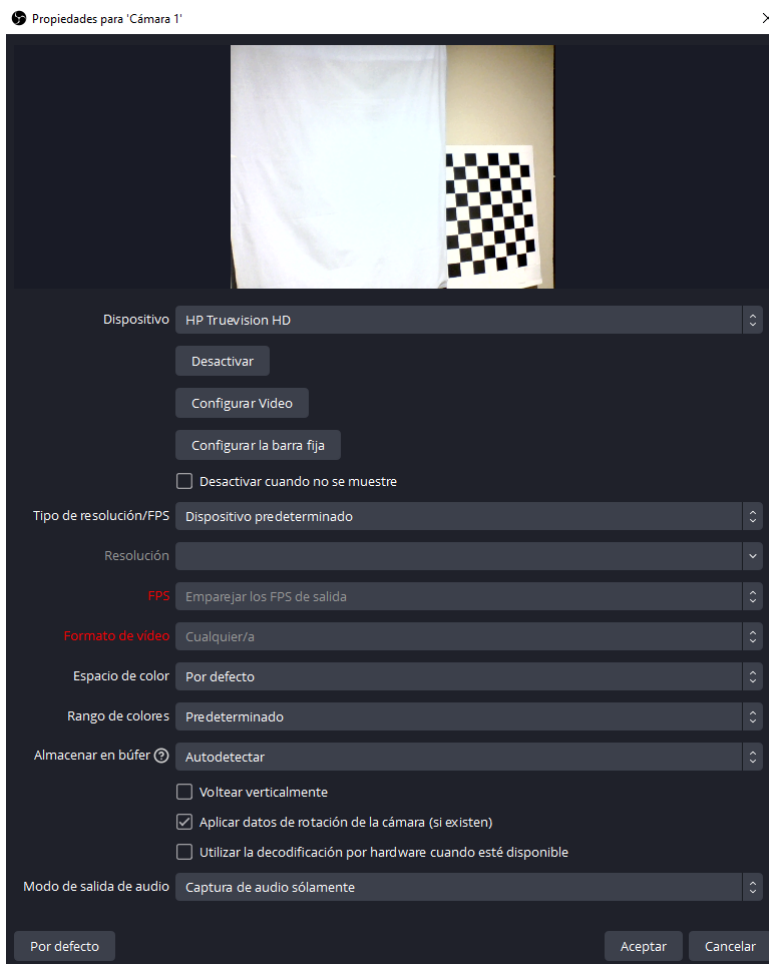
Selección de dispositivo de captura – OBS Studio.

4. En la nueva ventana que aparece, colocar el nombre adecuado para una de las cámaras web que se va a usar y pulsar en *Aceptar*.



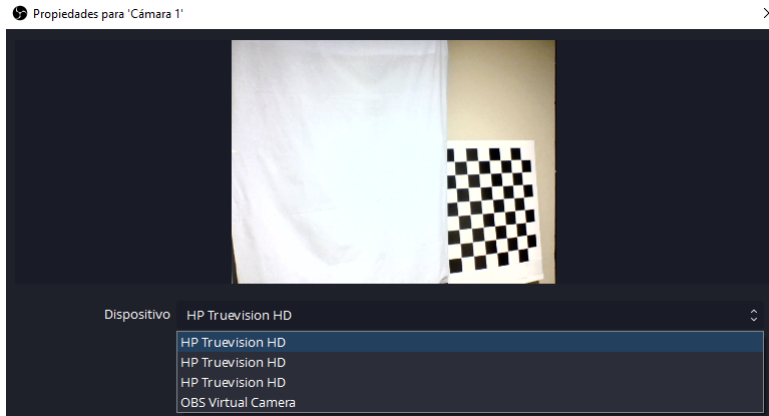
Creación de fuentes – OBS Studio.

5. El nuevo menú presentará las opciones de configuración del dispositivo.



Configuración del dispositivo – OBS Studio.

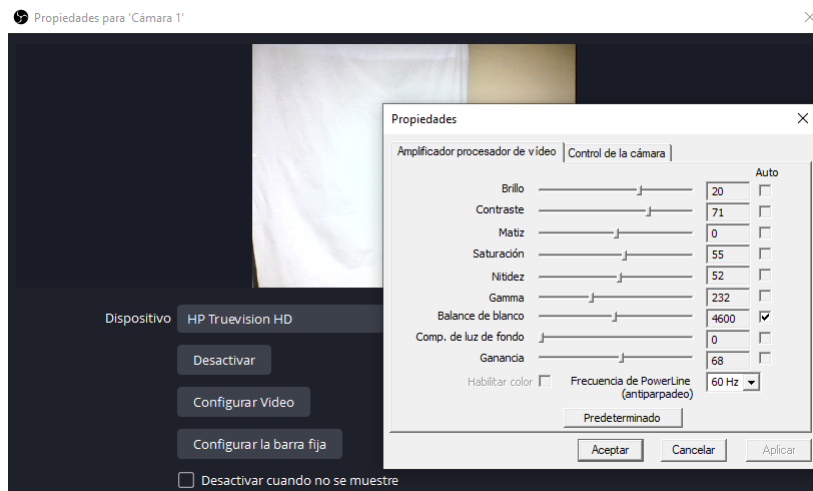
- La opción *Dispositivo* permite seleccionar a una de las cámaras web que se tiene conectadas al equipo. Si las cámaras son de la misma marca, esto se verá reflejado en el menú de selección.



Selección del dispositivo – OBS Studio.

Para poder saber qué cámara se seleccionará, se debe pulsar en el nombre de las opciones disponibles y mirar la vista preliminar que se presenta para conocer a que dispositivo se refiere el programa.

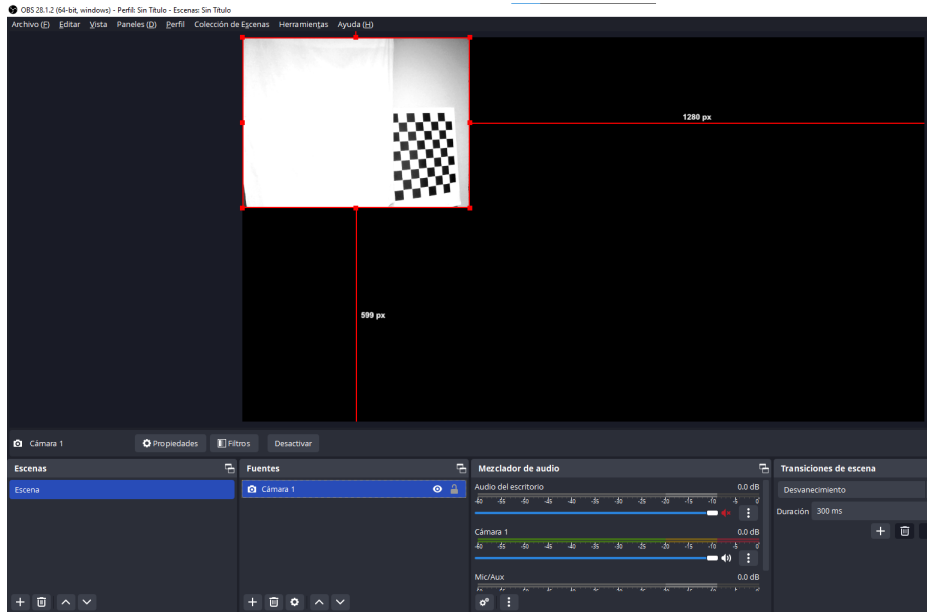
- La opción *Configurar video*, permite ajustar los valores de brillo, contraste, matiz, saturación, nitidez, gamma, balance de blanco, composición de luz de fondo y ganancia.



Configuración de video – OBS Studio.

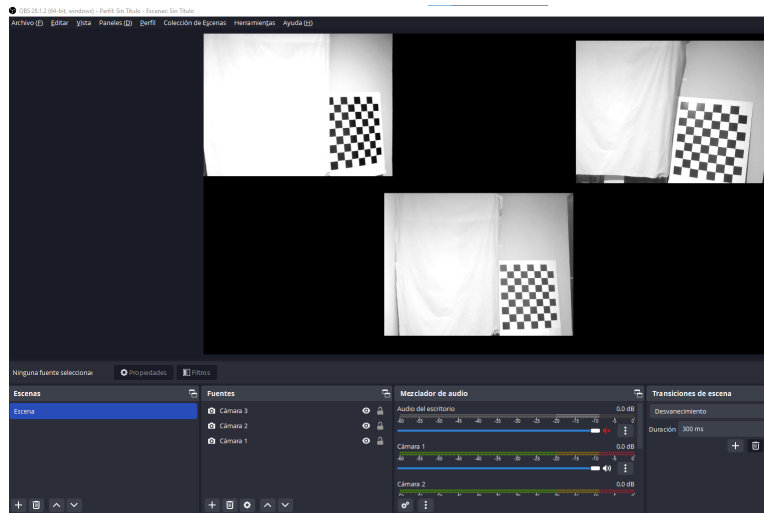
Se recomienda usar los valores de: Brillo = 30, Contraste = 71, Nitidez = 52, Saturación = 0. Se puede alterar estos valores hasta obtener una toma clara de la sección a grabarse.

6. Terminada la configuración de una cámara se la coloca en la parte de la pantalla negra que se desee, para tener control del área de grabación.



Colocación de video – OBS Studio.

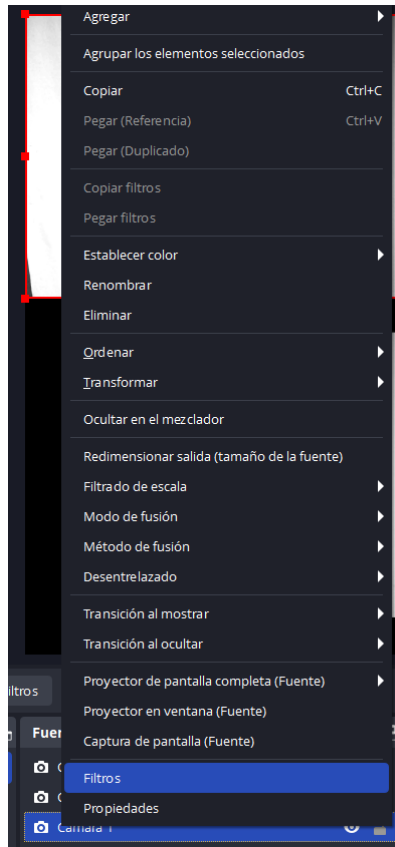
7. Repetir del paso 3 al 6 para los demás dispositivos.
8. Una vez realizada la configuración de 3 cámaras, se las coloca en la pantalla principal para poder hacer ajustes posteriores.



Ubicación de dispositivos de video – OBS Studio.

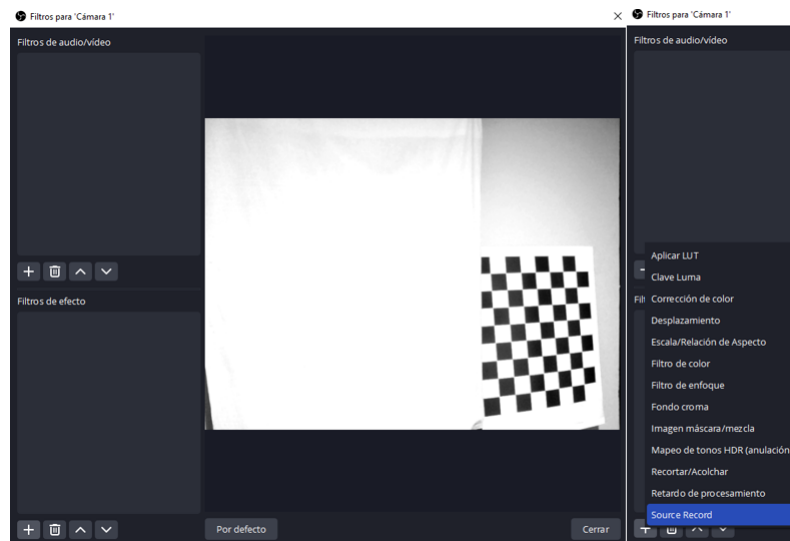
## b. Grabación de video

1. Dar clic derecho sobre una de las fuentes (cámara 1, 2 o 3) y seleccionar la opción filtros.



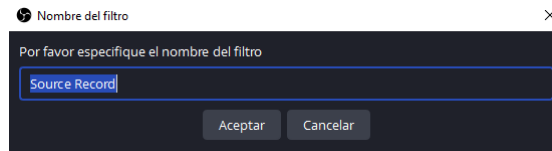
Selección de filtros – OBS Studio.

2. En la siguiente ventana pulsar en el signo + de filtros de efecto y después en la opción *Source Record*.



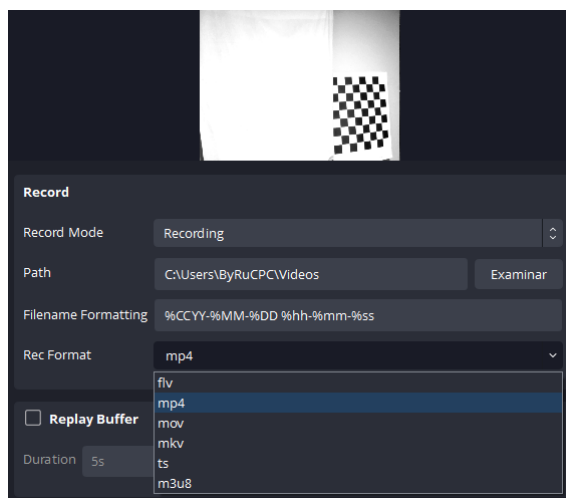
Selección de Source Record – OBS Studio.

3. Mantener el nombre del filtro en la siguiente ventana que aparece y pulsar en *Aceptar*.



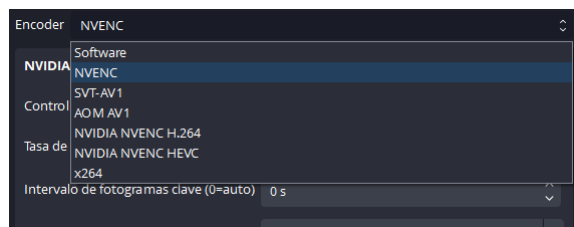
Especificación del nombre del filtro – OBS Studio.

4. En la siguiente ventana en la sección *Record*:
  - En *Record Mode*, escoger la opción *recording*.
  - Seleccionar la ruta donde se guardará el video de este dispositivo en la opción *Path*.
  - En la opción *Rec Format* seleccionar la opción *mp4*.



Selección del tipo de archivo – OBS Studio.

5. En esta misma ventana en la sección *Encoder* se selecciona la tarjeta gráfica que posea el computador para administrar la grabación. Si no se dispone de la misma se mantiene la opción por defecto que el programa reconozca. Se pulsa en la opción *cerrar*.

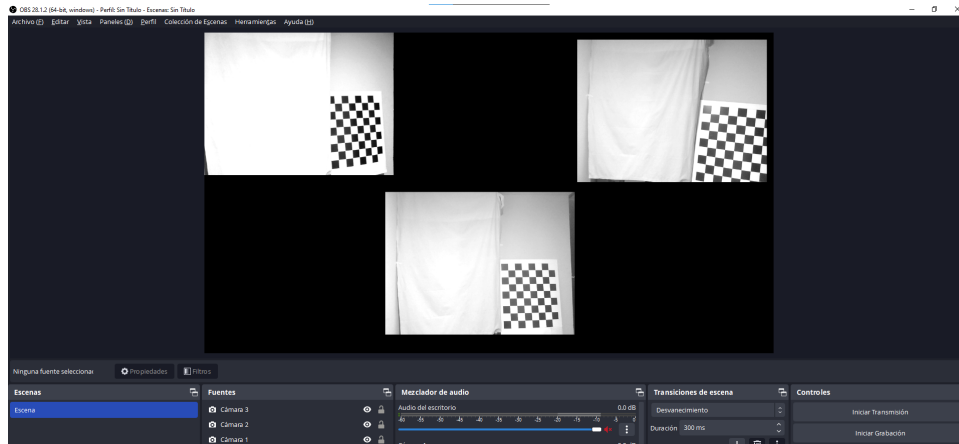


Selección de tarjeta de video – OBS Studio.

6. Se repite los pasos del 1 al 5 para cada uno de los dispositivos.
7. En la pantalla principal se pulsa en la opción *iniciar grabación* para iniciar el procedimiento de grabación del video. Este botón cambiará a la opción *Detener la grabación* una vez que comience el proceso, se debe pulsar en el mismo para



terminar el proceso.

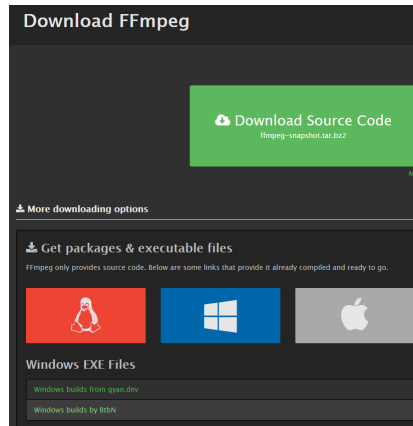


Inicio de grabación – OBS Studio.

8. Se crearán cuatro archivos, uno por cada dispositivo y uno de la toma general con las 3 cámaras. Verificar que se hayan creado los archivos en las rutas seleccionadas.

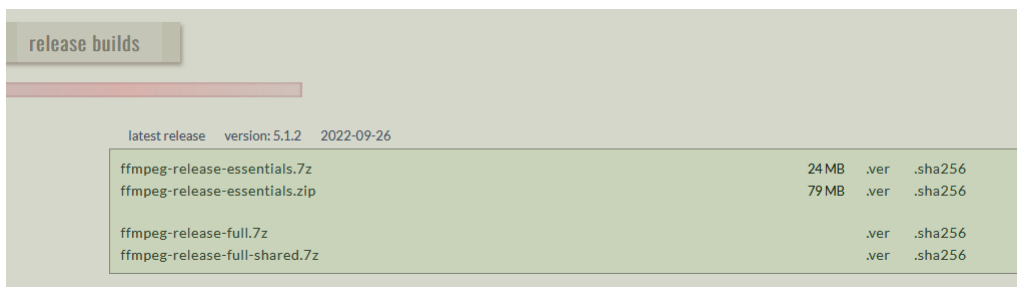
### III. Instalación de FFmpeg

1. Descargar el software de la página: <https://ffmpeg.org/download.html>.  
Seleccionar la opción *Windows builds from gyan.dev*.



Pantalla de descarga – FFmpeg.

2. En la nueva página, buscar la opción *release builds* y descargar *ffmpeg-release-full.7z*.



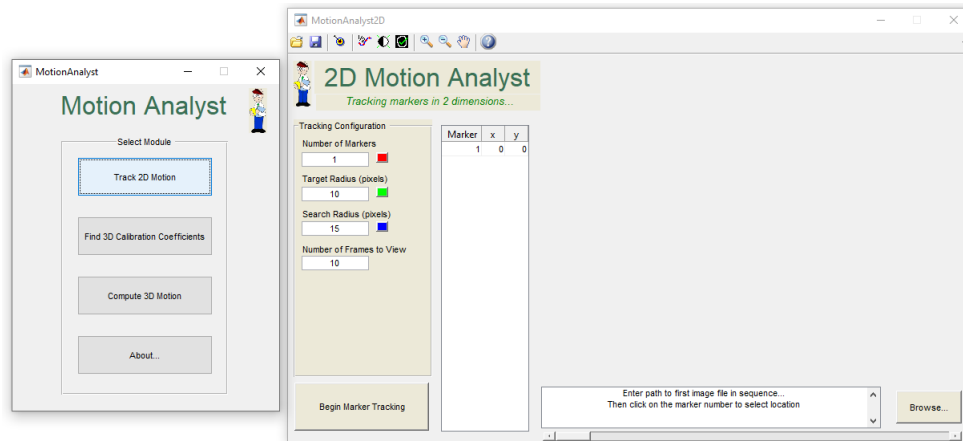
Repositorio – FFmpeg.

3. Extraer el archivo descargado.
  - Buscar el archivo en la carpeta donde se descargó.
  - Dar clic derecho en el nombre del archivo.
  - Seleccionar la opción *Extraer aquí* y esperar la extracción de los archivos. Esto creará una carpeta con el mismo nombre del archivo comprimido.
4. Renombrar la carpeta extraída con el nombre FFmpeg.
5. Cortar la carpeta renombrada y pegarla en la unidad principal del equipo (usualmente llamada "Windows (C:)" o "Disco Local (C:)").
6. Abrir el panel de control de entornos de variable.
  - Presionar la combinación Tecla de Windows + s para abrir la barra de búsqueda.
  - Escribir variables de entorno del sistema en la barra de búsqueda.
  - Pulsar en la opción Editar las variables del entorno del sistema.

- En la siguiente ventana pulsar en la opción Variables de entorno.
- 7. Seleccionar la opción *Path* en el campo “Variables de usuario para (nombre de usuario)” y dar clic en *Editar*. Un listado de variables aparecerá.
- 8. En la nueva ventana pulsar en la opción *Explorar* e ir al directorio en el cual se encuentra la carpeta, abrirlo y pulsar en la carpeta bin. La dirección quedará de la siguiente manera: C:\ffmpeg\bin. Pulsar en *Aceptar* y salir.
- 9. Una vez más pulsar en *Aceptar* y salir de la ventana.

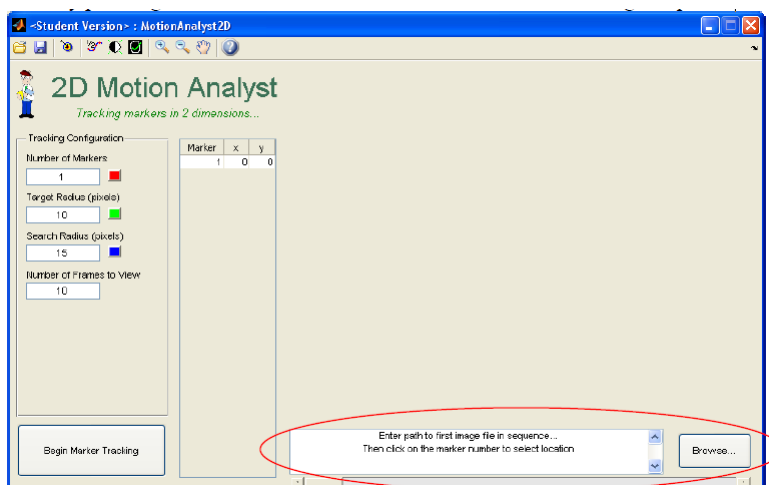
## IV. Análisis con Motion Analyst

1. Abrir MATLAB y colocar la carpeta *MotionAnalyst\_Version 1.6* en la ruta principal para poder ejecutar los archivos
2. Ejecutar el archivo *MotionAnalyst2D.m*



Interface principal – Motion Analyst.

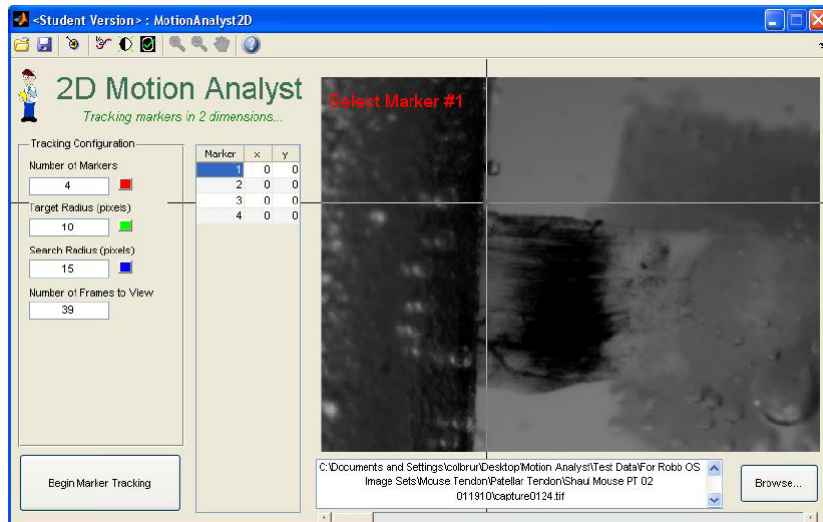
3. Comenzar pulsando el botón de búsqueda para seleccionar el primer archivo de la secuencia de imágenes.



Selección de archivo de carga – Motion Analyst.

Esto cargará el archivo y mostrará la ruta del mismo.

4. Una vez cargada la imagen, introducir el número de marcadores a rastrear en el área de Configuración de Rastreo. Esto expandirá la tabla para mostrar las coordenadas x, y de todos los marcadores.

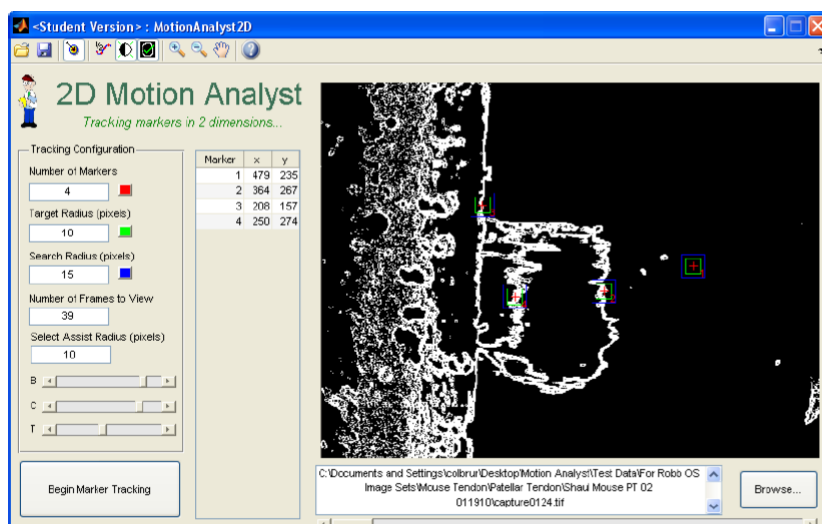


Introducción de marcadores – Motion Analyst

Hacer clic en el número del marcador en la tabla y aparecerá un aviso, junto con cruces, para ubicarlas, hacer clic en la imagen donde se colocará ese marcador en particular. Los valores de los píxeles x, y de este marcador aparecerán en la tabla. Repetir la operación para todos los marcadores. El software incrementará automáticamente cada marcador de la lista por encima del marcador seleccionado como punto de partida.

Los controles de zoom y panorámica de la esquina superior izquierda pueden utilizarse para ampliar la imagen y verificar que la selección de marcadores es la deseada.

5. La sección de configuración de seguimiento de la pantalla tiene varios parámetros. Hay selectores de color que permiten modificar los colores para que sean lo más visibles posible.



Configuración de rastreo – Motion Analyst.

- *Number of markers*: Número de marcadores que el programa espera rastrear.
- *Target Radius*: Es el tamaño de la región en la que se encuentra el marcador (o patrón) rastreado.
- *Search Radius*: Su función es similar a la de Target Radius, excepto que define el área de la imagen donde se buscará el patrón de píxeles del área objetivo de la imagen anterior.
- *Number of Frames to View*: Cuando se ejecuta el software de seguimiento, las imágenes de la secuencia se mostrarán en un nuevo gráfico y mostrarán un punto blanco donde se ha calculado el centro del marcador. El número de fotogramas para ver cerrará este nuevo gráfico después del número de fotogramas especificado. Tenga en cuenta que el seguimiento se producirá en todos los fotogramas de la carpeta independientemente de este parámetro.

6. Hacer clic en *Begin Marker Tracking*. Se creará automáticamente un archivo de control (denominado 'archivo de control.csv') con los parámetros existentes y se colocará en la carpeta donde se encuentran todas las imágenes.

A continuación, se iniciará el rastreo y aparecerá una ventana que mostrará el rastreo. Los marcadores se muestran como puntos verdes o rojos en la imagen y esto permite al usuario confirmar visualmente lo bien que se está rastreando. Los puntos verdes corresponden a que todos los marcadores se han movido de su posición anterior. Los puntos rojos corresponden a un fotograma en el que al menos uno de los marcadores no se ha movido y el algoritmo de movimiento sub-píxel está activado.

7. El progreso del rastreo (junto con cualquier error) se mostrará en el espacio de trabajo de MATLAB, así que asegúrese de revisar allí para obtener información. Una vez completado el rastreo, se creará un archivo de resultados en formato .csv y .mat.

## V. Instalación de FreeMoCap

1. Descargar e instalar Anaconda 3.9 (<https://repo.anaconda.com/archive/Anaconda3-2022.10-Windows-x86.64.exe>)
2. Abrir una ventana de símbolo del sistema o powershell para Anaconda e introducir los siguientes comandos:
  - a) Dentro del símbolo del sistema, ubicarse en el directorio principal del sistema (C:)
  - b) Crear un entorno Python3.7 Anaconda:

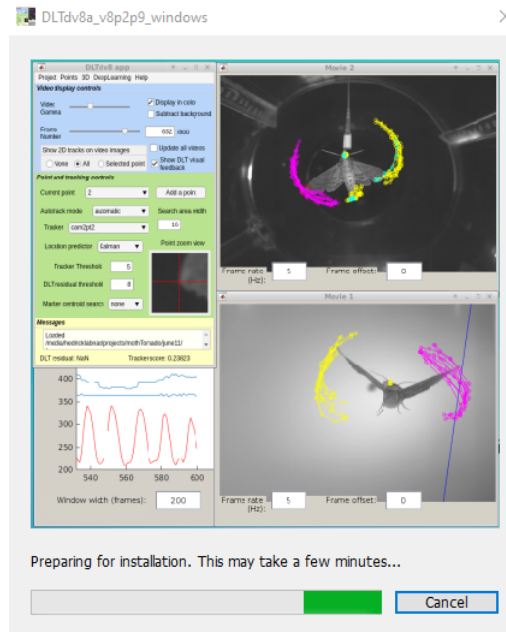
```
conda create -n freemocap-env python=3.7
```
  - c) Activar ese entorno recién creado:

```
conda activate freemocap-env
```
  - d) Instalar freemocap (versión 0.0.54) desde PyPi usando pip

```
pip install freemocap==0.0.54
```

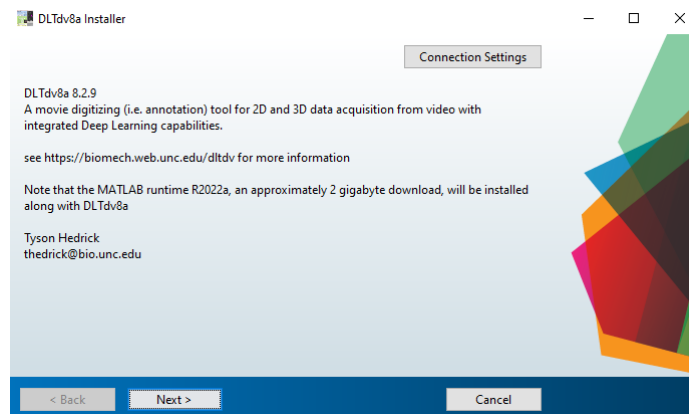
## VI. Instalación de DLTdv8a

1. Se descarga el archivo DLTdv8a\_v8p2p9\_windows.exe de la página oficial del proyecto (<https://biomech.web.unc.edu/dltdv/>).
2. En la carpeta de descarga se pulsa doble clic para iniciar la instalación.



Pantalla de inicio de instalación – DLTdv8a.

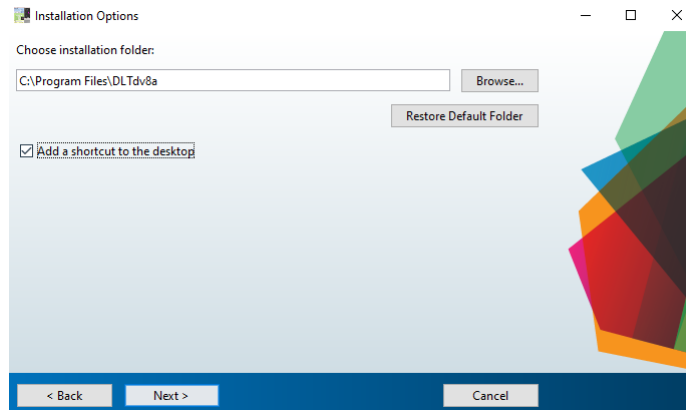
3. En la ventana que se abre se pulsa en *Next*.



Pantalla de detalles del programa – DLTdv8a.

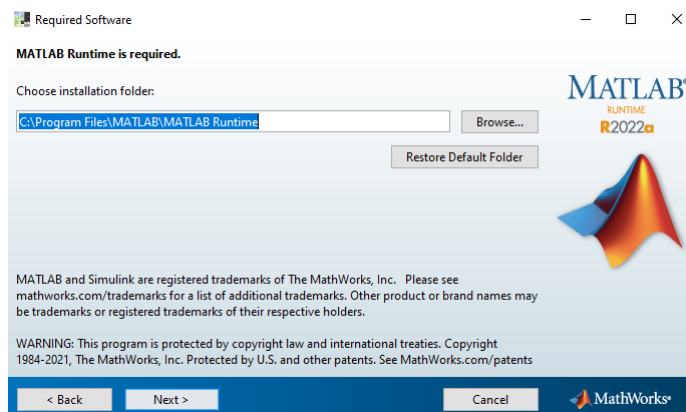
4. En la siguiente ventana se especifica la carpeta en la cual se instalará el programa. Se marca la opción *add a shortcut to the desktop* y confirmada la ruta se pulsa en *Next*.





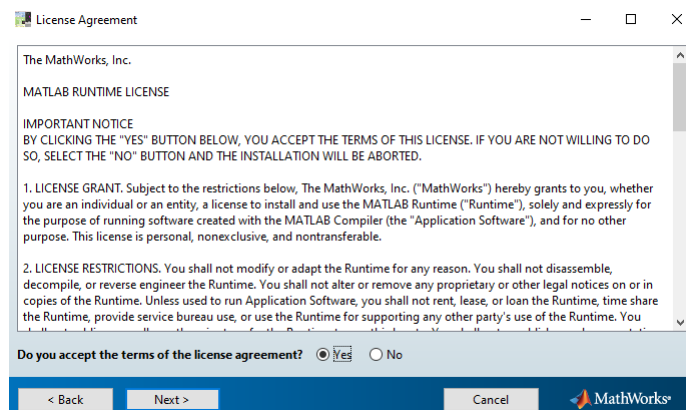
Ruta de instalación – DLTdv8a.

5. Se mantiene la ruta para instalar el complemento para MATLAB y se pulsa en *Next*.



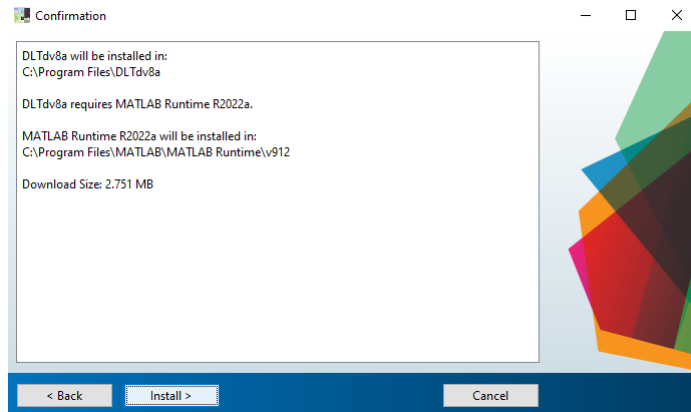
Ruta del complemento de MATLAB– DLTdv8a.

6. Aceptar las condiciones de la licencia marcando la opción *Yes* y pulsar en *Next*



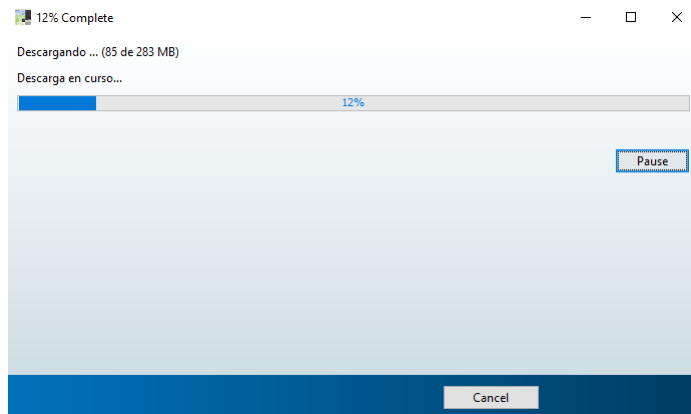
Acuerdo de licencia – DLTdv8a.

7. Revisar las rutas de instalación y pulsar en *Install*.



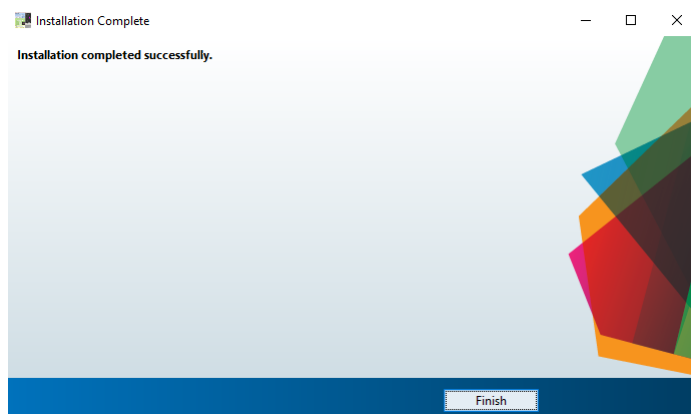
Revisión de rutas de instalación – DLTdv8a.

8. Esperar al progreso de la descarga e instalación.



Progreso de la instalación – DLTdv8a.

9. Pulsar en *Finish* una vez se ha acabado el proceso.



DLTdv8a.

## VII. Instalación de easyWand

1. Descargar easyWand de la página: <https://biomech.web.unc.edu/wand-calibration-tools/>
2. Descomprimir el archivo descargado (Theriaux\_et\_al\_2014\_v2021-02-11.zip).
3. Abrir MATLAB.
4. Establecer la carpeta EasyWand en el árbol de directorio principal de MATLAB y añadir la carpeta a la ruta principal para poder ejecutar.
5. Dentro de la carpeta easyWand5, dar doble clic sobre el archivo easyWand5.m y ejecutarlo con la opción *Run*.

## VIII. Scripts de calibración de tablero de ajedrez

### a. Lector tablero

```
1 % Script to detect checkerboard data with mp4 archives
2 % Ty Hedrick 2021-01-27
3 % Byron Freire 2022-07-09
4
5 %Seleccion del video a ser analizado
6 %Colocar el nombre del archivo y la extension
7 v=VideoReader('NombreDelArchivo.mp4');
8 %Seleccion del frame del video para verificar el tablero
9 frame=read(v,3);
10 imagesc(frame);
11 %Deteccion del tablero en el frame seleccionado
12 I=detectCheckerboardPoints(frame);
13 whos I; %Informacion sobre el tama o de la cuadrícula
14 %Creacion de la cuadrícula con los puntos detectados
15 hold on
16 plot(I(:,1),I(:,2),'ro')
17 %Creacion del punto inferior izquierdo
18 plot(I(1,1),I(1,2),'ro','markerfacecolor','r')
19 %Creacion del punto superior derecho
20 plot(I(end,1),I(end,2),'ro','markerfacecolor','b')
21 %Informacion del video
22 InfoWidth = v.Width;
23 InfoHeight = v.Height;
24 InfoNumFrames = v.NumFrames;
25 InfoFrameRate = v.FrameRate; %frames por segundo
```

### b. Bastón de calibración

```
1 % script to collect checkerboard calibration
2 % data with mp4 archives
3 % for three cameras
4 % Ty Hedrick 2021-01-27
```

```

5 % Byron Freire 2022-07-09
6
7 clear
8
9 % check for the right MATLAB toolbox
10 if exist('detectCheckerboardPoints')~=2
11     disp('Sorry, you need the detectCheckerboardPoints function')
12     disp('from the Computer Vision toolbox to run this script')
13     return
14 end
15
16 % set checkerboard file names
17 % insertar los nombres de los videos generados
18 % por cada camara
19 cam1=VideoReader('VideoCamara1.mp4');
20 cam2=VideoReader('VideoCamara2.mp4');
21 cam3=VideoReader('VideoCamara3.mp4');
22
23 % expected checkerboard size
24 % Colocar el primer valor del WHOS (size)
25 % de LectorTablero.m
26 expectedSize=56;
27
28 % number of video frames
29 % Colocar el valor de InfoNumFrames de LectorTablero.m
30 numFrames=492;
31
32 % process camera 1
33 cldata={};
34 for i=1:3:numFrames
35 % look at every 3rd frame to keep the amount
36 % of data manageable
37     I=detectCheckerboardPoints(read(cam1,i));

```

```

38  if size(I,1)==expectedSize
39      c1data{i,1}=I;
40  else
41      c1data{i,1}=NaN(56,2);
42  end
43  disp(i);
44 end
45
46 % process camera 2
47 c2data={};
48 for i=1:3:numFrames
49 % look at every 3rd frame to keep the amount
50 % of data manageable
51  I=detectCheckerboardPoints(read(cam2,i));
52  if size(I,1)==expectedSize
53      c2data{i,1}=I;
54  else
55      c2data{i,1}=NaN(56,2);
56  end
57  disp(i);
58 end
59
60 % process camera 3
61 c3data={};
62 for i=1:3:numFrames
63 % look at every 3rd frame to keep the amount
64 % of data manageable
65  I=detectCheckerboardPoints(read(cam3,i));
66  if size(I,1)==expectedSize
67      c3data{i,1}=I;
68  else
69      c3data{i,1}=NaN(56,2);
70  end

```

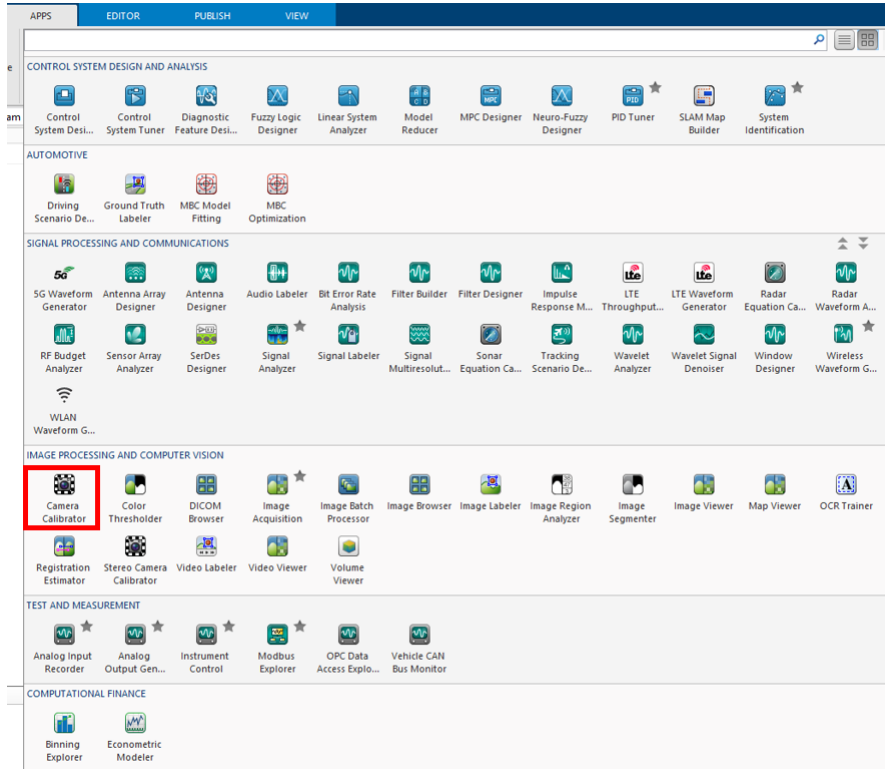
```

71 disp(i);
72 end
73
74 % let's build a wand - an array of paired points
75 % that should have a constant
76 % length regardless of their position
77 % in the calibrated volume
78 wand=[];
79 for i=1:3:numFrames
80 wand=[wand;[c1data{i}(1,:),c2data{i}(1,:),c3data{i}(1,:)...
81           ,c1data{i}(end,:),c2data{i}(end,:),c3data{i}(end,:)]];
82 end
83
84 % let's build the background points array
85 % - just all the points we collected
86 % stacked together with a cat()
87 bkg=[cat(1,c1data{:}),cat(1,c2data{:}),cat(1,c3data{:})];
88
89 % write out the paired and unpaired points files for
90 %easyWand / simpleWand / Argus::wand
91 dlmwrite('bkgmp4_3cam.csv',bkg);
92 dlmwrite('wandmp4_3cam.csv',wand);

```

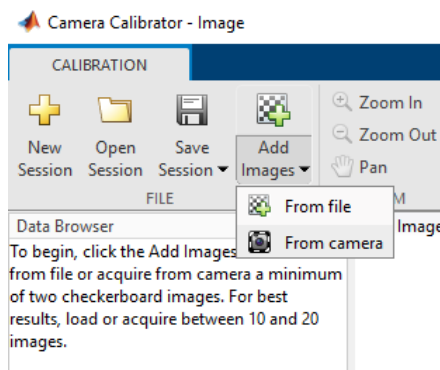
## IX. Uso de Camera Calibrator

1. Conectar solo una de las cámaras a usarse.
2. Colocar la cartilla de calibración 2 en un sitio fijo cercano a la cámara.
3. Abrir *MATLAB*.
4. En la sección *APPS*, pulsar sobre la opción *Camera calibrator*.



Interfaz de selección – Camera Calibrator.

5. En el nuevo menú, pulsar sobre la opción *Add images* y después en la opción *From camera*.

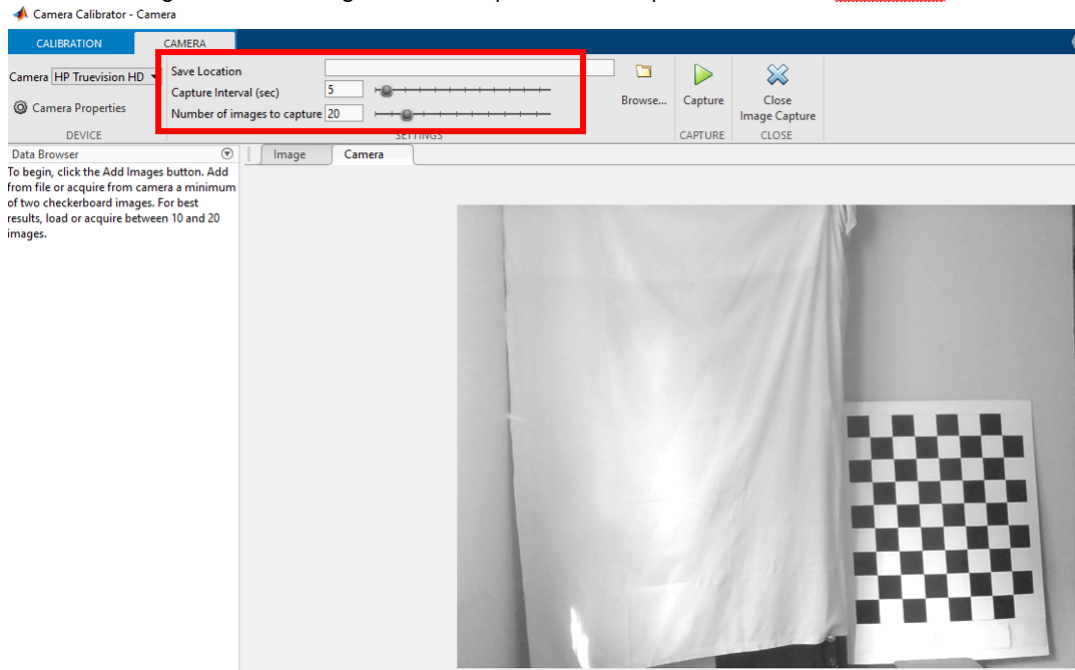


Selección de dispositivo de captura– Camera Calibrator.

6. Una vez que se active la opción *CAMERA*, personalizar las opciones:
  - *Save location*: Escoge la ruta en la cual se almacenarán los archivos de captura.

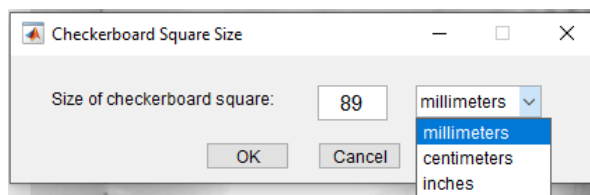


- *Capture Interval (sec)*: Se selecciona el intervalo entre tomas de captura.
- *Numbers of images to capture*: Especifica la cantidad de tomas de captura a realizarse.



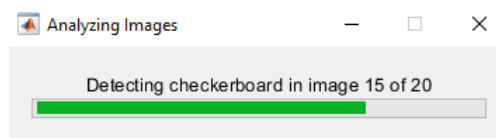
Configuración de opciones de captura – Camera Calibrator.

7. Con las opciones ya personalizadas, pulsar en el botón *Capture* y comenzar la grabación. Si se desea, se puede mover la cartilla de calibración dentro del área visual de la cámara.
8. Una vez que la captura ha concluido, aparecerá una ventana en la cual se solicita ingresar el tamaño de los cuadrados que forman el tablero, ingresar el tamaño que se ha usado en la unidad adecuada y pulsar en *OK*.



Tamaño de cuadrados del tablero – Camera Calibrator.

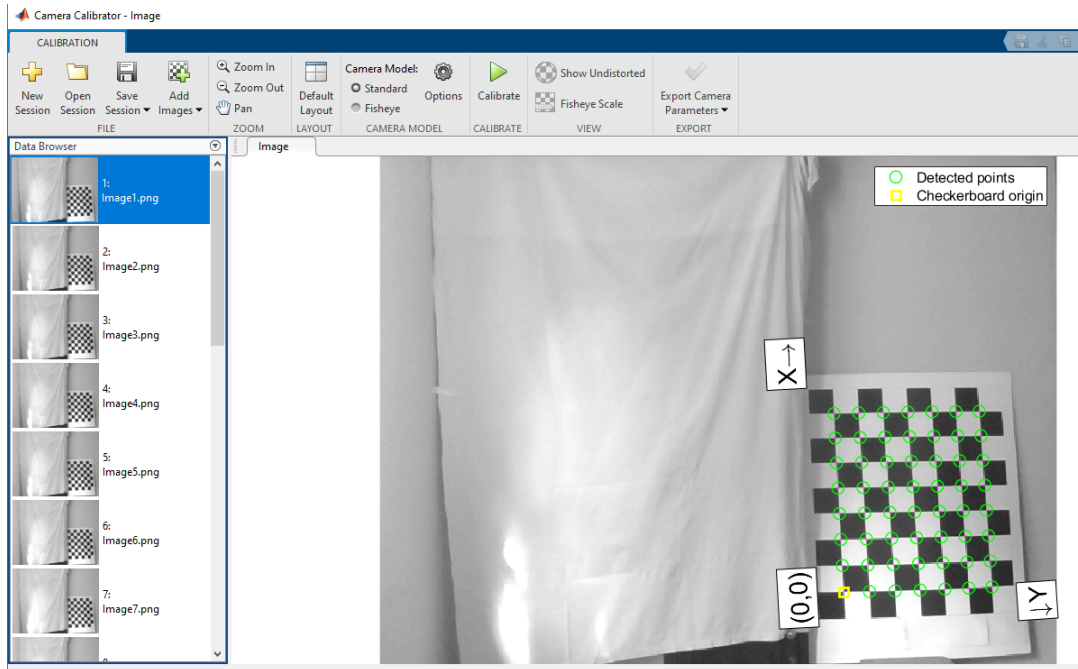
9. Se comenzará el reconocimiento en las imágenes capturadas.



Reconocimiento de imágenes – Camera Calibrator.

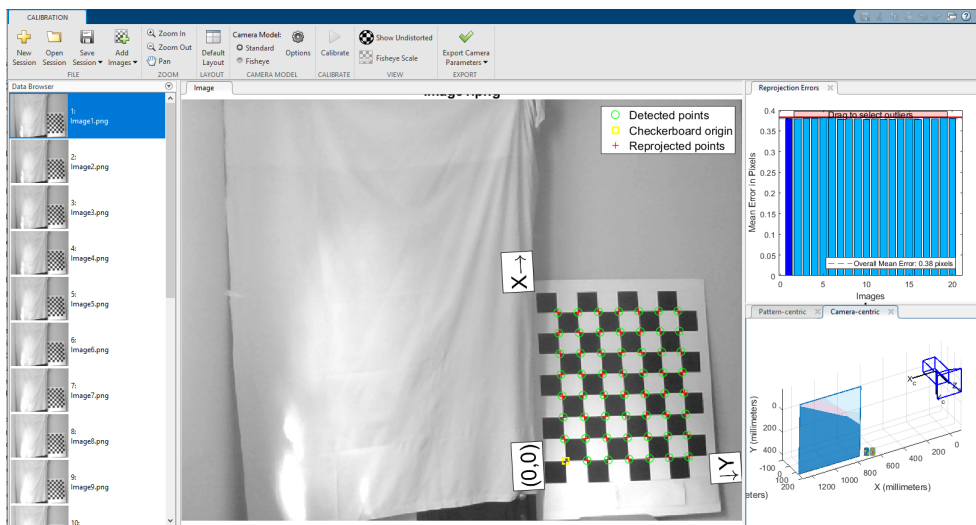
10. Una vez terminado el reconocimiento pulsar en la opción *Close camera capture*.

11. Aparecerá el resultado del reconocimiento.



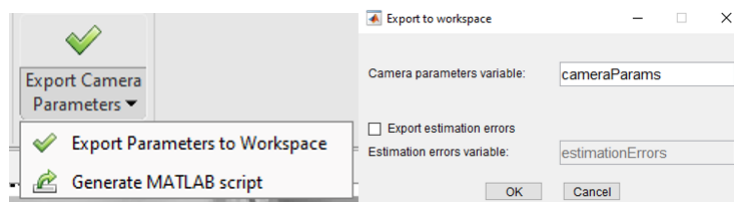
Resultado de reconocimiento – Camera Calibrator.

12. Pulsar en la opción *Calibrate* y aparecerá el resultado de la calibración.



Resultado de calibración – Camera Calibrator.

13. Pulsar en la opción *Export Parameters to Workspace* y en la nueva ventana seleccionar la opción *OK*.



Exportación de parámetros – Camera Calibrator.

14. Se mostrará el resultado en *Command window* y al pulsar doble clic en la matriz que se genera (sección *Workspace*) se tendrá el detalle de la calibración.

```
Command Window

cameraParams =

cameraParameters with properties:

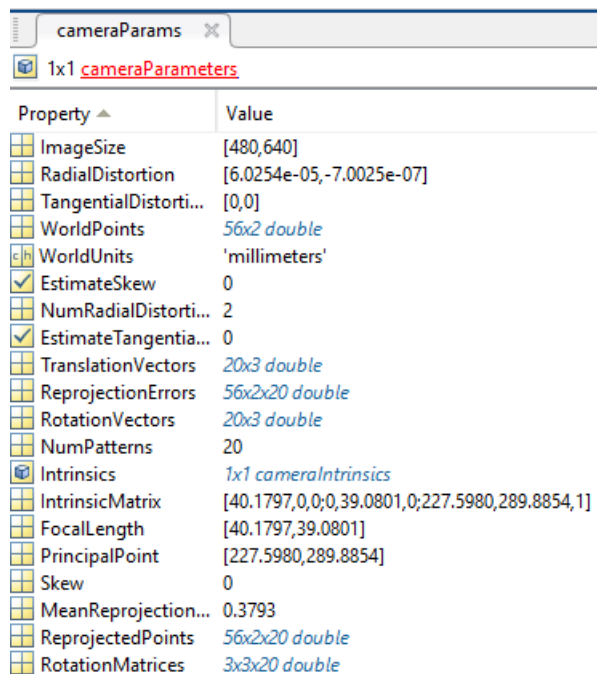
    Camera Intrinsic
        Intrinsic: [1x1 cameraIntrinsic]

    Camera Extrinsic
        RotationMatrices: [3x3x20 double]
        TranslationVectors: [20x3 double]

    Accuracy of Estimation
        MeanReprojectionError: 0.3793
        ReprojectionErrors: [56x2x20 double]
        ReprojectedPoints: [56x2x20 double]

    Calibration Settings
        NumPatterns: 20
        WorldPoints: [56x2 double]
        WorldUnits: 'millimeters'
        EstimateSkew: 0
    NumRadialDistortionCoefficients: 2
    EstimateTangentialDistortion: 0
```

Resultados command window – Camera Calibrator.



Property	Value
ImageSize	[480,640]
RadialDistortion	[6.0254e-05,-7.0025e-07]
TangentialDistorti...	[0,0]
WorldPoints	56x2 double
WorldUnits	'millimeters'
EstimateSkew	0
NumRadialDistorti...	2
EstimateTangentia...	0
TranslationVectors	20x3 double
ReprojectionErrors	56x2x20 double
RotationVectors	20x3 double
NumPatterns	20
Intrinsic	1x1 cameraIntrinsic
IntrinsicMatrix	[40.1797,0,0;0,39.0801,0;227.5980,289.8854,1]
FocalLength	[40.1797,39.0801]
PrincipalPoint	[227.5980,289.8854]
Skew	0
MeanReprojection...	0.3793
ReprojectedPoints	56x2x20 double
RotationMatrices	3x3x20 double

Resultados matriz – Camera Calibrator.