

UNIVERSIDAD DE MONTERREY
VICERRECTORÍA DE CIENCIAS DE LA SALUD



Proyecto de Evaluación Final

**- Ayuda técnica para paciente con malformación congénita
del pie-**

en opción al título de

Licenciatura en Ingeniero Biomédico

Autores

317889 Eugenia Muñoz Garza

322057 Héctor Gerardo Carvajal Cortés

Asesor (a)

Ing. Hiram Alejandro Cantú Campos

San Pedro Garza García, N.L. 30 de noviembre, 2019

Agradecimientos

Damos gracias en primer lugar a nuestras familias por el apoyo incondicional en este periodo de estudio. Principalmente gracias a nuestros padres: José Carlos Muñoz y Maria Eugenia Garza ; Daniel Carvajal y Mónica Cortés por su trabajo, esfuerzo y sacrificio para permitirnos concluir nuestros estudios profesionales; gracias a ustedes hemos logrado llegar hasta aquí y convertirnos en lo que somos.

También queremos agradecer a nuestros maestros por compartir con nosotros sus conocimientos y prepararnos para el ámbito laboral. Agradecemos a nuestro asesor y director de carrera el Ing. Hiram Cantú Campos por guiarnos, no solo en la elaboración de este trabajo, sino a lo largo de nuestra carrera universitaria brindando su apoyo en todo momento.

Agradecemos al profesor Sebastián Larraza por su disposición a apoyarnos en el proyecto, orientarnos y resolver las dudas que se presentaron durante el desarrollo del proyecto.

Un agradecimiento muy especial al profesor Braulio Domínguez que gracias a su interés y emoción por el proyecto nos motivaba a dar lo mejor de nosotros en cada etapa del proyecto; gracias por todo su tiempo invertido en apoyarnos y resolver nuestras dudas en cualquier momento y principalmente gracias por enseñarnos tantas cosas nuevas en tan poco tiempo.

Por último, agradecemos a todas aquellas personas que estuvieron con nosotros a lo largo de esta etapa y de alguna forma u otra nos ayudaron en la culminación de este proyecto.

Índice

1. Datos generales de la organización	5
2. Representante de la institución u organismo cliente	5
3. Descripción del problema o necesidad	5
4. Objetivos	6
4.1 Objetivo general	6
4.1.1 Propuestas	6
4.2 Objetivos particulares	6
5. Palabras clave	6
6. Alcance y entregables	6
7. Marco Teórico	7
7.1 Anatomía del pie	7
7.1.1 Retropié	7
7.1.2 Parte media	7
7.1.3 Antepié	7
7.2 Malformación congénita del pie	8
7.2.1 Tipos de malformaciones congénitas del pie	8
7.2.1.1 Pie equino	8
7.2.1.2 Pie valgo	8
7.2.1.3 Pie varo	9
7.2.1.3 Pie plano	9
7.3 Consecuencias de malformaciones congénitas del pie	10
7.4 Ayuda técnica	11
7.5 Ayudas técnicas para miembros inferiores	11
8. Metodología	12
8.1 Desarrollo de ayudas técnicas	12
8.1.1 Asesoramiento y evaluación por expertos de la salud	13
8.1.2 Diseño de ayudas técnicas	16
8.1.2.1 Plantillas ortopédicas para pie plano valgo	16
8.1.2.1.1 Molde de los pies	16
8.1.2.1.2 Escaneo	17
8.1.2.1.2.1 Captura de Fotografías	19
8.1.2.1.2.2 Alineación de fotografías	19
8.1.2.1.2.3 Creación de nube densa de puntos	20
8.1.2.1.2.4 Creación de mesh	20
8.1.2.1.3 Uso de software especializado	21
8.1.2.1.4 Selección de material	23

8.1.2.2 Órtesis tobillo-pie	24
8.1.2.2.1 Toma de medidas	24
8.1.2.2.2 Modelo 3D	25
8.1.2.2.3 Uso de software especializado	26
8.1.2.2.4 Selección de materiales a utilizar	27
8.1.3 Impresión en 3D	27
8.2 Desarrollo de metodología para un sistema de análisis de movimiento	28
8.2.1 Interfaz Sensor-Software	34
8.2.1.1 Sensor Kinect 2.0	34
8.2.1.2 Requisitos interfaz sensor-software	35
8.2.2 Adquisición de movimiento	36
8.2.3 Procesamiento de datos adquiridos	38
8.2.3.1 Alineación modelo base con modelo real	38
8.2.3.2 Biomech Add-On	40
8.2.4 Análisis de marcha	41
9. Resultados	43
9.1 Moldes de los pies	43
9.2 Escaneo de ambos pies mediante fotogrametría	43
9.3 Alineación de segmentos	44
9.3.1 Plantillas ortopédicas para pie plano-valgo	44
9.3.2 Órtesis tobillo-pie	45
9.4 Análisis de movimiento	46
9.4.1 Marcha Control	46
9.4.1.1 Ángulo de Rodilla Derecha-Pie Derecho	46
9.4.1.2 Ángulo de Rodilla Izquierda-Pie Izquierdo	47
9.4.1.3 Barra desplazadora y animación	48
9.4.2 Marcha del Paciente Semana 0	49
9.4.2.1 Ángulo de Rodilla Derecha-Pie Derecho	49
9.4.2.2 Ángulo de Rodilla Izquierda-Pie Izquierdo	50
9.4.3 Comparación Marcha control - Marcha del paciente semana 0	50
9.4.4 Comparación Marcha del paciente semana 0 - semana 1	51
9.4.5 Comparación Marcha del paciente semana 1 - semana 2	53
10. Conclusión	54
11. Acuerdo cliente-alumno	54
12. Referencia bibliográficas	55
13. Código de Honor	57
14. Anexos	58

1. Datos generales de la organización

- Nombre: Universidad de Monterrey
- Dirección: Av. Ignacio Morones Prieto 4500, Jesús M. Garza, 66238 San Pedro Garza García, N.L.
- Director General: -
- Tel: 8215 1000

2. Representante de la institución u organismo cliente

- Nombre: Ing. Hiram Alejandro Cantú Campos
- Puesto: Director del Programa Académico de Ingeniería Biomédica
- Tel: 8215-1000 Ext. 1601
- Correo: hiram.cantu@udem.edu

3. Descripción del problema o necesidad

La población total de México, según datos del 2014 del INEGI, era cerca de 120 millones de personas. El 6% de estas, es decir 7.2 millones de habitantes, sufren de algún tipo de discapacidad. Entre las discapacidades que predominan dentro de la población se encuentra la discapacidad motriz (64.1%), la discapacidad visual (58.4%), la cognitiva (38.8%), la auditiva (33.5%), entre otras. Sin embargo, la discapacidad más reportada entre los habitantes con un 64.1% es la de caminar o moverse utilizando sus piernas, es decir la discapacidad motriz (INEGI, 2014). De acuerdo con la Organización Mundial de la Salud, más de mil millones de personas en el mundo que sufren alguna discapacidad requieren una o más ayudas técnicas y solamente el 10% de las personas que las necesitan disponen de ellas. Se prevé que para el 2050 más de dos mil millones de personas necesitarán ayudas técnicas. Entre las personas que más necesitan ayudas técnicas se encuentran las que padecen alguna discapacidad, las personas mayores, aquellas que sufren de enfermedades no transmisibles y personas con trastornos de salud mental (Organización Mundial de la Salud, 2018).

Las ayudas técnicas, según la Cooperación Mundial sobre Tecnologías de Apoyo (GATE), mejoran la salud y el bienestar de las personas y de sus familias. El uso de dichos productos logra que las personas lleven una vida independiente y productiva. Se encuentra una amplia clasificación de ayudas técnicas y entre ellas están aquellas que ayudan a los miembros inferiores (Ministerio de Salud, 2018).

Existe el caso de un estudiante masculino de la Universidad de Monterrey que presenta una alteración en ambos pies impidiéndole realizar una marcha normal. Se plantea contribuir a la solución de esta necesidad participando en un proyecto interno de la Universidad de Monterrey en el que se le hará una evaluación

al estudiante para así poder proponer la ayuda técnica más adecuada que se adapte a sus necesidades.

4. Objetivos

4.1 Objetivo general

Desarrollo de ayudas técnicas para estudiante de la UDEM con malformación congénita de ambos pies y alteración en la marcha

4.1.1 Propuestas

- Plantilla ortopédica en ambos pies para recuperar el equilibrio y poder corregir la marcha
- Órtesis de tobillo-pie

4.2 Objetivos particulares

- Asesoramiento con expertos de salud para diagnóstico
- Diseño en 3D y fabricación digital de ayudas técnicas
- Desarrollo de metodología para el sistema de análisis de movimiento del paciente
- Validación de ayudas técnicas

5. Palabras clave

Pie plano-valgo, malformación congénita, ayuda técnica, análisis de marcha

6. Alcance y entregables

Alcance:

- Desarrollo e implementación de ayudas técnicas
- Desarrollo de metodología para sistema de análisis de movimiento

Entregables:

- Reporte académico
- Ayudas técnicas
- Metodología de análisis de movimiento
- Validación de ayudas técnicas

7. Marco Teórico

7.1 Anatomía del pie

En el pie humano encontramos una estructura mecánica que tiene una fuerza mayor que las demás estructuras óseas. Dicha estructura está conformada por 26 huesos y 33 articulaciones. El pie lo podemos dividir en tres diferentes segmentos que son: el retropié (amarillo), la parte media (naranja) y el antepié (rojo) como se muestra en la Figura 1. (Rovira, 2019)

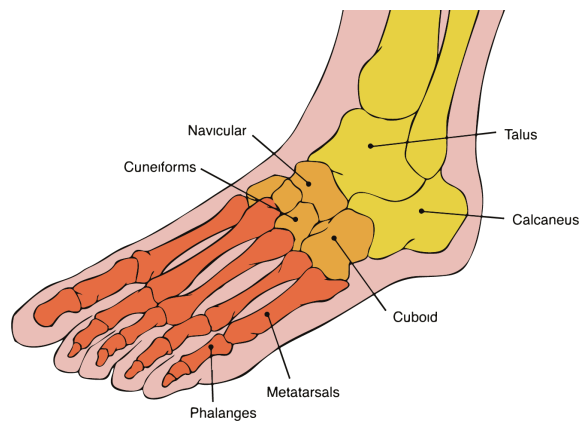


Figura 1. Segmentos del pie.

Recuperado de: <https://www.sportsinjurybulletin.com/foot-pain-looking-up-the-kinetic-chain/>

7.1.1 Retropié

El retropié es la parte posterior de la planta del pie y está compuesto por el talón y tobillo que anatómicamente se conocen como astrágalo y calcáneo. El astrágalo se une a la tibia y al peroné, y el calcáneo es el hueso más grande del pie y es amortiguado por una capa de grasa. (Rovira, 2019)

7.1.2 Parte media

La parte media del pie es la parte de en medio que está compuesta por cinco huesos: el cuboide, navicular y los tres huesos cuneiformes que estos funcionan como un tipo de amortiguador y forman el arco plantar. (Rovira, 2019)

7.1.3 Antepié

El antepié es la parte que incluye los cinco dedos del pie. En esta zona están las falanges y los cinco huesos que forman el metatarso. (Rovira, 2019)

7.2 Malformación congénita del pie

Una malformación es un defecto morfológico de alguna parte del cuerpo humano que en su proceso de desarrollo fue anormal de origen genético. Para el diagnóstico de una malformación congénita del pie es necesaria una evaluación del recién nacido, que esta se hace sobre bases clínicas cómo comenzar a valorar el retropié, la parte media y el antepié de forma individual; también es importante evaluar la elasticidad.

7.2.1 Tipos de malformaciones congénitas del pie

7.2.1.1 Pie equino

Deformidad que se caracteriza por el pie que está en una posición de flexión plantar con relación a la pierna y esto es causado por la contractura del músculo cuádriceps. El paciente con esta malformación tendrá una marcha donde apoyará solo el antepié como se muestra en la Figura 2 (Ortopedia Aeropuerto, S.F.).



Figura 2. Pie equino.

Recuperado de: <http://www.arcesw.com/dpmi.htm>

7.2.1.2 Pie valgo

Deformidad que se caracteriza por su superficie plantar que gira hacia afuera alejándose del plano medio como se muestra en la Figura 3. Observando la radiografía son pies de estructura normal, no hay algún tipo de luxación y esta deformidad es la más frecuente en partos de madres jóvenes sobre todo por una posición defectuosa del feto o compresión por útero (Ortopedia Aeropuerto, S.F.).



Figura 3. Pie valgo.

Recuperado de <http://www.arcesw.com/dpmi.htm>

7.2.1.3 Pie varo

Deformidad que se caracteriza por que el retropié está invertido; su superficie plantar gira hacia adentro acercándose al plano medio como se muestra en la Figura 4 (Ortopedia Aeropuerto, S.F.).



Figura 4. Pie varo.

Obtenido de: <http://www.ortoweb.com/blogortopedia/plantillas-ortopedicas-en-zaragoza-a-medida/pie-varo/>

7.2.1.3 Pie plano

Esta deformidad se caracteriza por que alguna zona de la bóveda plantar es demasiado baja o está desaparecida como se muestra en la Figura 5 y así habiendo más área de contacto de la planta del pie con el suelo. El retropié presenta deformidad en valgo y el antepié se aleja de la línea media del cuerpo. (Ortopedia Aeropuerto, S.F.)

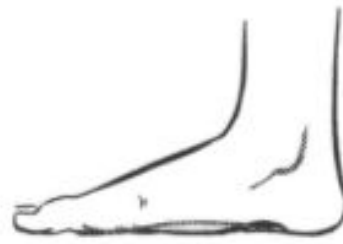


Figura 5. Pie plano.

Recuperado de: <http://www.arcesw.com/dpmi.htm>

7.3 Consecuencias de malformaciones congénitas del pie

Los pies son esenciales para poder ejecutar funciones diarias como por ejemplo, nos permiten el apoyo del peso corporal y el desplazamiento. Al existir alteraciones en los pies, su función puede verse perjudicada dando lugar a problemas en la rodilla, cadera y columna. Los problemas de la columna que pueden surgir son escoliosis siendo una curvatura lateral de la columna vertebral, cifosis refiriéndose a una curvatura exagerada hacia delante de la espalda o lordosis siendo una curvatura hacia adentro de la zona inferior de la espalda, como se muestran en la Figura 6.



Figura 6. Desviaciones anormal de la columna escoliosis, cifosis y lordosis.

Obtenido de: <https://www.bebec.com/producir/@zafrilla-servicios-medicos-fisioterapia/cifosis-consejos-y-causas>

Además las deformidades en los pies pueden también llegar a afectar la marcha (Romero Rubio, 2019). Los trastornos de la marcha hacen referencia a distintas características como por ejemplo una lentificación de la velocidad de la marcha, inestabilidad, alteración en las características del paso o modificación en la sincronía de ambas extremidades inferiores. Existen diversas causas de trastornos de la marcha sin embargo, entre las más frecuentes se encuentran las alteraciones músculo-esqueléticas entre ellas las malformaciones en los pies (Cerde A., 2013).

7.4 Ayuda técnica

Las ayudas técnicas son dispositivos utilizados por una persona que tenga algún tipo de discapacidad; son fabricadas con el propósito de prevenir, compensar, disminuir o neutralizar alguna deficiencia que muestre el paciente o algún tipo de discapacidad. El uso de las ayudas técnicas permite que el paciente pueda incrementar su nivel de independencia y autonomía personal con respecto a su entorno (ASEM, 2008).

7.5 Ayudas técnicas para miembros inferiores

Actualmente, en el mercado existen una variedad de ayudas técnicas para corregir problemas que se presentan en los miembros inferiores. Entre ellas existen plantillas ortopédicas, órtesis de tobillo-pie, órtesis de rodilla-tobillo-pie, entre otras. Por ejemplo, para tratar el problema de pie valgo existen órtesis plantares personalizadas para así lograr la realineación del pie en su mejor posición postural. Las órtesis ayudan a elevar el arco longitudinal situando una pieza correctora en la pronación así como se puede observar en la Figura 7. Las plantillas pueden ser fabricadas de diferentes materiales (PRIM, 2015).



Figura 7. Plantilla Ortopédica.

Obtenido de: https://si-en.valgus-pro-official.eu/article/13_valgus%20deformity%20of%20the%20big%20toe

Existen también las órtesis de tobillo-pie, conocidas como órtesis AFO, que son soportes destinados a controlar la posición y movimiento del tobillo y compensan la debilidad o corrigen deformidades. Dichas órtesis se usan para controlar la caída del pie causada por una variedad de trastornos neurológicos y musculoesqueléticos. La órtesis de tobillo-pie puede incluir una plantilla ortopédica como fue mencionado anteriormente para ayudar con la realineación del pie casi como se observa en la Figura 8 (AliMed,2014).



Figura 8. Órtesis tobillo-pie.

Obtenido de: https://www.algeos.com/DARCO_Body_Ankle_Vario.html

Las ayudas técnicas lo que hacen es compensar, mitigar o neutralizar la deficiencia, discapacidad o minusvalía presente. Existe una gran variedad de ayudas técnicas, aparte de las mencionadas, que ayudan a corregir problemas presentes en los miembros inferiores. Las ayudas técnicas mencionadas anteriormente son específicamente para los miembros inferiores y el uso de estas pueden aumentar la capacidad funcional de una persona para la realización de las tareas cotidianas. Los dispositivos les brindan estabilidad, mejor soporte y permiten un mejor desplazamiento del paciente.

8. Metodología

La metodología del proyecto se dividió en dos principales etapas las cuales consistieron en el desarrollo de dos ayudas técnicas y el desarrollo de la metodología para un sistema de análisis de movimiento. El desarrollo de las ayudas técnicas consistió en el diseño e implementación de dispositivos para corregir la malformación que presentaba el paciente. El desarrollo de la metodología para un sistema de análisis de movimiento consistió en establecer una serie de pasos necesarios que nos permitieran analizar la marcha del paciente y así validar las ayudas técnicas. Es decir, el análisis nos permitió identificar si las ayudas técnicas fueron útiles y lograron corregir de alguna manera la marcha anormal del paciente. Cada una de las dos etapas cuenta con sus respectivas series de pasos que se llevaron a cabo.

8.1 Desarrollo de ayudas técnicas

En la Figura 9 se esquematiza el procedimiento que se siguió para el desarrollo de las ayudas técnicas. Este se dividió en diferentes pasos entre ellos, el asesoramiento y evaluación por expertos de la salud, el diseño de ayudas técnicas

tanto de las plantillas ortopédicas para pie plano-valgo como la órtesis tobillo-pie y por último la impresión en 3D.

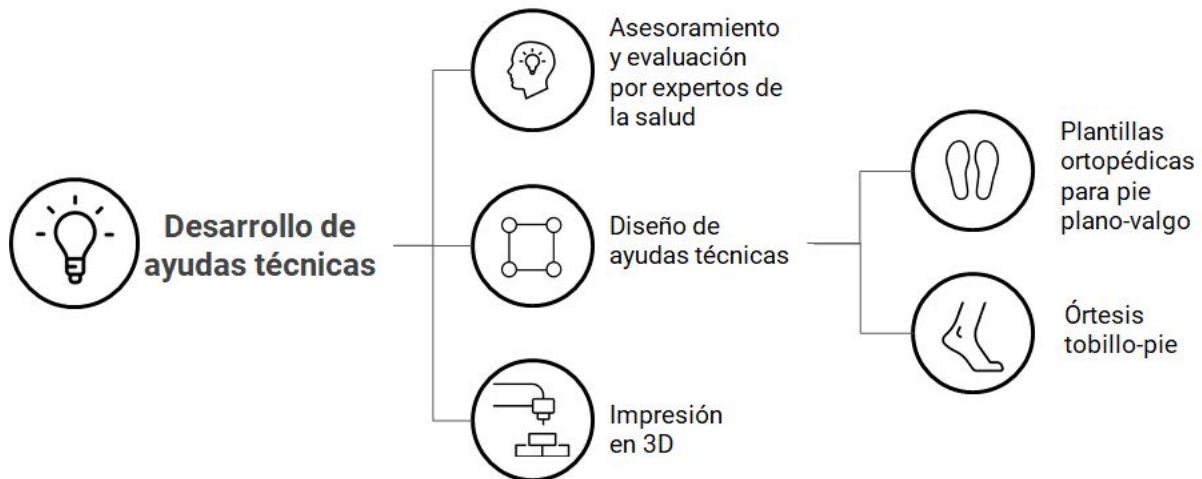


Figura 9. Etapas del desarrollo de ayudas técnicas.

8.1.1 Asesoramiento y evaluación por expertos de la salud

El asesoramiento con expertos fue un punto muy importante con el que decidimos empezar ya que era necesario la opinión de expertos sobre todo en el aspecto médico. Tuvimos diferentes reuniones entre ellas nos reunimos con el Dr. Daniel Solís, la Dra. Ivett Quiñones, con el equipo de trabajo de la empresa Proviem, y por último con los doctores del Centro Estatal de Rehabilitación y Educación Especial (CREE).

El Dr. Daniel Solís es fisioterapeuta y maestro en la Universidad de Monterrey; la recomendación más importante que nos dio fue que nos enfocáramos en las necesidades del paciente. Antes de decidir la ayuda técnica que queremos desarrollar es importante identificar lo que el paciente principalmente quiere corregir. Es decir, identificar si lo que quiere es aliviar el dolor al caminar, corregir su postura, corregir la marcha, un dispositivo discreto u ostentoso o un dispositivo que lo ayude a rehabilitar alguna parte del cuerpo. Además, otra de las cosas que nos recomendó fue algún dispositivo de rehabilitación para uso de noche. De esta manera, el dispositivo podría estar corrigiendo alguno de los problemas que presente el paciente en sus pies durante la noche y así mejorar su desempeño en las actividades que realice durante el día.

La Ing. Biomédico Ivett Quiñones Uriostegui con doctorado en Ciencias del deporte de alto rendimiento nos recomendó el uso de plantillas ortopédicas para corregir el pie y la postura. También nos indicó que se requerirán ejercicios para la pronta rehabilitación así como distintos estudios como por ejemplo el análisis de la marcha, estudio de postura y baropodometría. La Ing. menciona el posible uso de órtesis ya sea tobillo-pie o rodilla-tobillo-pie.

El Ing. Biomédico David Hinojosa y el Lic. Fisioterapeuta Israel Espinosa, de la empresa Proviem, en base a imágenes y videos nos orientaron en el tema explicandonos el tipo de soluciones que su empresa da a los pacientes con este tipo de malformaciones. Explicaron las diferentes pruebas que realizan, entre ellas una prueba que consiste en una plataforma de fuerzas para identificar la distribución de peso en ambos pies.

En el Centro Estatal de Rehabilitación y Educación Especial (CREE) nos reunimos con el Dr. José Barrón y el Dr. Guillermo J Salinas. Aquí fue donde evaluaron al paciente y le diagnosticaron una malformación denominada pie plano valgo. Nos recomendaron sacarle radiografías AP, oblicua y lateral de ambos pies como se observan en las Figuras 10, 11 y 12 respectivamente para así poder determinar la ayuda técnica más adecuada para su malformación. Los doctores analizaron las radiografías y mencionaron que unas plantillas ortopédicas podrían corregir su postura y disminuir el dolor que presentaba al apoyar el pie en el piso. El paciente lo que hace al caminar es apoyar todo su peso en el lado interno del pie lo que ha provocado una desviación en sus dedos. Además, al llevar todo el apoyo al lado interno le provoca dolor y mucha inestabilidad en su marcha. Los doctores recomendaron una plantilla para el pie izquierdo para corregir la manera en que pisa y una plantilla en el pie derecho para prevenir deformación ya que el pie derecho no tenía tanto problema como el otro.



Figura 10. Radiografías AP de los pies.



Figura 11. Radiografías oblicuas de los pies.

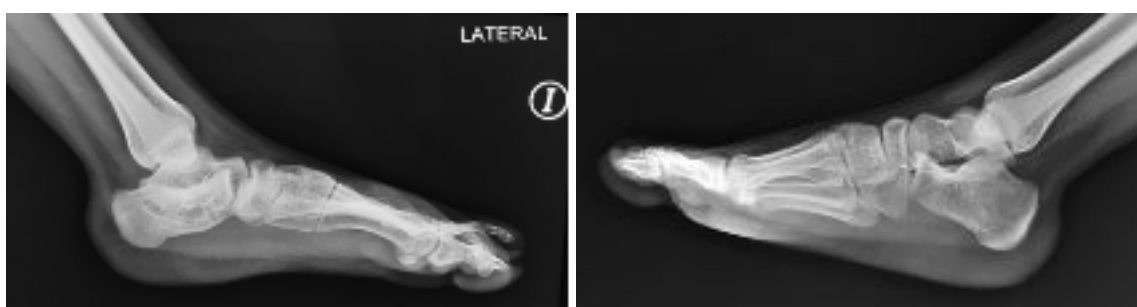


Figura 12. Radiografías laterales de los pies.

En base al análisis de las radiografías y las reuniones con diferentes expertos se tomó la decisión de desarrollar dos ayudas técnicas. La primer ayuda técnica que se decidió desarrollar fueron las plantillas ortopédicas para corregir la manera en que pisa el paciente al caminar y darle mayor estabilidad a su marcha. La segunda ayuda técnica fue la órtesis tobillo-pie. Se decidió desarrollar la órtesis ya que seguimos lo que el Dr. Daniel Solís nos recomendó: enfocarnos en la necesidad del paciente. El paciente nos comentó que siempre al levantarse presentaba mucho dolor en sus pies, principalmente en la planta del pie izquierdo. De acuerdo a los expertos, esto se debe a que sufre de una inflamación del tejido conectivo grueso que une el talón con los dedos del pie. Esta condición es conocida como fascitis plantar y por lo tanto el paciente presenta mucho dolor debido a dicha inflamación (Mayo Clinic, 2018). Esto puede ser doloroso y hacer más difícil el hecho de caminar. Una de las soluciones para este problema es el uso de órtesis nocturna. Por esto, se decidió el desarrollo de una órtesis con el fin de estirar el pie del paciente para así reducir o eliminar la inflamación de los tejidos blandos durante la noche y por lo tanto eliminar el dolor que siente el paciente al levantarse. (Medline Plus, s.f.)

8.1.2 Diseño de ayudas técnicas

8.1.2.1 Plantillas ortopédicas para pie plano valgo

El diseño de las plantillas ortopédicas para pie plano-valgo se dividió en cuatro diferentes etapas que se pueden observar en la Figura 13. Las etapas en las que se dividió el proceso fueron la creación del molde de los pies, el escaneo, el uso de software especializado y por último, la selección del material para la fabricación de las plantillas.

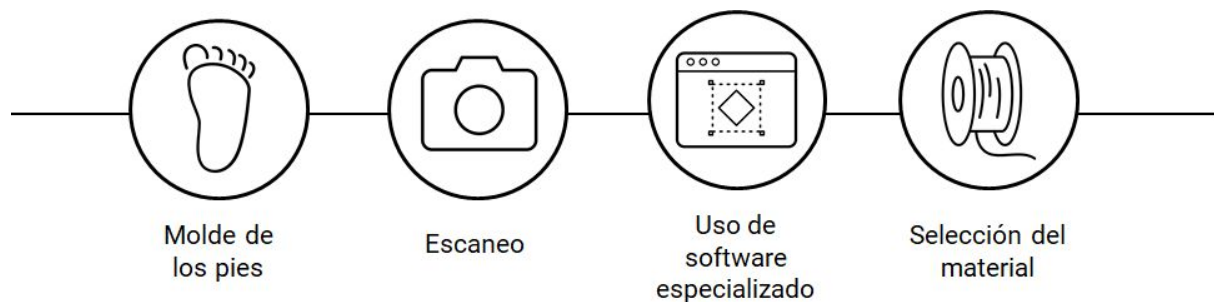


Figura 13. Etapas del diseño de plantillas ortopédicas para pie plano-valgo.

8.1.2.1.1 Molde de los pies

Como se mencionó anteriormente, se decidió obtener unos moldes de los pies del paciente para así escanearlos y conseguir el modelo digital de estos. Se siguieron una serie de pasos que se pueden observar en la Figura 14 para obtener el molde de los pies del paciente y lograr que la geometría fuera idéntica a los originales. Primero, se tuvo que hacer una mezcla de alginato (a). El alginato es un material utilizado por odontólogos para la obtención de los moldes de las dentaduras. La mezcla se tuvo que hacer muy rápido ya que era necesario vaciarlo al recipiente antes de que se convirtiera en gel. Una vez realizada la mezcla, se vacía en el recipiente donde se encuentra el pie del paciente (b) y se esperan unos minutos a que gelifique (c) y así el paciente pueda retirar su pie. Al retirar su pie, queda un molde de alginato (d). Después, se procede a hacer una mezcla de yeso para vaciar en dicho molde (e). El yeso es vaciado en el molde de alginato (f) y se debe esperar aproximadamente 15-20 minutos a que se endurezca por completo para poder proceder a desmoldar (g). El último paso es retirar el molde de alginato y lo que se obtiene es el molde del pie del paciente en yeso (h).

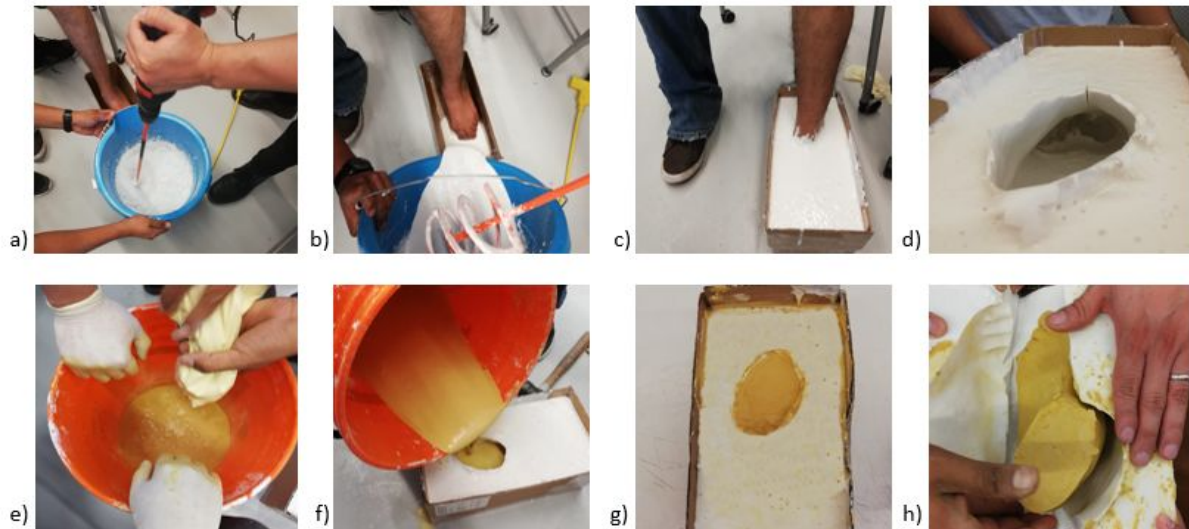


Figura 14. Pasos para la obtención de los moldes de los pies.

8.1.2.1.2 Escaneo

Una vez obtenidos los moldes de los pies, se puede llevar a cabo el escaneo de estos. Existen diversas técnicas de escaneo de objetos entre ellas una llamada fotogrametría. La fotogrametría es una técnica que permite crear modelos 3D a partir de múltiples fotografías tomadas desde diferentes ángulos usando un software especializado que sea capaz de interpretar las variaciones de perspectiva y crear una nube de puntos tridimensional (3D Collective, 2018). El objetivo es realizar un escaneo de algún objeto o parte del cuerpo humano por medio de fotografías y transformarlo a un modelo digital. Lo que hace el software utilizado es reconocer patrones para así poder reconstruir la geometría del objeto que fue escaneado y generar los datos de malla y textura.

En un principio lo que se hizo fue escanear directamente el pie del paciente en lugar de los moldes. Sin embargo, la luz se reflejaba mucho en la piel y se presentaron diversos problemas en cuanto a la iluminación y a la geometría del pie. Es por esto, que decidimos hacer ingeniería inversa y en lugar de fotografiar directamente el pie del paciente optamos por sacar unos moldes de ambos pies y fotografiarlos. Además, al cambiar de técnica nos dimos cuenta que la geometría de la superficie plantar era la deseada. Es decir, al capturar directamente el pie del paciente, este se encontraba suspendido en el aire y la geometría cambiaba. En cambio, al capturar el molde del pie la geometría era la adecuada ya que el molde fue hecho simulando el apoyo de los pies con el piso. Al obtener los moldes se logró conseguir la geometría necesaria para el diseño de la ayuda técnica.

El software especializado que se utilizó para llevar a cabo la técnica de fotogrametría fue Agisoft Metashape. Este software realiza procesamiento de

imágenes digitales y genera datos espaciales. El software solamente elige las imágenes que se encuentren en buenas condiciones y elimina todas aquellas que no sirvan para crear la nube de puntos. Además, Agisoft es capaz de determinar la posición en la que fueron tomadas las fotografías para poder determinar si al momento de capturar las fotografías la cámara se encontraba bien alineada. Al obtener la nube de puntos de baja resolución se procede a crear una nube densa de puntos que será útil para la construcción de la malla del modelo 3D. Dicha nube contiene millones de puntos que representan la posición tridimensional de cada uno de los vértices calculados. Los puntos no definen una geometría tridimensional todavía sino que simplemente son puntos en el espacio. Para poder generar la geometría de alta resolución se necesita construir la malla y de esa manera se obtendrá un modelo de 3D de alta resolución muy parecido al objeto real que fue escaneado. Todas estas etapas se realizan utilizando el software mencionado, Agisoft Metashape, después de haber capturado correctamente las fotografías. (Agisoft, 2019)

Es importante capturar correctamente las fotografías del objeto seleccionado y para esto existen diferentes puntos que se deben considerar. Entre ellos se encuentran los siguientes:

- ISO: de preferencia debe ser el más bajo posible para así evitar el ruido. El ruido puede confundir al software durante el proceso de escaneo.
- Iluminación y fondo: Es importante contar con fuentes de luz grandes y situarlas de tal manera que se eliminen las sombras indeseadas. Para objetos brillantes es preferible utilizar un fondo negro y para menos brillantes se recomienda utilizar un chroma o un fondo neutro para evitar contaminación.
- Cantidad de fotos: Se recomienda un mínimo de entre 150-200 fotografías por modelo para obtener mejores resultados. Sin embargo la cantidad de fotos varía y lo recomendado es capturar cada una de las caras del objeto de interés para que el software pueda generar el modelo digital correctamente.
- Formato de archivo: Se recomienda utilizar el formato RAW.

Siguiendo los puntos establecidos se obtendrán resultados de buena calidad ya que el software será capaz de generar un modelo 3D muy parecido al objeto original escaneado (3D Collective, 2018). La técnica de la fotogrametría para el escaneo de los moldes consistió en los siguientes pasos: la captura de fotografías, la alineación de las fotografías, la creación de nube densa de puntos en un software especializado y por último, la creación de la malla.

8.1.2.1.2.1 Captura de Fotografías

Este paso es la parte más importante del proceso ya que si no se capturan buenas fotografías, el escaneo no será exitoso. Para obtener buenas fotografías y cumplir con el objetivo final se tuvieron que tomar en cuenta diferentes factores. Era necesario controlar la luz, enfocar solamente el objeto de interés, eliminar sombras y alinear correctamente la cámara para que todas las fotografías fueran tomadas a la misma altura y ángulo. Al tomar las fotografías se siguió una trayectoria horizontal a la misma altura y avanzando pasos pequeños entre cada toma para capturar cada parte del pie. Fue importante rodear completamente el objeto de interés para que al momento de pasarlo al software pudiera generar el modelo lo más parecido al real. Algunos resultados de las fotografías obtenidas para el molde del pie izquierdo se pueden observar en la Figura 15. Se logró un buen control de luz para evitar la mayor cantidad de sombras y así obtener buenos resultados. El mismo procedimiento se siguió para fotografiar el molde del pie derecho.



Figura 15. Fotografías obtenidas del molde del pie izquierdo.

8.1.2.1.2.2 Alineación de fotografías

Una vez capturadas las fotografías, se importaron al software Agisoft Metashape. Si las fotografías se hicieron en condiciones controladas el software considerara todas sin eliminar para construir la nube de puntos ya que serán de buena calidad. Ya que están cargadas las imágenes a utilizar en el software, se alinean para que se pueda calcular la posición en la que fueron tomadas. Después de alinearse, se obtendrá una nube de puntos dispersos de baja resolución y una aproximación de la posición de donde se tomaron las fotos como se muestra en la Figura 16. En la imagen se puede observar la trayectoria que se siguió para tomar las fotografías. La alineación de fotos permite comprobar si la cámara estaba correctamente alineada. En este paso se puede identificar si hay algún problema con el proceso ya que podemos revisar cuáles imágenes si acepto el software y cuáles no. También, al observar la nube de puntos generada, se observa si tenemos puntos residuales no deseados que se puedan eliminar para así agilizar el siguiente paso del proceso que es la creación de la nube densa de puntos. (Agisoft, 2019)

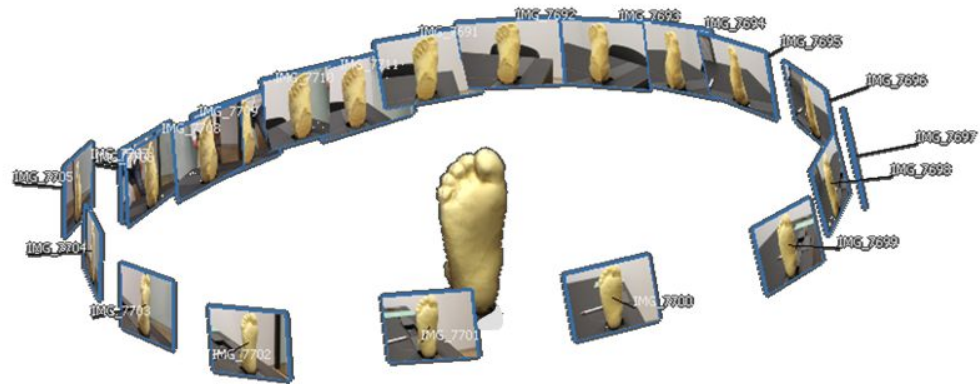


Figura 16. Posición de fotografías capturadas.

8.1.2.1.2.3 Creación de nube densa de puntos

Este paso puede tardar mucho tiempo por lo que es importante asegurarse que todo esté correctamente alineado. Una forma de acelerar el proceso es eliminando todo el fondo que pudo haber detectado el programa y dejando solamente el objeto de interés. Basado en la aproximación de posición de la cámara, el programa calcula la información de profundidad de cada captura para combinarla en una sola nube densa de puntos como se observa en la Figura 17.



Figura 17. Nube densa de puntos.

8.1.2.1.2.4 Creación de mesh

Basado en la información obtenida por la nube de puntos se construye una malla como se observa en la Figura 18 y así se genera un modelo 3D de alta resolución que representa de manera casi idéntica al objeto escaneado, en este

caso, el pie del paciente. Por último, es necesario proyectar las fotografías sobre la geometría ya obtenida y generar la textura.



Figura 18. Creación de malla.

Obteniendo el escaneo como se mostró en el proceso anterior, es posible importar el modelo digital a un software especializado para poder diseñar las ayudas técnicas y así se adapten a la geometría de los pies del paciente.

8.1.2.1.3 Uso de software especializado

El escaneo de los moldes permitió digitalizar los pies del paciente para lograr que las plantillas se adaptaran a la geometría de las plantas de los pies. El software que se utilizó fue Rhino 3D con el plug-in de LutraCad el cual se especializa en diseño de plantillas. Aquí se importaron los escaneos de ambos pies del paciente para proceder a diseñar las plantillas. Lo primero que hizo el software fue generar en automático el contorno de ambas plantillas como se observa en la Figura 19. El software nos permitió ajustar con puntos de control el contorno de cada plantilla para que así se adaptara adecuadamente a la forma de de los pies del paciente como se observa en la Figura 20.

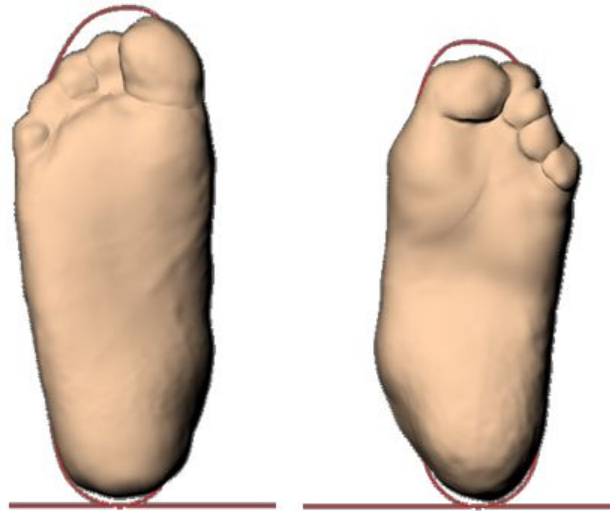


Figura 19. Contorno plantillas ortopédicas sin ajustar.

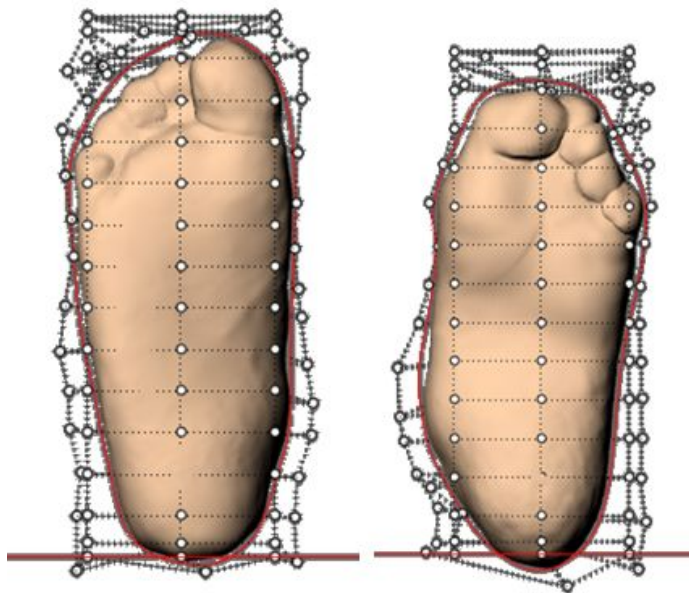


Figura 20. Contorno plantillas ortopédicas ajustados.

El software LutraCad, después de hacer el ajuste manual de los contornos, generó en automático las superficies de las plantillas como se puede observar en la Figura 21. Cada pie tenía diferente geometría y por lo tanto el diseño de la plantilla para cada pie variaba. En la Figura 21, se puede observar que el arco del pie izquierdo estaba más pronunciado que el del pie derecho. También, se puede observar que los dedos del pie izquierdo se encuentran más tensionados que los dedos del pie derecho, a esto se le conoce como “dedos en martillo”; por lo tanto iba a ser necesario agregar mayor altura a la plantilla izquierda para compensar la diferencia de alturas entre ambos pies.

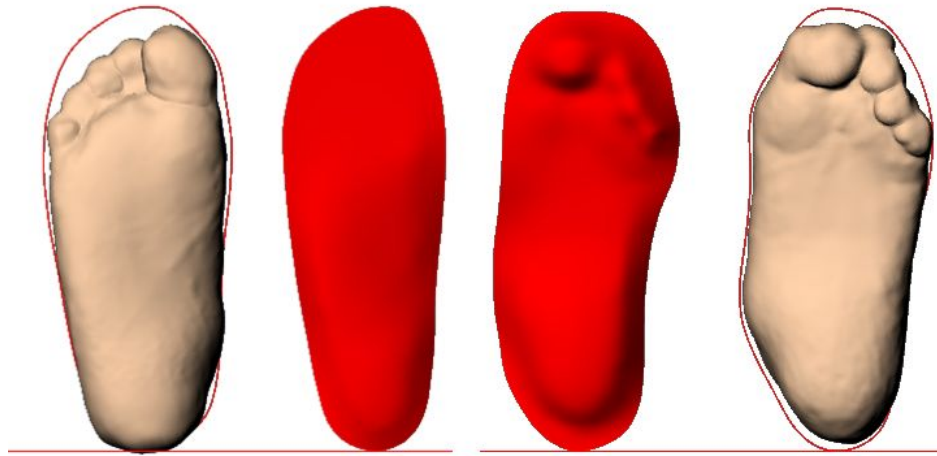


Figura 21. Superficies plantillas.

Una vez que se tenía las superficies de las plantillas, utilizando el mismo software Rhino 3D, se agregó un grosor de 3mm a cada plantilla como se observa en la Figura 22.

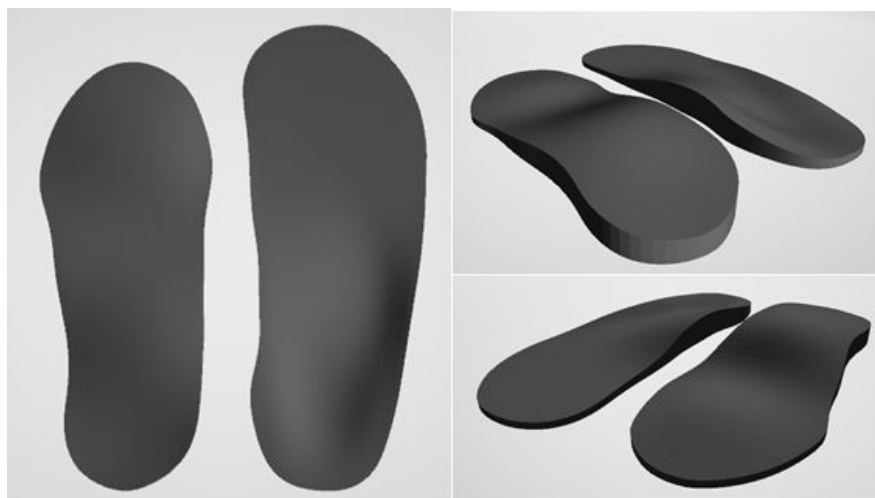


Figura 22. Plantillas ortopédicas para pie plano-valgo

8.1.2.1.4 Selección de material

El siguiente paso, la selección de materiales, fue muy importante para después poder llevar a cabo la fabricación de las plantillas ortopédicas para pie plano-valgo. Era necesario un material flexible que el paciente no sintiera incómodo al utilizar pero a su vez un material con mínima deformidad ya que se tenía que mantener la geometría de la superficie plantar. Además, el material requería alta durabilidad para evitar el rápido desgaste ya que las plantillas serían de uso diario. Tomando en cuenta estas especificaciones se evaluaron tres distintos materiales: el PLA, el TPU/TPE y el PETG como se pueden observar en la Tabla 1. En base a los datos de la tabla describiendo cada material se tomó la decisión que para las

plantillas ortopédicas se iba a utilizar el filamento TPU/TPE ya que cumplía con las especificaciones buscadas.

	PLA	TPU/TPE	PETG
Dureza	Alta	Media	Alta
Deformación	Mínima	Mínima	Mínima
Flexibilidad	Baja	Muy alta	Media
Durabilidad	Media	Muy alta	Alta

Tabla 1. Tabla comparativa de filamentos para impresión en 3D

8.1.2.2 Órtesis tobillo-pie

El diseño de la órtesis tobillo-pie se dividió en cuatro diferentes etapas que se pueden observar en la Figura 23. Las etapas en las que se dividió el proceso fueron la toma de medidas, la creación del modelo 3D, uso de software especializado y por último, la selección del material para la fabricación de las

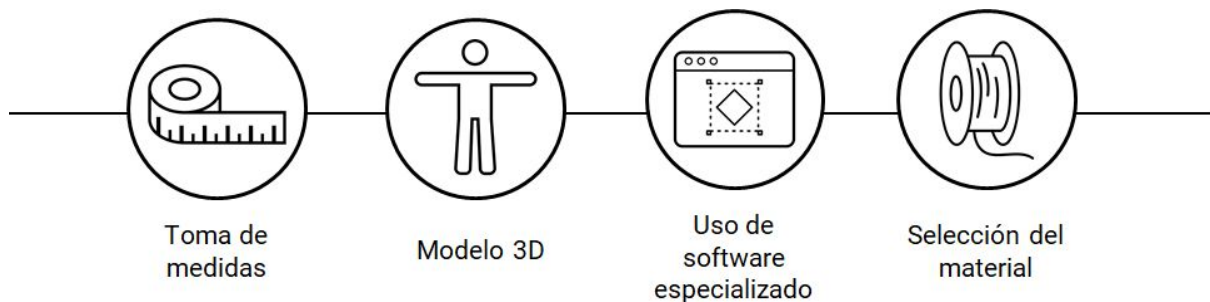


Figura 23. Etapas del diseño de la órtesis tobillo-pie.

8.1.2.2.1 Toma de medidas

Para el diseño de la órtesis tobillo-pie fue necesario primero tomar medidas de los miembros inferiores del paciente, para esto se tomó el diámetro de distintas secciones entre ellas la rodilla, la pantorrilla y el tobillo así como también el largo entre la rodilla y el tobillo como se muestra en la Figura 24.

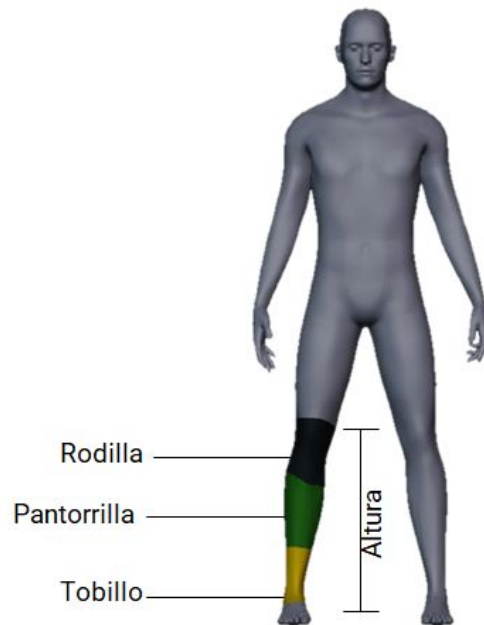


Figura 24. Toma de medidas del paciente

8.1.2.2.2 Modelo 3D

Una vez teniendo las medidas del paciente se utilizó el software llamado MakeHuman dónde se genera en automático un modelo humano en 3D con dimensiones parametrizadas como se muestra en la Figura 25. Aquí se puede modificar el modelo añadiendo las medidas que correspondan al paciente. Lo que se hizo fue añadir las medidas del diámetro de la rodilla, pantorrilla y tobillo del paciente para así lograr que el modelo fuera lo más parecido al paciente. Las medidas restantes del cuerpo humano no fueron consideradas ya que solamente nos importaban las medidas de los miembros inferiores donde se iba a colocar la órtesis. Este modelo se exportó a un software de modelado digital llamado Blender para el diseño de la órtesis tobillo-pie.



Figura 25. Modelo 3D de MakeHuman con medidas del paciente

8.1.2.2.3 Uso de software especializado

Al tener el modelo de los miembros inferiores con sus respectivas medidas del paciente en el software Blender lo que se hizo fue el diseño de la órtesis. Sobre la pierna del modelo se construyó la órtesis para que así se adecuara exactamente a la forma de la pierna del paciente como se observa en la Figura 26. Para que se fuera creando la órtesis tuvimos que ir modificando los vértices de la figura, moviendo puntos y ajustando ángulos para que el dispositivo tuviera el diseño que buscábamos.



Figura 26. Diseño de órtesis en software Blender.

Después de terminar el diseño de la órtesis, con las medidas adecuadas para que se ajustara correctamente al paciente, se utilizó un tissue add-on Dual Mesh del diseñador computacional Alessandro Zomparelli para generar una forma orgánica de celdas como se muestra en la Figura 27. Además de agregarle dicho add-on nos dimos cuenta que era mucho más cómodo una órtesis que no cubriera toda la pierna y por eso decidimos reducir el largo.



Figura 27. Órtesis tobillo-pie.

8.1.2.2.4 Selección de materiales a utilizar

El siguiente paso, la selección de materiales, fue muy importante para después poder llevar a cabo la fabricación de la órtesis tobillo-pie y tratar la condición que presentaba el paciente: fascitis plantar. Para esto se requería un material con una dureza alta para así mantener el pie en una posición específica. Además, el material requería alta durabilidad para evitar el rápido desgaste ya que será de uso diario. Tomando en cuenta estas especificaciones se evaluaron tres distintos materiales: el PLA, el TPU/TPE y el PETG como se pueden observar en la Tabla 2. En base a los datos de la tabla describiendo cada material se tomó la decisión que para la órtesis tobillo-pie se iba a utilizar el filamento PLA ya que cumplía con las especificaciones buscadas.

	PLA	TPU/TPE	PETG
Dureza	Alta	Media	Alta
Deformación	Mínima	Mínima	Mínima
Flexibilidad	Baja	Muy alta	Media
Durabilidad	Media	Muy alta	Alta

Tabla 2. Tabla comparativa de filamentos para impresión en 3D.

8.1.3 Impresión en 3D

Al obtener el diseño final, con ayuda de softwares especializados, de ambas ayudas técnicas se llevó a cabo el proceso de fabricación de cada una de ellas. El proceso de fabricación que se decidió utilizar fue la impresión en 3D. La impresión en 3D es un proceso que produce objetos sólidos tridimensionales a partir de un archivo digital. El producto final de la impresión de las plantillas ortopédicas para pie plano-valgo se puede observar en la Figura 28. El filamento utilizado para la impresión fue el TPU en color azul. Por otra parte, el producto final de la ortesis tobillo-pie se puede observar en la Figura 29. La estructura externa fue la que se imprimió en 3D sin embargo, se decidió agregar accesorios extras para mayor comodidad del paciente. Se agregaron tiras de velcro para sujetar correctamente la órtesis durante la noche así como también un relleno interno de PVC para evitar molestias del paciente al tener contacto directo con la estructura de la órtesis. El filamento utilizado para la impresión fue el PLA en color negro.



Figura 28. Plantillas ortopédicas para pie plano-valgo impresas en 3D.



Figura 29. Órtesis tobillo-pie impresas en 3D.

8.2 Desarrollo de metodología para un sistema de análisis de movimiento

Previo al desarrollo de la metodología para el sistema de análisis de movimiento fue importante entender lo que era una marcha normal contra una anormal, estudiar las fases del ciclo de la marcha, conocer los dispositivos y sistemas que existen para la captura de movimiento así como también los softwares especializados en el estudio de la marcha. Una vez adquirido este conocimiento se pudo proceder a la captura de movimiento del cuerpo humano así como también al análisis de marcha del paciente.

La marcha

La marcha es el patrón de la forma como una persona camina y se produce como resultado de la acción coordinada de diversos sistemas musculares. En otras

palabras, la marcha humana normal se ha descrito como “una serie de movimientos alternantes, rítmicos, de las extremidades y del tronco que determinan un desplazamiento hacia delante del centro de gravedad (Osorio & Valencia, 2013)”. La marcha normal consiste en un pie en contacto con el suelo conocido como la fase de apoyo mientras otro avanzando en el aire conocido como la fase de balanceo. Ambos movimientos son los principales componentes de lo que se conoce como el ciclo de la marcha. El ciclo de la marcha se define como un intervalo de tiempo durante el cual se completa una secuencia de eventos que se repite regularmente. Sin embargo, actualmente el tiempo de cada fase del ciclo se describe en términos de porcentaje ya que se ha observado que los eventos ocurren en una secuencia similar independiente del tiempo. (Gamble & Rose, 2006, pg.40)

El ciclo de la marcha está marcado por una serie de etapas que se pueden dividir de la siguiente manera (Bohórquez Ávila, n.d.):

- Contacto talón suelo
- Apoyo completo de la planta del pie
- Despegue del talón
- Despegue de los dedos
- Oscilación de la pierna
- Contacto talón suelo

La duración de cada etapa corresponde a un porcentaje del ciclo total. El primer contacto del pie con el suelo se designa como 0% mientras que el segundo contacto del mismo pie con el suelo como 100%; el ciclo por lo tanto va de 0%-100%. El ciclo comienza con la fase de apoyo o soporte (60%), este empieza en el instante en que el talón de la pierna de referencia entra en contacto con el suelo y termina con el despegue del pie, es decir cuando los dedos se elevan del suelo. Por otro lado, la fase de balanceo (40%) empieza con la aceleración del extremo de la pierna después de que los dedos dejan el suelo. El pie avanza hacia adelante en el aire preparándose para el siguiente apoyo. La fase de balanceo termina en el momento en que el talón vuelve a tener contacto con el suelo. En la Figura 30 se observa la división entre ambas fases del ciclo de la marcha. La fase de apoyo se indica en color azul mientras la fase de balanceo en color rojo. Es importante entender cada fase del ciclo y entender lo que sucede con cada segmento de la pierna al caminar para así poder llevar a cabo un análisis de la marcha. (Acosta, n.d.)

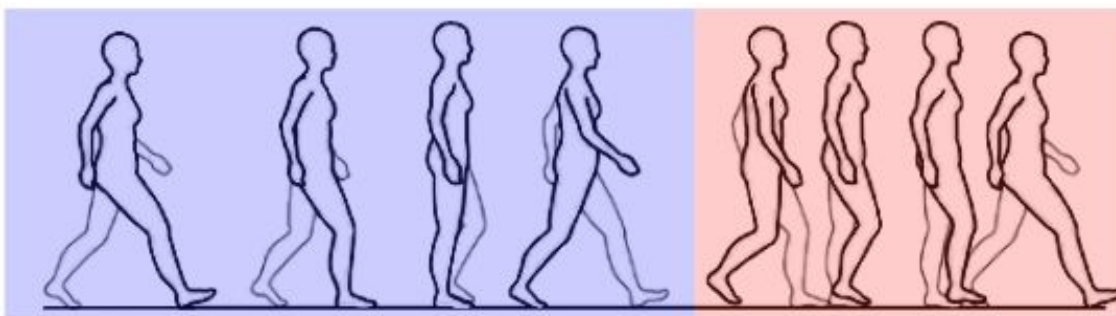


Figura 30. Ciclo de la marcha

Obtenido de: https://www.researchgate.net/figure/Visual-illustration-of-gait-phases-and-the-corresponding-terminology-Note-Adapted-from_fig2_330223176

Análisis de marcha

El análisis de la marcha es el estudio del movimiento del cuerpo humano. El estudio del movimiento del cuerpo humano permite analizar las características propias de cada individuo, para en función de los resultados, diagnosticar patologías o alguna disfunción de las extremidades inferiores que pueden ser tratadas por especialistas (Sumba, Cuenca & Delgado, 2013). La importancia de realizar dichos estudios es que ayuda a comprender de mejor manera el mecanismo de desplazamiento.

La rama que se encarga de la descripción del movimiento del cuerpo sin tomar en cuenta las fuerzas que lo causan es la cinemática. La cinemática es el estudio que describe los movimientos del cuerpo en conjunto así como también los movimientos relativos de las partes del cuerpo durante las diferentes fases. Dicha rama es aplicada al análisis de la marcha considerando cada segmento de la extremidad como un cuerpo rígido. Los segmentos que participan en la movilidad del cuerpo humano generalmente se dividen en dos: los segmentos de carga que son HAT (cabeza, brazos, tronco) y los segmentos del aparato locomotor que incluyen la pelvis, muslo, pierna y pie como se muestra en la Figura 31. Estos últimos segmentos se articulan con la unión lumbosacra, el tobillo, la rodilla y la cadera. Durante el análisis, se mide la posición de las articulaciones y segmentos del cuerpo para así observar las variaciones de los ángulos de los segmentos durante cada fase del ciclo de la marcha. (Miller, 2005, pg. 292)

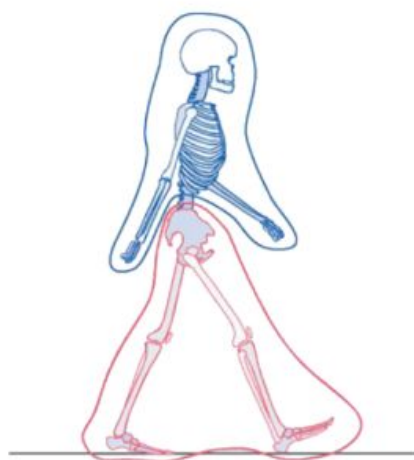


Figura 31. Segmentación del cuerpo humano.

Obtenidos de: https://books.google.com.mx/books?id=V_p50E-Up71C&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=snippet&q=locomotor&f=false

Captura de movimiento

El análisis de movimiento tiene gran impacto en el diagnóstico de las dificultades de una persona al caminar. Es necesario una descripción detallada de las acciones que ocurren con cada articulación y es por eso que se han desarrollado varias técnicas para lograr obtener dicha información. Actualmente existen diversas tecnologías disponibles en el mercado que permiten realizar el análisis. Las tecnologías existentes permiten lo que se conoce como la captura de movimiento (MoCap) que consiste en grabar los movimientos de un humano y recrearlos en modelos digitales. Mocap permite la captura de parámetros de movimiento lineales y coordenadas angulares, velocidades y aceleraciones para extremidades y articulaciones. Esto se logra mediante el seguimiento de un conjunto de puntos de interés en una escena por un determinado tiempo. Dichos puntos se ubican en las áreas de la persona donde existe mayor información de movimiento y hacen referencia a las articulaciones del cuerpo humano. (Gómez Echeverry, et al., 2018)

La captura de movimiento está compuesta por sensores y sistemas ópticos de captura que son utilizados para identificar la posición de un sujeto en el espacio, mediante el uso de cámaras. Existen diferentes sistemas para obtener los datos de posición entre ellos se encuentran los sistemas ópticos con marcadores, los sistemas sin marcadores y los sistemas no-ópticos. Al utilizar sistemas ópticos con marcadores, estos se colocan en las articulaciones del sujeto para detectar los movimientos que realiza cada segmento. Aunque es un método preciso, la colocación de marcadores es un método laborioso que requiere mucho tiempo. Además, dicho método requiere de equipo costoso, conocimiento de un experto para la colocación de los marcadores y un entorno de laboratorio específico. Es por esto que se desarrollaron nuevas técnicas en las que los marcadores no son necesarios ya que con cámaras y softwares se puede realizar el análisis. Esto se

logra mediante algoritmos que analizan las acciones del usuario, identificando formas humanas, movimientos, expresiones y gestos. La ventaja de este método es que es mucho menos costoso, fácil de manipular y no se necesita conocimiento de expertos ya que no se colocan marcadores en el sujeto. (Shen & Ting,2014)

Los sistemas ópticos mencionados anteriormente utilizan cámaras que graban al sujeto en movimiento. Cada cámara extrae coordenadas en 2D que son analizadas y el resultado genera coordenadas en 3D. Dichos sistemas miden el desplazamiento a diferencia de los sistemas no-ópticos que miden rotación, aceleración y flexión. La ventaja de estos últimos es que no requieren de tecnología compleja para recopilar los datos del movimiento. Un ejemplo de sistemas no-ópticos es el uso de acelerómetros. (Shen & Ting,2014)

Software MoCap

El Kinect es el dispositivo que cuenta con una cámara para llevar a cabo la grabación del movimiento del cuerpo humano. Sin embargo, también es necesario contar con un software especializado que reciba esa información para poder ser analizada. Existen distintos softwares para la captura de movimiento sin marcadores entre ellos uno llamado iPi Soft-Motion Capture™. El software es capaz de trabajar con 1 o 2 cámaras del Kinect para rastrear los movimientos. Dicho software cuenta con dos programas, iPi Recorder que se dedica a la grabación e iPi MoCap Studio que analiza los movimientos de las grabaciones realizadas. El programa iPi Recorder captura, reproduce y procesa grabaciones de video de cámaras y sensores de profundidad, por ejemplo del sensor Kinect. Los registros capturados pueden ser utilizados para el rastreo de movimiento en el programa iPi MoCap Studio. Este último, procesa los videos grabados con iPi Recorder y genera un modelo 3D del movimiento del cuerpo. Lo que hace es calcular la postura humana aplicando la cinemática y logra coincidir el modelo 3D con la posición real del cuerpo humano.

Una parte importante del iPi Mocap Studio es el complemento con el que cuenta llamado iPi Biomech. Lo que hace es, a partir de la locomoción humana registrada, calcula las coordenadas de las articulaciones, ángulos Euler, velocidades lineales y angulares y aceleraciones. Como se mencionó anteriormente, el cuerpo humano puede ser modelado como un sistema de segmentos rígidos. Para el análisis de la marcha, el programa iPi Mocap Studio divide el cuerpo en los siguientes segmentos que se pueden observar en la Figura 32. Cada segmento se puede ver en la Tabla 3. Al dividir el cuerpo humano en segmentos, se obtienen las coordenadas y se logra calcular ángulos entre cada segmento. (Afanasyev et al., 2015)

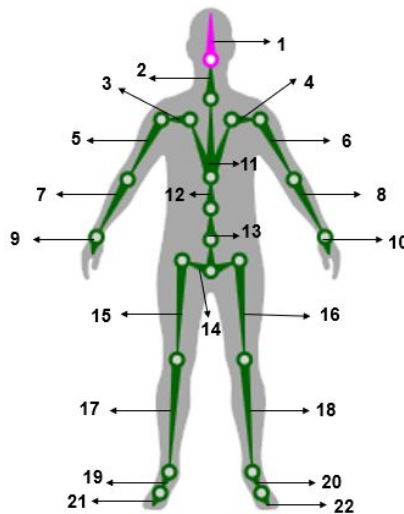


Figura 32. Identificación segmentos en cuerpo humano.

SEGMENTOS	
1 Cabeza	12 Zona torácica
2 Cuello	13 Zona lumbar
3 Clavícula derecha	14 Pelvis
4 Clavícula izquierda	15 Muslo derecho
5 Brazo derecho	16 Muslo izquierdo
6 Brazo izquierdo	17 Pierna derecha
7 Antebrazo derecho	18 Pierna izquierda
8 Antebrazo izquierdo	19 Pie derecho
9 Mano derecha	20 Pie izquierdo
10 Mano izquierda	21 Pie derecho
11 Tórax	22 Pie izquierdo

Tabla 3. Segmentos cuerpo humano.

Obtenido de: <https://dspace.ups.edu.ec/bitstream/123456789/5092/1/UPS-CT002692.pdf>

Las herramientas mencionadas anteriormente, tanto el sensor Kinect como el software de captura de movimiento proporcionan información que sirve para poder realizar un análisis de marcha. El análisis de marcha ayudará a identificar si existe algún problema en la locomoción humana y podrá dar paso a la recomendación de alguna ayuda técnica para mejorar el problema detectado y de esta manera corregir la marcha del paciente.

Una vez entendido el concepto de la marcha y las implicaciones de un análisis correcto se procedió a desarrollar la metodología para el sistema de análisis de movimiento. En la Figura 33 se esquematiza el procedimiento que se siguió el cual se dividió en diferentes pasos entre ellos, la interfaz sensor-software, la adquisición de movimiento, el procesamiento de datos adquiridos y por último el análisis de la marcha.



Figura 33. Etapas del desarrollo de la metodología para el sistema de análisis de movimiento.

8.2.1 Interfaz Sensor-Software

8.2.1.1 Sensor Kinect 2.0

Como se mencionó anteriormente, se ha realizado mayor investigación en los sistemas que no utilizan marcadores debido al fácil manejo de ellos. Entre los sistemas de captura de movimiento sin marcadores más notables se encuentra el dispositivo Kinect. El dispositivo fue desarrollado para la detección de movimiento para la consola de videojuegos de Xbox y posteriormente adoptado para PC con Windows para ser útil en otros campos como la robótica, el seguimiento del esqueleto humano, reconstrucción 3D, terapia asistencial y la biomecánica.

El sensor Kinect cuenta con una cámara RGB (Red, Blue, Green), un conjunto de micrófonos, un sensor de profundidad capaz de capturar el movimiento tridimensional de un cuerpo humano y es además capaz de reconocer la voz y el rostro. Actualmente, existen dos versiones del Kinect sin embargo, la más nueva (Kinect v2 para Xbox One) incluye mejores funciones que la pasada y es con la que se han obtenido mejores resultados en los estudios de captura de movimiento. Esto se debe a que cuenta con una cámara RGB e IR con mayor resolución y un campo de visión más amplio que la versión anterior. En la Tabla 4 se puede observar la

comparación entre ambas versiones del Kinect mencionando las características principales.

Característica	Kinect v1	Kinect v2
Cámara RGB	640 x 480 30fps	1920 x 1080 30 fps
Cámara de profundidad	320 x 240	512 x 424
Distancia de profundidad máxima	~4.5 M	~4.5 M
Distancia de profundidad mínima	40 cm	50 cm
Campo de visión horizontal	57 grados	70 grados
Campo de visión vertical	43 grados	60 grados
Articulaciones definidas	20 articulaciones	26 articulaciones
USB	2.0	3.0
SO compatible	Win 7, Win 8	Win 8

Tabla 4. Tabla comparativa versiones de Kinect.

Obtenida de: <http://zugara.com/how-does-the-kinect-2-compare-to-the-kinect-1>

El dispositivo Kinect que se observa en la Figura 34 (A) cuenta con un sistema de coordenadas que se establece de la manera en que se puede observar en la Figura 34 (B). Los datos obtenidos por la cámara de profundidad son procesados para generar la posición de cada articulación y almacenarlas en coordenadas (x,y,z). Se puede ver que el eje z positivo se extiende en la dirección en que apunta el Kinect.



Figura 34. (A) Kinect v2 (B) Sistema de coordenadas del sensor.

8.2.1.2 Requisitos interfaz sensor-software

Lo primero que se tuvo que hacer para poder empezar a utilizar el sensor fue lograr la conexión del dispositivo a una computadora, el sensor que se va a utilizar es el Kinect V2. Para esto fue necesario adquirir un adaptador de Kinect para

Windows y cumplir con ciertos requisitos. Entre los requisitos estaban que el sistema operativo debía ser Windows 8 en adelante, contar con un procesador de 64 bits y de doble núcleo a 3,2 GHz, un puerto de USB 3.0 y por último 3GB de RAM. La conexión del Kinect a la computadora por medio del adaptador se puede visualizar en la Figura 35. El cable del Kinect (1) se conecta al adaptador (2) y de él salen dos cables más. Un cable que es el adaptador USB 3.0 (3) que va conectado a la computadora y el otro cable que es la fuente de alimentación del adaptador (4,5). Además, para que la computadora pudiera detectar el Kinect fue necesario descargar el software de Microsoft llamado Kinect para Windows SDK 2.0. Dicho software contiene todos los controladores necesarios para que opere correctamente el Kinect en la computadora.

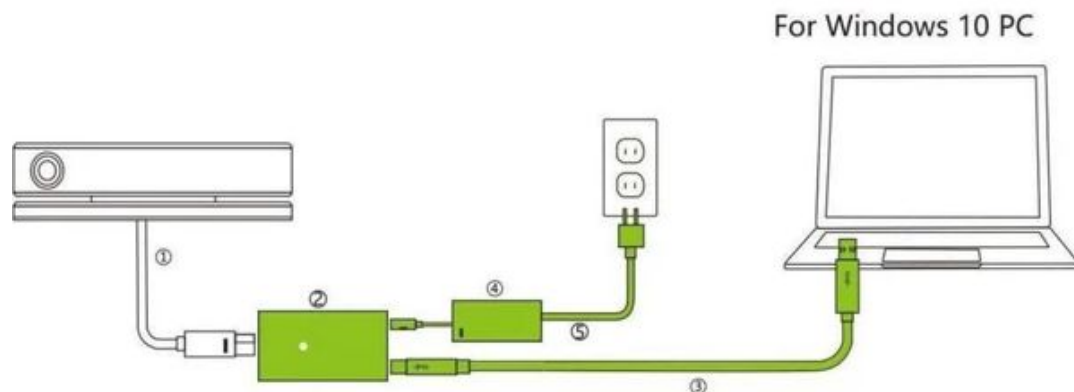


Figura 35. Conexión Kinect a computadora.

Obtenido de: https://cdn.shopify.com/s/files/1/2532/7020/files/Kinect_adapter_-_GadgetiCloud_5_grande.jpg?v=1544085810

8.2.2 Adquisición de movimiento

Para poder desarrollar el sistema de análisis de marcha primero fue necesario capturar el movimiento del cuerpo humano. Para esto, se utilizó el dispositivo Kinect v2 que se colocó enfrente del paciente a 3.80m de distancia como se observa en la Figura 36. Además se midió la distancia que podía caminar el individuo para evitar salirse de los rangos del campo de visión del sensor. La distancia que era capaz de recorrer fueron 2.3m y en esta distancia se completaron un ciclo en una marcha normal.

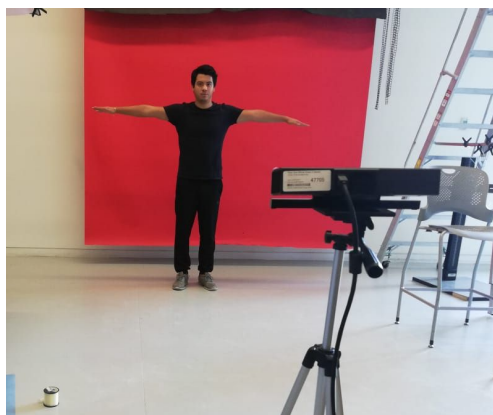


Figura 36. Ubicación Kinect.

Antes de comenzar a grabar, fue necesario evaluar el fondo de la grabación. Ambas acciones se realizaron utilizando el programa iPi Recorder. Es importante que al llevar a cabo la evaluación no se encuentren personas enfrente de la cámara ni objetos moviéndose. El objetivo de la evaluación es eliminar la mayor cantidad de ruido posible para obtener mejores resultados de la grabación. Es importante minimizar la cantidad de puntos amarillos en la escena ya que estos indican que la profundidad en esa parte es desconocida. Otro punto importante a considerar es que la cámara debe ser capaz de captar el cuerpo completo del paciente y los puntos del piso para tomarlo como referencia.

Para verificar que la evaluación haya sido exitosa, es decir, que el Kinect se encuentre bien colocado y el fondo sea el indicado aparecerá una ventana de estadísticas en la parte inferior de la pantalla. Esta ventana contiene valores que indican el rendimiento de captura de video de esa cámara. La información que se encuentra incluye el número bueno de frames que se registraron así como también el número de frames malos por segundo. En caso de que los valores se encuentren en rojo como se observa en la Figura 37 significa que hay que ubicar correctamente el Kinect o modificar el fondo de la escena para lograr que todos los frames sean registrados y no existan valores rojos.

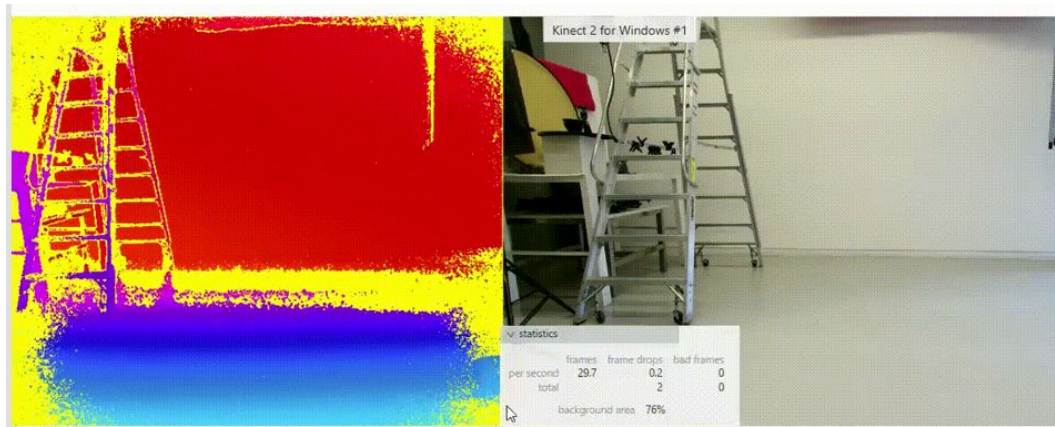


Figura 37. Evaluación de fondo.

Una vez realizada la evaluación, se procede a grabar la marcha del paciente enfrente del sensor Kinect. Es necesario que la pose inicial del usuario sea en “T pose” como se observa en la Figura 38 esto con el propósito de que al pasarlo al programa iPi MoCap pueda detectar el cuerpo y pueda generar un modelo 3D en base a este. Al iniciar la grabación, el usuario puede bajar los brazos y comenzar la marcha. En nuestro caso, fue necesario iniciar el ciclo de la marcha con el pie derecho ya que es el que vamos a tomar como referencia para el análisis. Al finalizar la grabación, el video se guarda en un formato específico (.iPiVideo) que solamente puede ser abierto por iPi Mocap para procesarlo y extraer la animación.

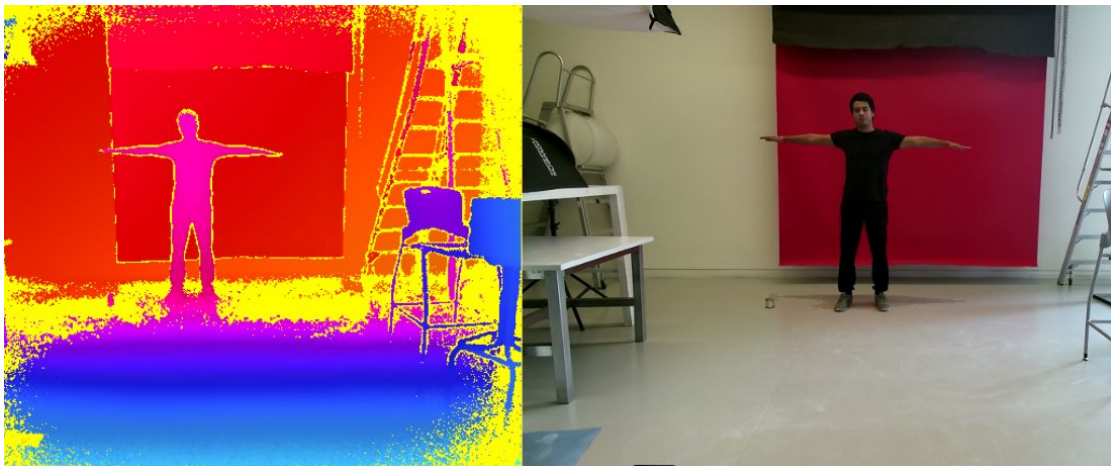


Figura 38. Posición inicial “T-pose”.

8.2.3 Procesamiento de datos adquiridos

Después de realizar la grabación, el video debe ser importado al programa iPi MoCap para ser analizado. Este último programa lo que hace es extraer la información de la captura de movimiento para generar un modelo en 3D. Al tener el modelo generado, se puede obtener información como la posición de articulaciones, velocidades, aceleraciones y ángulos de la marcha del paciente.

8.2.3.1 Alineación modelo base con modelo real

El primer paso al importar el video fue alinear el modelo generado por el programa con el modelo real del video. En la Figura 39 se puede observar que el modelo base generado por el programa es el de enfrente en color azul. Este modelo ya incluye la segmentación del cuerpo humano para así poder analizar cada articulación. El modelo de la parte trasera en color rosa es el real obtenido por la grabación en iPi Recorder. El objetivo fue coincidir el modelo 3D del programa con la posición real del cuerpo humano. Para juntarlos fue necesario mover el modelo azul así como también ajustar las piernas y brazos para que coincidieran con el real.

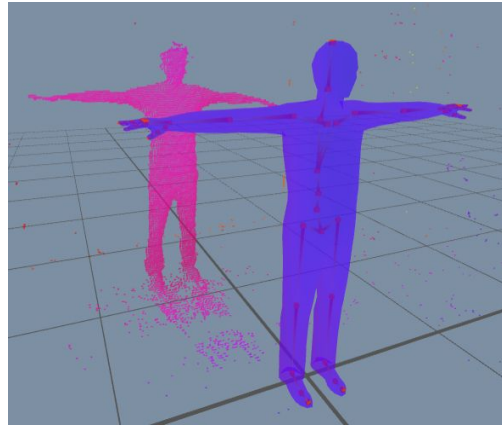


Figura 39. Modelo base y modelo real.

Después de juntar los modelos, fue necesario reajustar la pose para lograr que el modelo 3D copiara exactamente el movimiento del modelo del video. Para hacer esto fue necesario irnos a la pestaña de *Tracking* y seleccionar el botón de *Refit Pose*. De esta manera, existe mayor precisión en la coincidencia de la pose y se puede corregir errores de seguimiento, después se elige la opción *Track Forward* para que el modelo copie los movimientos del video. Después, la parte de post-procesamiento consistió en eliminar el ruido de la animación. Este paso es importante para evitar temblores o movimientos involuntarios que no coincidan con los reales que fueron grabados. Para la eliminación del ruido se oprime el botón de *Jitter Removal* que también se encuentra en la pestaña de *Tracking*. Todos estos ajustes se realizan en la ventana que se observa en la Figura 40.

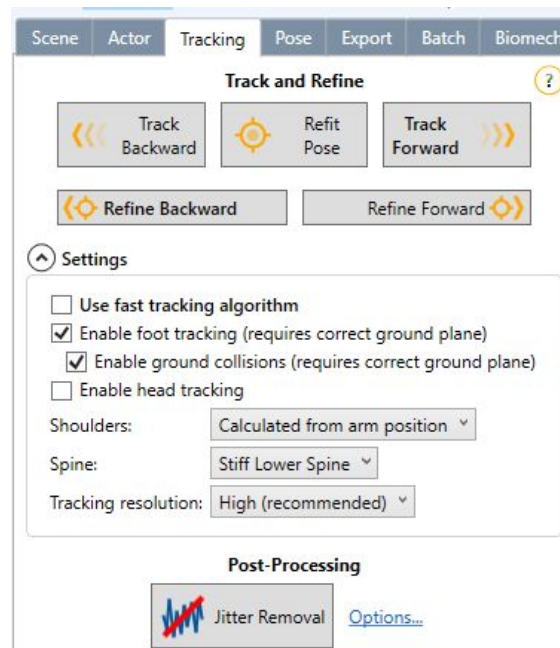


Figura 40. Ventana *Tracking*.

8.2.3.2 Biomech Add-On

Al obtener la animación en 3D copiando los movimientos realizados por el paciente se puede hacer un análisis biomecánico completo. Para esto es necesario que el software cuente con la extensión *Biomech* ya que permitirá analizar el movimiento humano. Lo primero que se tuvo que hacer fue seleccionar los huesos que se querían analizar. En este caso, como se hizo un análisis de la marcha solamente seleccionamos los huesos de las extremidades inferiores. Estos huesos incluyen los de los muslos, piernas, pies y dedos del pie. Al seleccionar los huesos, se procede a seleccionar las cantidades lineales que se desean obtener. Lo que se seleccionó fueron solamente coordenadas ya que al obtener las posiciones de las articulaciones en el espacio nos iba a permitir realizar los cálculos de ángulos entre cada segmento. Al seleccionar las cantidades lineales es necesario también seleccionar el sistema de coordenadas más conveniente. En este caso, el sistema de coordenadas seleccionado fue con respecto al piso. Todas estas acciones se realizan en la pestaña de Biomech que se puede visualizar en la Figura 41. Al completar la selección de todo lo que se quiere analizar, el último paso consiste en exportar los datos obtenidos. Los datos de las posiciones de los huesos al realizar la marcha pueden exportarse al programa MATLAB para ser analizados mediante gráficas.

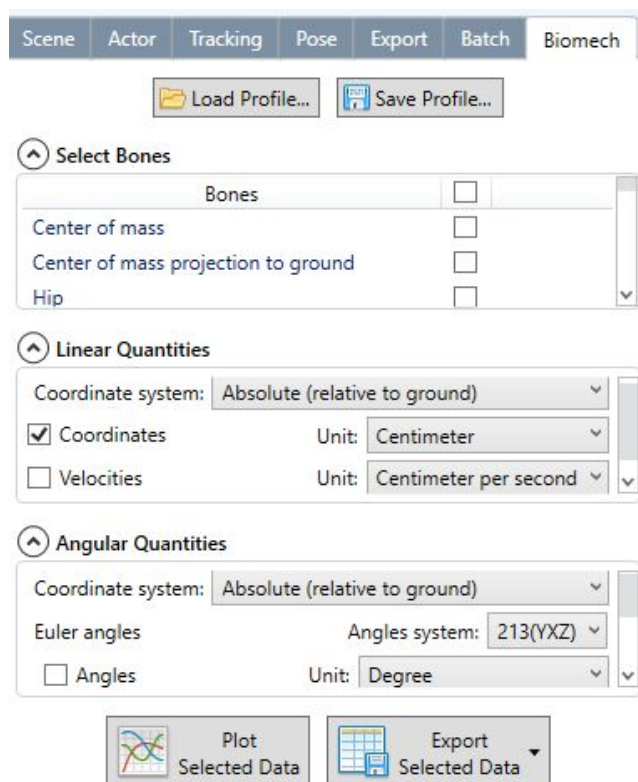


Figura 41. Ventana *Biomech*.

8.2.4 Análisis de marcha

Ya una vez teniendo la matriz de datos importada en Matlab donde encontramos coordenadas de posición en el espacio (x,y,z) de los huesos seleccionados se calcularon los ángulos entre distintos segmentos. En Matlab se desarrolló un código de programación que fuera capaz de calcular automáticamente los ángulos entre segmentos durante la marcha una vez ingresada la matriz de datos de las coordenadas. Se comprobó que el programa realizará correctamente las operaciones mediante el análisis manual del cálculo de ángulos. Se utilizaron diferentes fórmulas como las que se presentarán a continuación.

Primero, se obtuvieron los vectores que representaban los segmentos entre sus respectivos puntos, utilizando la siguiente fórmula:

$$A = [x_1 \ y_1 \ z_1]$$

$$B = [x_2 \ y_2 \ z_2]$$

$$\overrightarrow{AB} = \langle x_2 - x_1 \quad y_2 - y_1 \quad z_2 - z_1 \rangle$$

Se calcularon en total seis segmentos, tres del miembro inferior derecho y tres del miembro inferior izquierdo. Para poder obtener el ángulo entre dos segmentos, los vectores, se deben calcular restando el punto en común entre ellos menos el punto exterior. Por ejemplo para obtener el ángulo entre el muslo y la pierna fue necesario calcular los segmentos AB y CB como se muestran en la Figura 42. Para ello tuvimos que restar las coordenadas del punto en común entre ellos que fue la rodilla (B) menos los extremos que fueron la cabeza del fémur (A) y el tobillo (C).

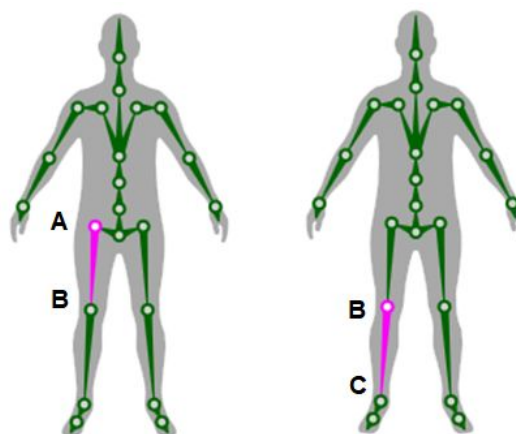


Figura 42. (A) Muslo (B) Pierna

El mismo procedimiento se utilizó para calcular el ángulo entre la planta del pie y la pierna, se calcularon los segmentos BC y DC que se muestran en la Figura 43 restando las coordenadas del punto en común menos las coordenadas de los puntos exteriores. Los puntos fueron el talón (C) menos los puntos exteriores que fueron la rodilla (B) y los dedos del pie (D).

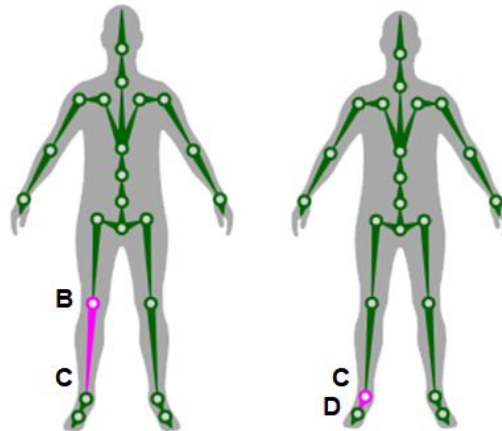


Figura 43. (A) Pierna (B) Pie

Este procedimiento para calcular los segmentos se repite de igual manera para los segmentos del miembro inferior izquierdo. Ya teniendo los seis segmentos calculados procedemos a calcular el ángulo entre ellos con la siguiente fórmula:

$$\theta = \cos^{-1}\left(\frac{\overrightarrow{AB} \cdot \overrightarrow{CB}}{(\|\overrightarrow{AB}\|)(\|\overrightarrow{CB}\|)}\right)$$

Donde el producto punto $\overrightarrow{AB} \cdot \overrightarrow{CB}$ se calcula a partir de la siguiente fórmula:

$$\vec{A} \cdot \vec{B} = \vec{A}_x \cdot \vec{B}_x + \vec{A}_y \cdot \vec{B}_y + \vec{A}_z \cdot \vec{B}_z$$

Donde la magnitud de cada vector se calcula a partir de la siguiente fórmula:

$$\|\overrightarrow{AB}\| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Cada una de las operaciones se hizo por separado para al último integrar todos en una sola fórmula, esto también ayudó a corregir errores con mayor

facilidad ya que se identificaba rápidamente el problema; cada uno de los resultados de cada operación se guardó en un *struct* para su verificación con el cálculo manual.

Una vez teniendo todos los ángulos calculados se guardaron en un *struct* para poder graficarlos en MATLAB donde se compararon los ángulos entre el muslo-pierna y pierna-planta del pie de una marcha normal y una anormal. Para un mayor entendimiento de las gráficas se realizó una animación de la marcha donde al mismo momento una barra se desplazó a lo largo de la gráfica en donde estaba sincronizada con el movimiento de la animación.

9. Resultados

9.1 Moldes de los pies

En la Figura 44 podemos observar los moldes en yeso de los pies del paciente que se hicieron con la ayuda del molde de alginato. El uso del molde en yeso de los pies tiene distintos beneficios, como por ejemplo el que el paciente no tenga que estar presente en todo momento cuando se realicen pruebas, también se pueden capturar mejor las fotografías de distintos ángulos ya que la manipulación de un molde es mucho más fácil que trabajar directamente con el pie del paciente.



Figura 44. Moldes de yeso de ambos pies del paciente.

9.2 Escaneo de ambos pies mediante fotogrametría

Una vez realizada la técnica de la fotogrametría al molde del pie, el resultado fue el que se observa en la Figura 45, donde podemos observar que el pie izquierdo del paciente es más corto que el pie derecho midiendo aproximadamente 21.5 cm contra el derecho que mide aproximadamente 23.5 cm teniendo 2 cm de diferencia. Estos modelos de los pies del paciente fueron lo que se utilizaron con el plug-in LutraCad para el diseño de las plantillas.



Figura 45. Escaneo de ambos pies del paciente

9.3 Alineación de segmentos

9.3.1 Plantillas ortopédicas para pie plano-valgo

En la Figura 46 podemos observar dos imágenes, la imagen superior se observa el ángulo entre el punto medio del talón del pie y el punto medio del tobillo de ambos pies del paciente donde el ángulo en el pie izquierdo es de 198° y el derecho de 191° , cuando se debería tener un ángulo cerca de los 180° . En la imagen inferior se observa el ángulo entre el punto medio del talón del pie y el punto medio del tobillo de ambos pies del paciente con el uso de las plantillas ortopédicas desarrolladas donde el ángulo en el pie izquierdo es de 192° y el derecho de 183° . Los resultados obtenidos demuestran la corrección que hacen las plantillas ortopédicas a la postura del pie del paciente. Los valores de los ángulos obtenidos con el uso de plantillas se aproximan más al valor deseado (180°). El uso de plantillas permite una mejor alineación del talón con el tobillo de los pies del paciente.

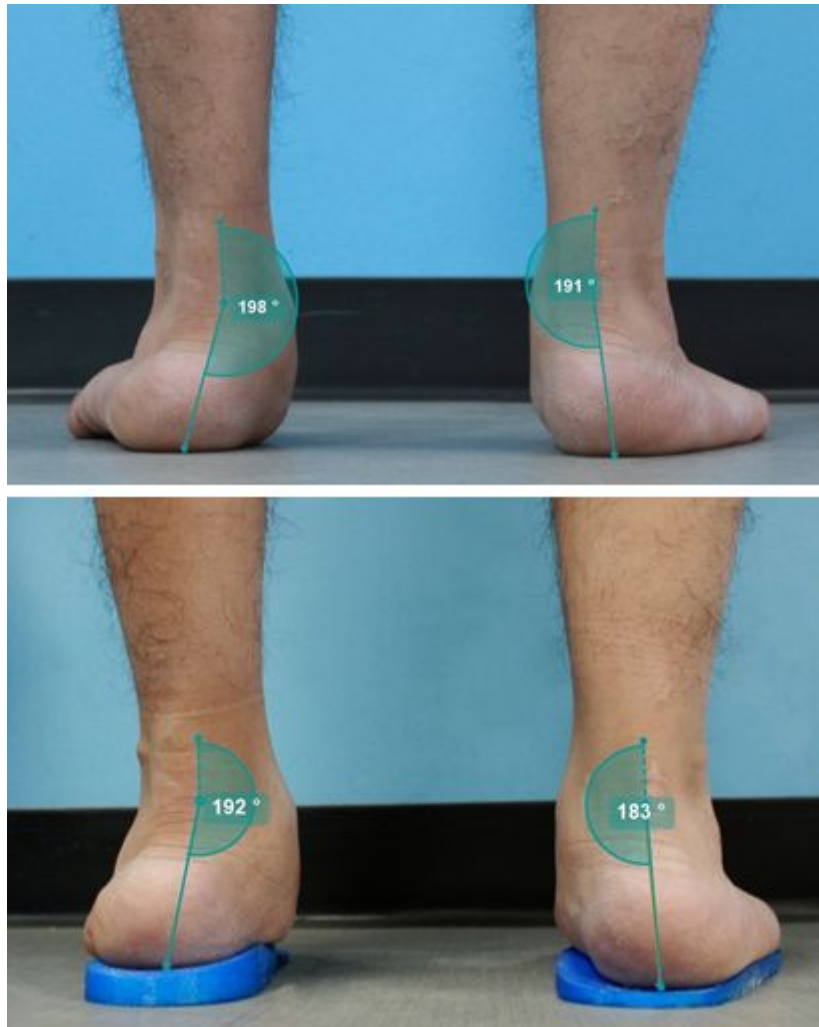


Figura 46. Ángulos entre talón y tobillo con y sin ayuda técnica

9.3.2 Órtesis tobillo-pie

En la Figura 47 se pueden observar dos imágenes. La imagen del lado izquierdo muestra la posición del pie del paciente al estar completamente acostado. El ángulo que se forma entre el pie del paciente con respecto al piso es de 129° ; se observa que el pie se encuentra caído y es por esto que el paciente presenta dolor al levantarse debido a la fascitis plantar que presenta. La imagen del lado derecho muestra el pie del paciente utilizando la órtesis tobillo-pie desarrollada; el ángulo que se forma entre el pie el piso es de 118° . Al observar las imágenes se concluye que el uso de la ayuda técnica corrigió la caída del pie alrededor de unos 11° . Con esto se comprueba que la órtesis sirve para mantener la postura correcta del pie durante la noche y así evitar tensión en la fascia plantar. Al evitar dicha tensión se evita la inflamación de los tejidos y elimina el dolor presentado.



Figura 47. Ángulos entre el pie y el piso.

9.4 Análisis de movimiento

Las gráficas presentadas a continuación muestran los ángulos entre los segmentos seleccionados para el análisis: muslo-pierna y pierna-planta del pie. Es decir, las gráficas permiten observar los ángulos que producen las rodillas y los pies durante un ciclo completo de la marcha. De esta manera se pudo llevar a cabo el análisis de movimiento observando las variaciones de ángulos entre la marcha control y la marcha del paciente. Además el análisis permitió detectar las diferencias presentadas sin el uso de ayudas técnicas y con el uso de ayudas técnicas. Al observar las gráficas pudimos determinar si las ayudas técnicas fueron efectivas y ayudaron a corregir la marcha del paciente.

9.4.1 Marcha Control

Las primeras gráficas obtenidas fueron aquellas correspondientes a la marcha de un individuo que no presentaba malformación en sus pies. Dichas gráficas muestran datos que entran dentro del rango de lo que se considera una marcha normal. Se decidió denominar estos datos como la marcha control ya que sería nuestro punto de referencia y fue la que utilizamos para comparar contra la marcha del paciente que si presentaba malformación en sus pies.

9.4.1.1 Ángulo de Rodilla Derecha-Pie Derecho

En la Figura 48 se presenta una gráfica que muestra los ángulos de la rodilla y el pie durante diferentes fases de la marcha. En la gráfica se observan dos líneas de datos; la línea azul hace referencia al ángulo de la rodilla derecha mientras que la línea verde hace referencia al ángulo del pie derecho. También se puede observar que los cuadros por segundos se encuentran resaltados de diferentes colores para representar las dos diferentes fases de la marcha. La fase de apoyo se representa con el color azul mientras que la fase de balanceo se representa con el

color rojo. Los datos presentados en la gráfica son aquellos correspondientes a un ciclo completo de la marcha.



Figura 48. Ángulo de Rodilla derecha y Pie derecho de Marcha Control

9.4.1.2 Ángulo de Rodilla Izquierda-Pie Izquierdo

En la Figura 49 se presenta una gráfica que muestra los ángulos de la rodilla y el pie durante diferentes fases de la marcha. En la gráfica se observan dos líneas de datos; la línea azul hace referencia al ángulo de la rodilla izquierda mientras que la línea verde hace referencia al ángulo del pie izquierdo. También se puede observar que los cuadros por segundos se encuentran resaltados de diferentes colores para representar las dos diferentes fases de la marcha. La fase de apoyo se representa con el color azul mientras que la fase de balanceo se representa con el color rojo. Los datos presentados en la gráfica son aquellos correspondientes a un ciclo completo de la marcha.



Figura 49. Ángulo de Rodilla Izquierda y Pie izquierdo de Marcha Control

9.4.1.3 Barra desplazadora y animación

En la Figura 50 se pueden observar las dos gráficas explicadas anteriormente. En la parte superior se encuentran las gráficas con una barra negra que se va desplazando a lo largo de los datos para ir indicando el progreso de la animación. La animación, que se encuentra en la parte inferior, muestra ambas extremidades inferiores. La pierna derecha es la de color amarillo mientras que la pierna izquierda es la de color azul. Justo en el punto donde está localizada la barra desplazadora en la gráfica superior izquierda se puede observar el ángulo que realiza la rodilla. El valor que se indica en ese punto se encuentra entre 120-140 grados. Si comparamos estos datos con la animación, podemos verificar que el movimiento coincide con el valor de ángulo que muestra la gráfica. Por otro lado, si observamos la gráfica “Rodilla izquierda-Pie izquierdo” la barra desplazadora indica que la articulación de la rodilla se encuentra a un ángulo entre 160-180 grados. De la misma manera, si comparamos el valor del ángulo con la animación podemos verificar que sí coinciden. La rodilla izquierda se encuentra recta mientras que la rodilla derecha se flexiona y los ángulos que realiza la articulación son coherentes con los expresados en ambas gráficas.

Cabe mencionar que la ventana que se observa en la Figura 50, la cual incluye la barra desplazadora y la animación, también aparecerá cuando se este analizando la marcha del paciente.

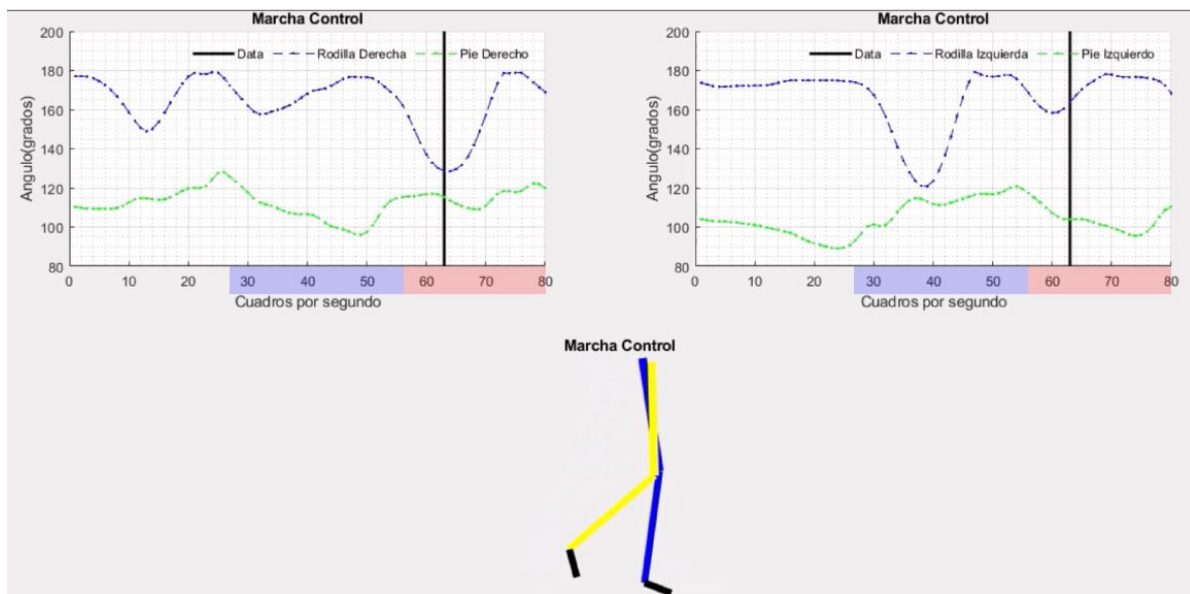


Figura 50. Resultados de ángulos y animación de la Marcha Control

9.4.2 Marcha del Paciente Semana 0

Las otras gráficas obtenidas fueron aquellas correspondientes a la marcha del paciente que presentaba malformación en los pies impidiéndole realizar una marcha normal. Dichas gráficas son las que tuvimos que comparar con la marcha control para observar las variaciones en los ángulos y determinar las diferencias que existían entre ambas. La gráficas de la marcha del paciente semana 0 hacen referencia a la marcha del paciente sin el uso de las ayudas técnicas desarrolladas.

9.4.2.1 Ángulo de Rodilla Derecha-Pie Derecho

En la Figura 51 se presenta una gráfica que muestra los ángulos de la rodilla y el pie durante diferentes fases de la marcha. En la gráfica se observan dos líneas de datos; la línea roja hace referencia al ángulo de la rodilla derecha mientras que la línea celeste hace referencia al ángulo del pie derecho. También se puede observar que los cuadros por segundos se encuentran resaltados de diferentes colores para representar las dos diferentes fases de la marcha. La fase de apoyo se representa con el color azul mientras que la fase de balanceo se representa con el color rojo. Los datos presentados en la gráfica son aquellos correspondientes a un ciclo completo de la marcha.

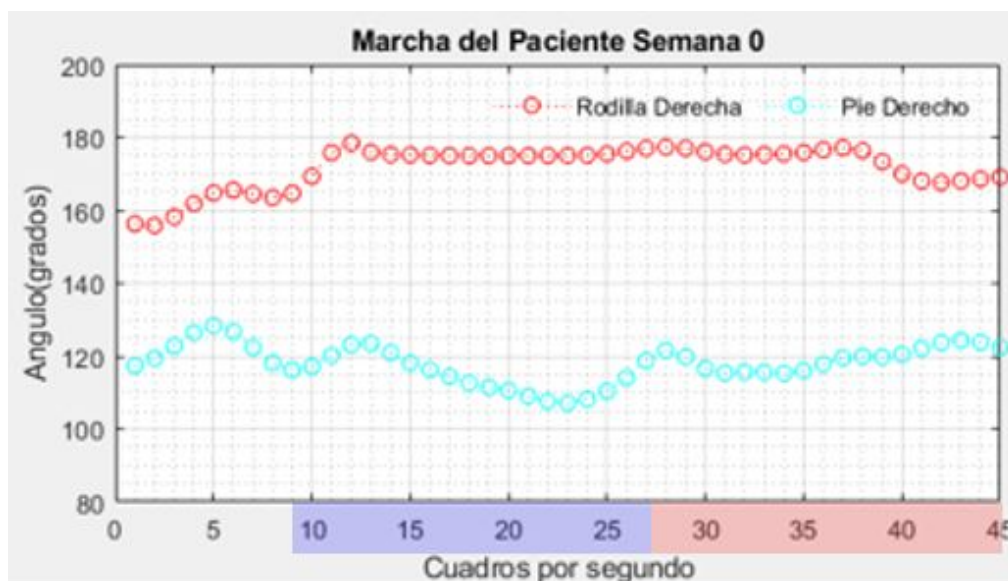


Figura 51. Ángulo de Rodilla derecha y Pie derecho de Marcha del Paciente Semana 0

9.4.2.2 Ángulo de Rodilla Izquierda-Pie Izquierdo

En la Figura 52 se presenta una gráfica que muestra los ángulos de la rodilla y el pie durante diferentes fases de la marcha. En la gráfica se observan dos líneas de datos; la línea roja hace referencia al ángulo de la rodilla izquierda mientras que la línea celeste hace referencia al ángulo del pie izquierdo. También se puede observar que los cuadros por segundos se encuentran resaltados de diferentes colores para representar las dos diferentes fases de la marcha. La fase de apoyo se representa con el color azul mientras que la fase de balanceo se representa con el color rojo. Los datos presentados en la gráfica son aquellos correspondientes a un ciclo completo de la marcha.

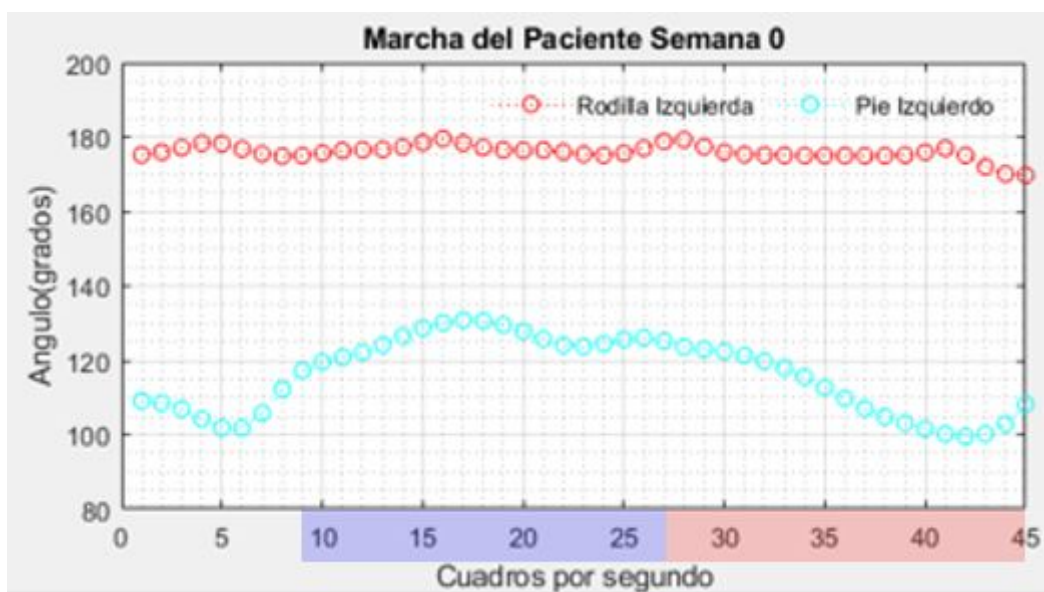


Figura 52. Ángulo de Rodilla izquierda y Pie izquierdo de Marcha del Paciente Semana 0

Las gráficas presentadas aparecerán en una misma ventana junto con una barra desplazadora indicando el progreso de los datos así como también una animación de la marcha del paciente semana 0. Por cuestiones de repetitividad, se omite la explicación de dicha ventana ya que fue explicada anteriormente para la marcha control y cumple con las mismas características.

9.4.3 Comparación Marcha control - Marcha del paciente semana 0

En la Figura 53 se puede observar la comparación realizada entre la marcha control y la marcha paciente semana 0. En el nivel superior se encuentran las gráficas de la marcha control y en el nivel inferior las de la marcha del paciente semana 0; cada una con su respectiva división entre la fase de apoyo (azul) y la fase de balanceo (rojo). Se puede observar que los ángulos presentados en la marcha del

paciente fueron muy diferentes a los presentados en las gráficas de marcha control ya que los movimientos del paciente eran mucho más limitados que los del individuo de la marcha control. La marcha del paciente semana 0 muestra ángulos que representan muy poca flexión en las articulaciones durante un ciclo completo de la marcha. Una de las diferencias más notorias se puede observar en el ángulo de la rodilla derecha al comenzar la marcha. En la marcha control, la pierna está completamente en extensión y la rodilla derecha alcanza un ángulo de casi 180°. En la marcha del paciente semana 0, la pierna no se encuentra completamente en extensión ya que la rodilla derecha inicia con un ángulo de 160°. Otra diferencia que se encuentre entre ambas marchas es el número de cuadros por segundo en un ciclo. La marcha control completa un ciclo de la marcha en 53 cuadros por segundo mientras que la marcha del paciente completa un ciclo de la marcha en 36 cuadros por segundo. Esto quiere decir que los pasos de la marcha del paciente son muy cortos a comparación de los pasos de la marcha control.

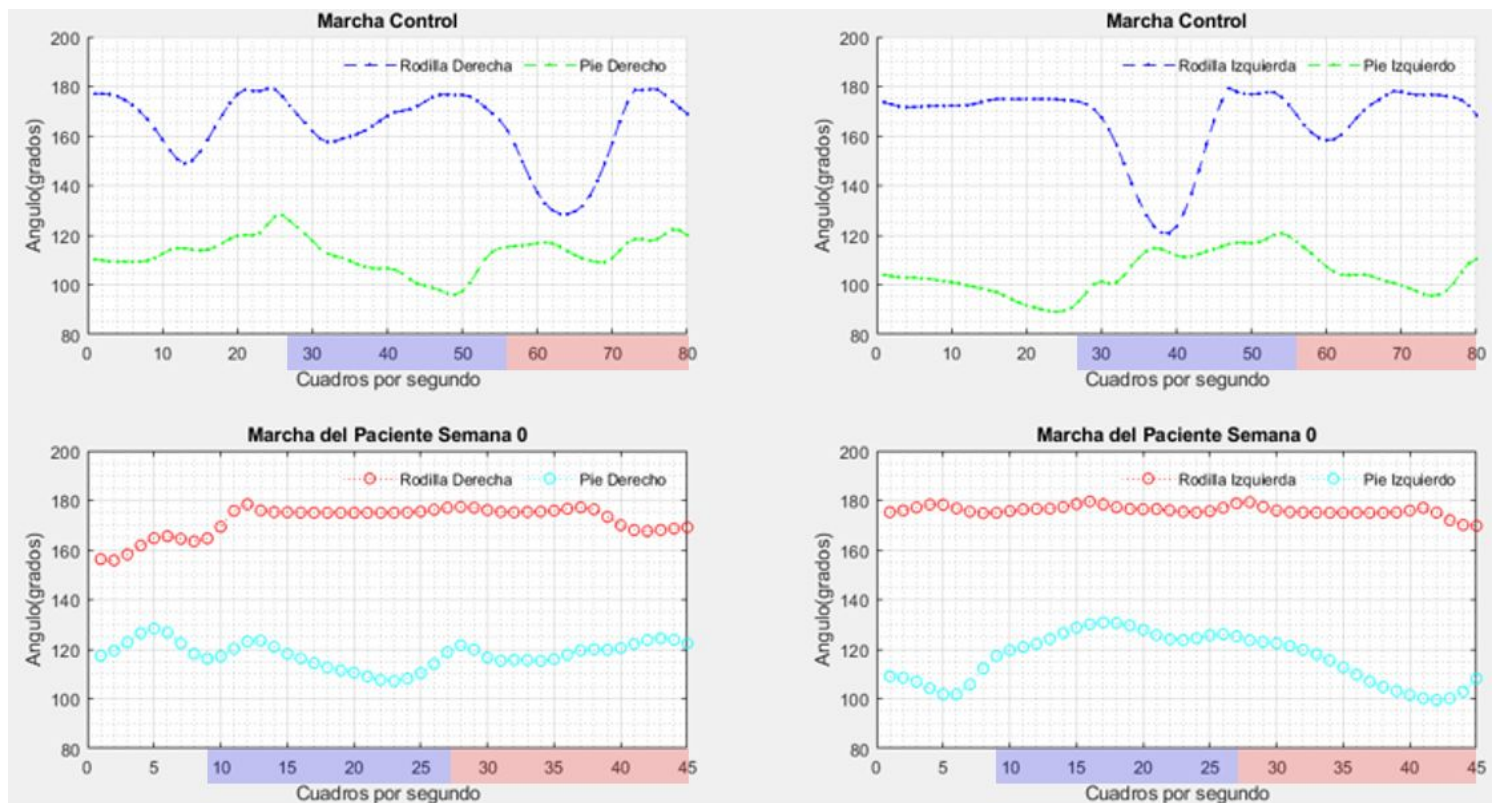


Figura 53. Comparativa de ángulos de Marcha Control y Marcha del Paciente Semana 0

9.4.4 Comparación Marcha del paciente semana 0 - semana 1

En la Figura 54 se puede observar la comparación realizada entre la marcha paciente semana 0 y la marcha paciente semana 1. En el nivel superior se encuentran las gráficas de la marcha paciente semana 0 y en el nivel inferior las de

la marcha del paciente semana 1; cada una con su respectiva división entre la fase de apoyo(azul) y la fase de balanceo(roja). Esta comparación se realizó con el fin de observar los cambios que hubo en la marcha del paciente a una semana de uso de las plantillas ortopédicas. Uno de los cambios que se observa en al iniciar la marcha, el ángulo de la rodilla es diferente. En la marcha del paciente semana 0 comienza la rodilla derecha con un ángulo de 160° mientras que en la marcha del paciente semana 1 comienza la rodilla derecha con un ángulo de aproximadamente 180° . Este último dato es muy similar al de la marcha control por lo tanto observamos que las plantillas si estaban cumpliendo correctamente su función. El cambio de ángulo de 160° a 180° se debió a que la altura con la que fue diseñada la plantilla izquierda permitió nivelar ambas piernas y lograr la completa extensión de ellas al iniciar la marcha. La pierna derecha logró estar extendida ya que era de la misma longitud que la pierna izquierda gracias a la altura que se le añadió a la plantilla. Otro cambio que pudimos observar fue en la cantidad de cuadros por segundo. En la marcha del paciente semana 0 se tiene 45 cuadros por segundo y la marcha del paciente semana 1 se tiene 66 cuadros por segundo, esto quiere decir que las plantillas ortopédicas están haciendo que el paciente de pasos más largos. Lo que se busca es tener una marcha alrededor de 80 cuadros por segundo ya que así se tiene la marcha control. También se observa mayor flexión en las articulaciones tanto de la rodilla izquierda como del pie izquierdo durante la marcha del paciente semana 1.

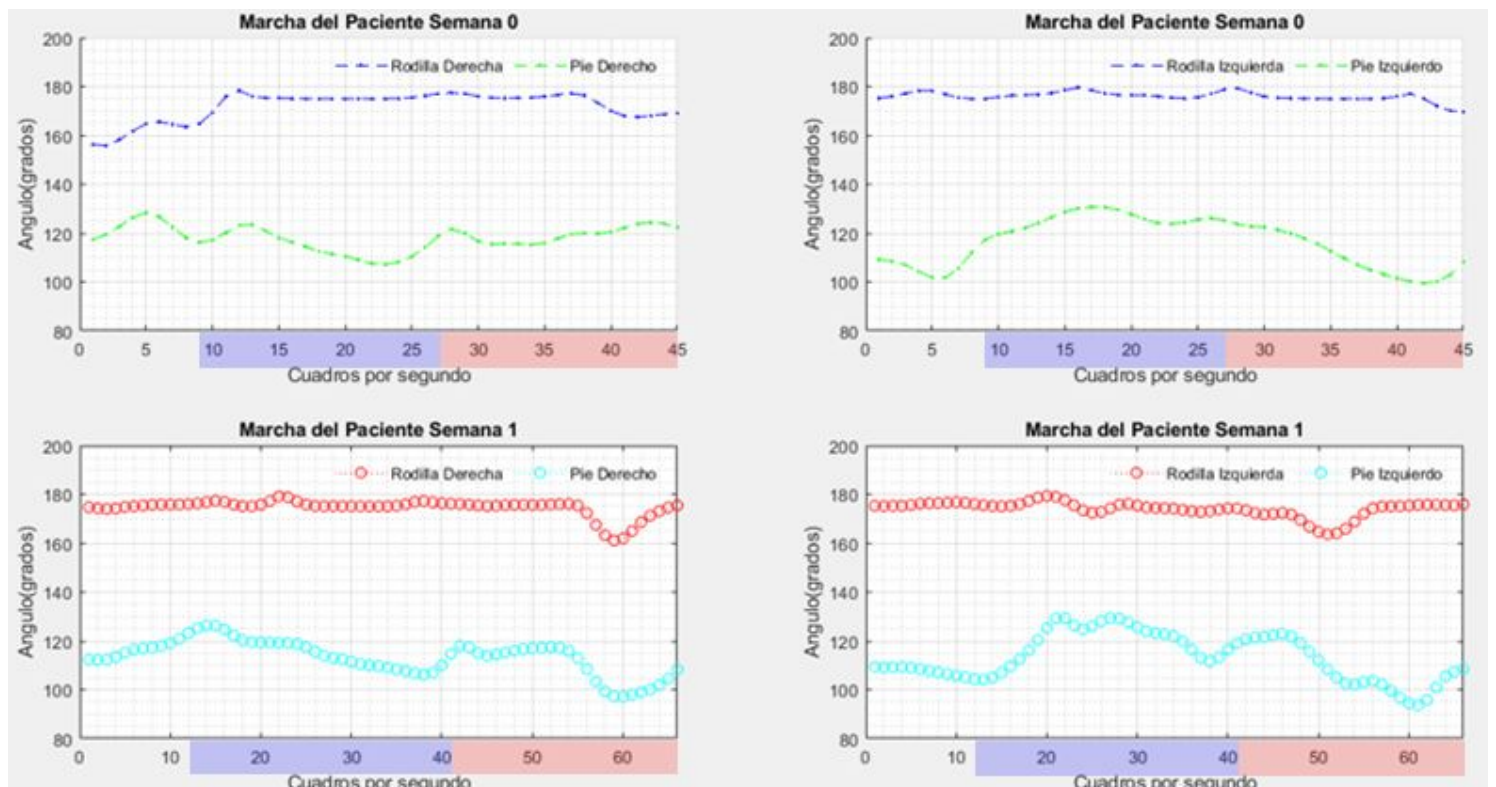


Figura 54. Comparativa de ángulos de Marcha del Paciente Semana 0 y Marcha del Paciente Semana 1.

9.4.5 Comparación Marcha del paciente semana 1 - semana 2

En la Figura 55 se puede observar la comparación realizada entre la marcha paciente semana 1 y la marcha paciente semana 2. En el nivel superior se encuentran las gráficas de la marcha paciente semana 1 y en el nivel inferior las de la marcha del paciente semana 2; cada una con su respectiva división entre la fase de apoyo(azul) y la fase de balanceo(roja). Esta comparación se realizó con el fin de observar los cambios que hubo en la marcha del paciente a dos semana de uso de las plantillas ortopédicas. Uno de los cambios más notables que se observa es que el paciente tuvo mucha más flexión en sus rodillas en la semana 2. Este último dato es más similar al de la marcha control por lo tanto observamos que las plantillas si estaban cumpliendo correctamente su función. Otro cambio que pudimos observar fue en la cantidad de cuadros por segundo. En la marcha del paciente semana 1 se tiene 66 cuadros por segundo y la marcha del paciente semana 2 se tiene 50 cuadros por segundo, esto quiere decir que se redujo nuevamente la cantidad de cuadros por segundos, pero la ventaja es que se está observando más flexión en sus rodillas. Lo que se busca es tener una marcha alrededor de 80 cuadros por segundo ya que así se tiene la marcha control.

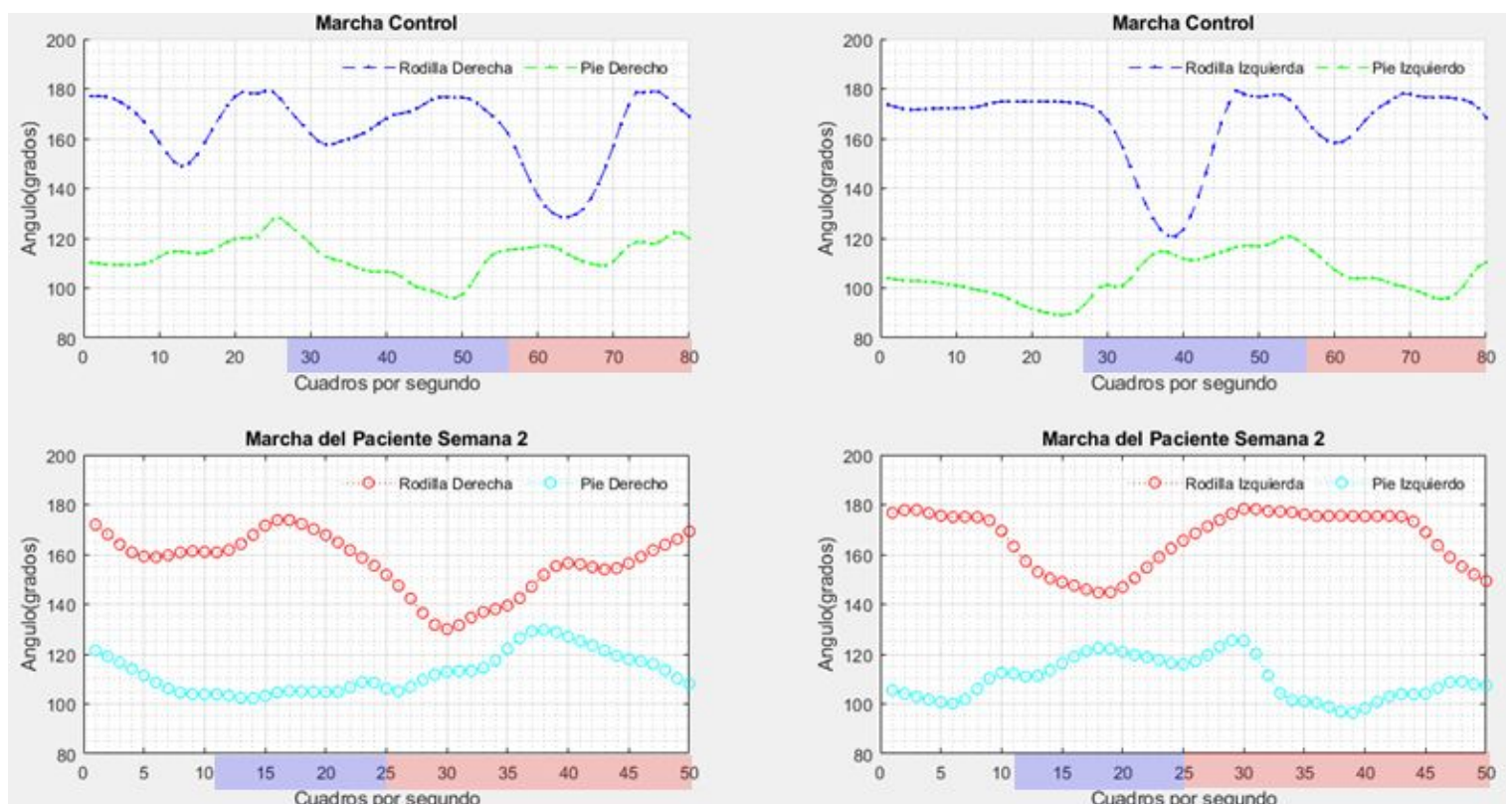


Figura 55. Comparativa de ángulos de Marcha del Paciente Semana 1 y Marcha del Paciente Semana 2.

10. Conclusión

Se cumplieron con las expectativas del proyecto desarrollado en sus partes esenciales: el diseño 3D de las ayudas técnicas y el desarrollo de la metodología para el sistema de análisis de movimiento. Para las ayudas técnicas, se debe considerar realizar ajustes en el futuro ya que al paso del tiempo se espera que las plantillas ortopédicas realicen cambios tanto funcionales como estructurales y por lo tanto se necesitará desarrollar unas plantillas con diferente diseño para que se adapten correctamente a las nuevas necesidades del pie del paciente. El desarrollo de la metodología para el análisis de movimiento en este caso fue utilizada para adquirir los ángulos de las articulaciones de los miembros inferiores. Sin embargo, la metodología desarrollada puede utilizarse para adquirir ángulos de cualquier articulación del cuerpo humano y para validar diferentes ayudas técnicas aparte de las diseñadas en este proyecto con el fin de identificar si le brindaron beneficios al paciente.

A lo largo del proyecto se presentaron distintos inconvenientes donde el proyecto se complicó; una de ellas fue el escaneo de los pies del paciente mediante la fotogrametría. Sin embargo, el uso de un molde de yeso de los pies del paciente facilitó en muchos aspectos el proceso. También, al ser un proyecto con una metodología extensa la oportunidad de realizar las pruebas finales fueron muy pocas debido a falta de tiempo. Sin embargo, en las pocas pruebas realizadas se pueden observar cambios positivos en el paciente. Los resultados obtenidos comprueban la efectividad de las ayudas técnicas y el impacto positivo que se tuvo con respecto a la postura y estabilidad del paciente. Se espera que con el uso constante de las ayudas técnicas desarrolladas se vean cambios más significativos en la marcha del paciente.

Así como se desarrollaron las plantillas ortopédicas y la órtesis tobillo-pie para la alineación de segmentos y corrección de la postura, se debe además considerar someter al paciente a rehabilitación debido a que presenta espasticidad en sus músculos. La rehabilitación podría permitir obtener mejores resultados de la marcha del paciente y lograr aproximarse un poco más a los valores de la marcha control. Además, se sugiere darle seguimiento al paciente realizando evaluaciones de su marcha una vez al mes para así observar la evolución que presenta.

11. Acuerdo cliente-alumno

Se buscará cubrir con fondos internos la adquisición de los recursos necesarios, más si no se consiguen, correría por nuestra cuenta.

12. Referencia bibliográficas

3D Collective. (2018, April 17). Guia de Fotogrametría. Retrieved from <https://3dcollective.es/fotogrametria/>

Acosta, P. (n.d.). Biomecánica de la marcha . Retrieved from <https://fisiointegracion.files.wordpress.com/2010/05/biomecanica-marcha.pdf>

Agisoft. (2019). Agisoft Metashape User Manual . Retrieved from https://www.agisoft.com/pdf/metashape-pro_1_5_en.pdf

AliMed. (2014, Noviembre 17). AFOs. Obtenido el 15 de Julio, 2019 de <https://www.alimed.com/afo-info-blog/>

Afanasyev, Ilya & Magid, Evgeni & Danilov, Igor & Gabbasov, Bulat. (2015). Toward a human-like biped robot gait: Biomechanical analysis of human locomotion recorded by Kinect-based Motion Capture system. 10.1109/ISMA.2015.7373477.

ASEM Galicia. (2008). *Las ayudas técnicas*[PDF].

Bohórquez Ávila , C. A. (n.d.). Estudio sobre la marcha humana . *Revista Ingenio Libre* , 55–62. Retrieved from <http://www.unilibre.edu.co/revistaingeniolibre/revista5/articulos/Estudio-sobre-la-marcha-humana-1.pdf>

Cerda A., L. (2013, Mayo 7). *Evaluación del paciente con trastorno de la marcha*[PDF]. Santiago.

De Pablos, J. (2015, Junio 13). *Dismetrías de los miembros inferiores*[PDF]. Pamplona.

InformedHealth. (2018, Junio 28). Foot deformities. Obtenido el 15 de Julio, 2019 de <https://www.ncbi.nlm.nih.gov/books/NBK513132/>

Instituto Nacional de Estadística y Geografía (México). La discapacidad en México, datos al 2014. Obtenido el 1 de Noviembre, 2019 de <http://coespo.qroo.gob.mx/Descargas/doc/DISCAPACITADOS/ENADID 2014.pdf>

L. L. Gómez Echeverry, A. M. Jaramillo Henao, M. A. Ruiz Molina, S. M. Velásquez Restrepo, C. A. Páramo Velásquez, G. J. Silva Bolívar, “Human motion capture and analysis systems: a systematic review”, *Prospectiva*, Vol 16, N° 2, 24-34, 2018.

López-Terradas, J. M. (2008). *Alteraciones de la marcha*[PDF]. Madrid.

Mayo Clinic. (2018, March 7). Fascitis plantar. Retrieved November 9, 2019, from <https://www.mayoclinic.org/es-es/diseases-conditions/plantar-fasciitis/symptoms-causes/syc-20354846>.

Medline Plus. (n.d.). Fascitis plantar. Retrieved November 9, 2019, from <https://medlineplus.gov/spanish/ency/article/007021.htm>.

Miller, F. (2005). *Cerebral Palsy* . Retrieved from https://books.google.com.mx/books?id=V_p50E-Up7IC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=snippet&q=locomotor&f=false

Ministerio de Salud. (2018, Mayo 3). Salud y su Jornada de Ayudas Técnicas. Obtenido el 15 de Julio, 2019 de <https://www.saludohiggins.cl/?p=9770>

Organización Mundial de la Salud. (2018, Mayo 18). Tecnología de asistencia. Obtenido el 15 de Julio, 2019 de <https://www.who.int/es/news-room/fact-sheets/detail/assistive-technology>

Ortopedia Aeropuerto. (S.F.). Patologías infantiles del pie. Obtenido el 15 de Julio, 2019 de http://www.ortopediaaeropuerto.com/attachments/article/155/Patologias%20del%20pie%20infant_oa.pdf

Ortopedia. (2013, Marzo 27). Órtesis de columna vertebral y tronco. Obtenido el 15 de Julio, 2019 de <https://www.ortopediaonlinetarancon.com/ortesis-de-columna-vertebral-y-tronco/>

Osorio, J. H., & Valencia, M. H. (2013, Mayo 7). *Bases para el entendimiento del proceso de la marcha humana*[PDF]. Caldas, Colombia.

PRIM. (2015). *Guía de Podología y sus indicaciones ortésicas*[PDF]. Madrid.

Protésica. (n.d.). Órtesis larga con articulaciones tipo KAFO. Obtenido el 15 de Julio, 2019 de <https://protesica.com.co/portfolio-item/ortesis-kafo/>

Romero Rubio, M. T., Dra. (2019, Marzo 28). Deformidades de los pies en los niños. Obtenido el 15 de Julio, 2019 de <https://www.webconsultas.com/bebes-y-ninos/afecciones-tipicas-infantiles/deformidades-de-los-pies-en-los-ninos-13971>

Rose, J., & Gamble, J. G. (2006). *Human Walking* (3rd ed.). Philadelphia: Lippincott Williams and Wilkins.

Rovira, I. (2019, Enero 31). Partes y huesos del pie humano: Anatomía y funciones. Obtenido el 15 de Julio, 2019 de <https://viviendolasalud.com/cuerpo-y-mente/partes-huesos-del-pie>

Shen, A.W., & Ting, T. (2014). Marker-less motion capture for biomechanical analysis using the Kinect sensor.

Sumba , C., Cuenca, M. del C., & Delgado , K. M. (2013). Diseño e implementación de un sistema para el análisis de movimiento humano usando sensores Kinect, 1–102.

13. Código de Honor

Damos nuestra palabra de haber realizado este escrito con integridad académica.

14. Anexos

Anexo 1 - Código desarrollado en software Matlab para el cálculo de ángulos entre segmentos

```
%Calculo de angulos entre segmentos con matriz importada del software IPI
%SOFT
clear all
clc
close all
load('caminandoground_export.mat')%se carga la matriz
n = length(JointsData.RThigh.Coordinate);%Cantidad de frames que se grabaron

%Este FOR es para guardar cada una de las coordenadas x,y,z del RShin en
%otro Struct
for i = 1:n
    JointsData.RShin.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RShin.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RShin.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RShin.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

%Este FOR es para guardar cada una de las coordenadas x,y,z del LShin en
%otro Struct
for i = 1:n
    JointsData.LShin.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LShin.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
```

```

    Marcha.LShin.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LShin.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RThigh en
%otro Struct

```

for i = 1:n
    JointsData.RThigh.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RThigh.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RThigh.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RThigh.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LThigh en
%otro Struct

```

for i = 1:n
    JointsData.LThigh.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LThigh.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LThigh.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames

```

```

C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
Marcha.LThigh.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RFoot en
%otro Struct
for i = 1:n
    JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RFoot.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RFoot.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RFoot.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LFoot en
%otro Struct
for i = 1:n
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LFoot.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LFoot.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LFoot.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

end

%Este FOR es para guardar cada una de las coordenadas x,y,z del RToe en
%otro Struct
for i = 1:n
    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RToe.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RToe.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RToe.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LToe en
%otro Struct
for i = 1:n
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LToe.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LToe.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LToe.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end
%Para obtener el angulo entre dos segmentos primero tuvimos que contruir
%los segmentos utilizando las coordenadas, haciendo la operacion

```

```
%RShin-Rthigh,RShin-RFoot,LShin-Lthigh,LShin-LFoot y tambien obtener la magnitud de
estos
%segmentos
```

```
%Este FOR es para obtener el segmento Rshin-Rthigh
```

```
for i = (1:n)
```

```
    A=Marcha.RShin.CoordenadasP1(i,1)-Marcha.RThigh.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
```

```
    Marcha.RSRT.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
```

```
    B=Marcha.RShin.CoordenadasP1(i,2)-Marcha.RThigh.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
```

```
    Marcha.RSRT.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
```

```
    C=Marcha.RShin.CoordenadasP1(i,3)-Marcha.RThigh.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
```

```
    Marcha.RSRT.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
```

```
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
```

```
    Marcha.RSRT.MagnitudP1(i,1)=D;%Se guardo en otro Struct
```

```
    i=i+1;
```

```
end
```

```
%Este FOR es para obtener el segmento Lshin-Lthigh
```

```
for i = (1:n)
```

```
    A=Marcha.LShin.CoordenadasP1(i,1)-Marcha.LThigh.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
```

```
    Marcha.LSLT.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
```

```
    B=Marcha.LShin.CoordenadasP1(i,2)-Marcha.LThigh.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
```

```
    Marcha.LSLT.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
```

```
    C=Marcha.LShin.CoordenadasP1(i,3)-Marcha.LThigh.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
```

```
    Marcha.LSLT.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
```

```
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
```

```
    Marcha.LSLT.MagnitudP1(i,1)=D;%Se guardo en otro Struct
```

```
    i=i+1;
```

```
end
```

```
%Este FOR es para obtener el segmento Rshin-RFoot
```

```
for i = (1:n)
```

```
    A=Marcha.RShin.CoordenadasP1(i,1)-Marcha.RFoot.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
```

```
    Marcha.RSRF.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
```

```
    B=Marcha.RShin.CoordenadasP1(i,2)-Marcha.RFoot.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
```

```
    Marcha.RSRF.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
```

```

C=Marcha.RShin.CoordenadasP1(i,3)-Marcha.RFoot.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
Marcha.RSRF.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
Marcha.RSRF.MagnitudP1(i,1)=D;%Se guardo en otro Struct
i=i+1;
end

```

%Este FOR es para obtener el segmento LShin-LFoot

```
for i = (1:n)
```

```

A=Marcha.LShin.CoordenadasP1(i,1)-Marcha.LFoot.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
Marcha.LSLF.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
B=Marcha.LShin.CoordenadasP1(i,2)-Marcha.LFoot.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
Marcha.LSLF.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
C=Marcha.LShin.CoordenadasP1(i,3)-Marcha.LFoot.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
Marcha.LSLF.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
Marcha.LSLF.MagnitudP1(i,1)=D;%Se guardo en otro Struct
i=i+1;
end

```

%Este FOR es para obtener el segmento RFoot-RToe

```
for i = (1:n)
```

```

A=Marcha.RFoot.CoordenadasP1(i,1)-Marcha.RToe.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
Marcha.RFRT.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
B=Marcha.RFoot.CoordenadasP1(i,2)-Marcha.RToe.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
Marcha.RFRT.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
C=Marcha.RFoot.CoordenadasP1(i,3)-Marcha.RToe.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
Marcha.RFRT.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
Marcha.RFRT.MagnitudP1(i,1)=D;%Se guardo en otro Struct
i=i+1;
end

```

%Este FOR es para obtener el segmento LFoot-LToe

```
for i = (1:n)
```

```

A=Marcha.LFoot.CoordenadasP1(i,1)-Marcha.LToe.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
Marcha.LFLT.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
B=Marcha.LFoot.CoordenadasP1(i,2)-Marcha.LToe.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
Marcha.LFLT.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
C=Marcha.LFoot.CoordenadasP1(i,3)-Marcha.LToe.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
Marcha.LFLT.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
Marcha.LFLT.MagnitudP1(i,1)=D;%Se guardo en otro Struct
i=i+1;
end

```

```

%Este FOR es para obtener el segmento RFoot-RShin
for i = (1:n)
A=Marcha.RFoot.CoordenadasP1(i,1)-Marcha.RShin.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
Marcha.RFRS.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
B=Marcha.RFoot.CoordenadasP1(i,2)-Marcha.RShin.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
Marcha.RFRS.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
C=Marcha.RFoot.CoordenadasP1(i,3)-Marcha.RShin.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
Marcha.RFRS.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
Marcha.RFRS.MagnitudP1(i,1)=D;%Se guardo en otro Struct
i=i+1;
end

```

```

%Este FOR es para obtener el segmento LFoot-LShin
for i = (1:n)
A=Marcha.LFoot.CoordenadasP1(i,1)-Marcha.LShin.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
Marcha.LFLS.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
B=Marcha.LFoot.CoordenadasP1(i,2)-Marcha.LShin.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
Marcha.LFLS.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
C=Marcha.LFoot.CoordenadasP1(i,3)-Marcha.LShin.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
Marcha.LFLS.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
Marcha.LFLS.MagnitudP1(i,1)=D;%Se guardo en otro Struct

```



```
    i=i+1;
end
```

%Ya teniendo los la magnitud de los segmentos ya hechos, seguimos con la
%formula multiplicando las magnitudes de los dos segmentos

%Este FOR es para obtener el producto de las magnitudes RSRF*RSRT
for i = (1:n)

```
    A=Marcha.RSRF.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    B=Marcha.RSRT.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    C=A*B;%Multiplicacion de las dos magnitudes
```

```
    Marcha.R1PMP1(i,1)=C;%El resultado se guardo en otro Struct
```

```
    i=i+1;
end
```

%Este FOR es para obtener el producto de las magnitudes LSLF*LSLT
for i = (1:n)

```
    A=Marcha.LSLF.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    B=Marcha.LSLT.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    C=A*B;%Multiplicacion de las dos magnitudes
```

```
    Marcha.L1PMP1(i,1)=C;%El resultado se guardo en otro Struct
```

```
    i=i+1;
end
```

%Este FOR es para obtener el producto de las magnitudes RFRT*RFRS
for i = (1:n)

```
    A=Marcha.RFRS.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    B=Marcha.RFRT.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    C=A*B;%Multiplicacion de las dos magnitudes
```

```
    Marcha.R2PMP1(i,1)=C;%El resultado se guardo en otro Struct
```

```
    i=i+1;
end
```

%Este FOR es para obtener el producto de las magnitudes LFLT*LFLS
for i = (1:n)

```
    A=Marcha.LFLS.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    B=Marcha.LFLT.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    C=A*B;%Multiplicacion de las dos magnitudes
```

```

    Marcha.L2PMP1(i,1)=C;%El resultado se guardo en otro Struct
    i=i+1;
end

```

%Ya teniendo la multiplicacion de las magnitudes de los segmentos, seguimos con la %formula realizando el producto punto entre los dos segmentos

%Este FOR es para obtener el pructo punto entre los dos segmentos
%RSRT*.RSRF

```

for i = (1:n)
    A=Marcha.RSRT.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RSRF.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
    Marcha.R1PPP1(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener el pructo punto entre los dos segmentos
%LSLT*.LSLF

```

for i = (1:n)
    A=Marcha.LSLT.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.LSLF.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
    Marcha.L1PPP1(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener el pructo punto entre los dos segmentos
%RFRT*.RFRS

```

for i = (1:n)
    A=Marcha.RFRT.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RFRS.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
    Marcha.R2PPP1(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener el pructo punto entre los dos segmentos
%LFLT*.LFLS

```

for i = (1:n)

```

```

A=Marcha.LFLT.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
B=Marcha.LFLS.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
Marcha.L2PPP1(i,1)=C;%Se guarda el resultado en otro Struct
i=i+1;
end

```

```

%Una Vez ya teniendo el resultado de la multiplzacion de las magnitudes y
%el producto punto de los segmentos seguimos con la formula que seria el
%coseno inverzo de el producto punto de los segmentos entre el producto de
%las magnitudes

```

```

%Este FOR es para obtener a en angulo en grados de la Pierna Derecha
for i = (1:n)
A=Marcha.R1PPP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
B=Marcha.R1PMP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
C=acosd(A/B);%Se caclula en angulo
Marcha.Angulo.RTRSRF1(i,1)=C;%Se guarda el resultado en otro Struct
i=i+1;
end

```

```

%Este FOR es para obtener a en angulo en grados de la Pierna Izquierda
for i = (1:n)
A=Marcha.L1PPP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
B=Marcha.L1PMP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
C=acosd(A/B);%Se caclula en angulo
Marcha.Angulo.LTSLF1(i,1)=C;%Se guarda el resultado en otro Struct
i=i+1;
end

```

```

%Este FOR es para obtener a en angulo en grados del pie derecho
for i = (1:n)
A=Marcha.R2PPP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
B=Marcha.R2PMP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
C=acosd(A/B);%Se caclula en angulo
Marcha.Angulo.RTRFRS1(i,1)=C;%Se guarda el resultado en otro Struct
i=i+1;
end

```

```

%Este FOR es para obtener a en angulo en grados del pie izquierdo
for i = (1:n)
    A=Marcha.L2PPP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    B=Marcha.L2PMP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    C=acosd(A/B);%Se caclula en angulo
    Marcha.Angulo.LTLFLS1(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

```

%%
% load('caminandomalground_export.mat')
% load('Carlos1_export.mat')
load('Carlos1_export1CICLO.mat')
n2 = length(JointsData.RThigh.Coordinate);%Cantidad de frames que se grabaron

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RShin en
%otro Struct
for i = 1:n2
    JointsData.RShin.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RShin.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RShin.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RShin.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LShin en
%otro Struct
for i = 1:n2
    JointsData.LShin.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X

```

```

    Marcha.LShin.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LShin.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LShin.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RThigh en

%otro Struct

for i = 1:n2

```

    JointsData.RThigh.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RThigh.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RThigh.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RThigh.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LThigh en

%otro Struct

for i = 1:n2

```

    JointsData.LThigh.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LThigh.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames

```

```

    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LThigh.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LThigh.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RFoot en
%otro Struct
for i = 1:n2
    JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RFoot.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RFoot.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RFoot.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LFoot en
%otro Struct
for i = 1:n2
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LFoot.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LFoot.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2

```

```

    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LFoot.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RToe en
%otro Struct
for i = 1:n2
    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RToe.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RToe.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RToe.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LToe en
%otro Struct
for i = 1:n2
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LToe.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LToe.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z

```

```

    Marcha.LToe.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Para obtener el angulo entre dos segmentos primero tuvimos que contruir
%los segmentos utilizando las coordenadas, haciendo la operacion
%RShin-Rthigh,RShin-RFoot,LShin-Lthigh,LShin-LFoot y tambien obtener la magnitud de
estos
%segmentos

```

```

%Este FOR es para obtener el segmento Rshin-Rthigh
for i = (1:n2)
    A=Marcha.RShin.CoordenadasP2(i,1)-Marcha.RThigh.CoordenadasP2(i,1);%Se hizo la
resta en coordenada X
    Marcha.RSRT.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.RShin.CoordenadasP2(i,2)-Marcha.RThigh.CoordenadasP2(i,2);%Se hizo la
resta en coordenada Y
    Marcha.RSRT.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.RShin.CoordenadasP2(i,3)-Marcha.RThigh.CoordenadasP2(i,3);%Se hizo la
resta en coordenada Z
    Marcha.RSRT.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
    Marcha.RSRT.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento Lshin-Lthigh
for i = (1:n2)
    A=Marcha.LShin.CoordenadasP2(i,1)-Marcha.LThigh.CoordenadasP2(i,1);%Se hizo la
resta en coordenada X
    Marcha.LSLT.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.LShin.CoordenadasP2(i,2)-Marcha.LThigh.CoordenadasP2(i,2);%Se hizo la
resta en coordenada Y
    Marcha.LSLT.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.LShin.CoordenadasP2(i,3)-Marcha.LThigh.CoordenadasP2(i,3);%Se hizo la
resta en coordenada Z
    Marcha.LSLT.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
    Marcha.LSLT.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento Rshin-RFoot

```



```

for i = (1:n2)
    A=Marcha.RShin.CoordenadasP2(i,1)-Marcha.RFoot.CoordenadasP2(i,1);%Se hizo la
    resta en coordenada X
    Marcha.RSRF.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.RShin.CoordenadasP2(i,2)-Marcha.RFoot.CoordenadasP2(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.RSRF.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.RShin.CoordenadasP2(i,3)-Marcha.RFoot.CoordenadasP2(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.RSRF.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados
    Marcha.RSRF.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener el segmento Lshin-LFoot

```

for i = (1:n2)
    A=Marcha.LShin.CoordenadasP2(i,1)-Marcha.LFoot.CoordenadasP2(i,1);%Se hizo la
    resta en coordenada X
    Marcha.LSLF.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.LShin.CoordenadasP2(i,2)-Marcha.LFoot.CoordenadasP2(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.LSLF.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.LShin.CoordenadasP2(i,3)-Marcha.LFoot.CoordenadasP2(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.LSLF.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados
    Marcha.LSLF.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener el segmento RFoot-RToe

```

for i = (1:n2)
    A=Marcha.RFoot.CoordenadasP2(i,1)-Marcha.RToe.CoordenadasP2(i,1);%Se hizo la
    resta en coordenada X
    Marcha.RFRT.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.RFoot.CoordenadasP2(i,2)-Marcha.RToe.CoordenadasP2(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.RFRT.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.RFoot.CoordenadasP2(i,3)-Marcha.RToe.CoordenadasP2(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.RFRT.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados

```

```

    Marcha.RFRT.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

%Este FOR es para obtener el segmento LFoot-LToe
for i = (1:n2)
    A=Marcha.LFoot.CoordenadasP2(i,1)-Marcha.LToe.CoordenadasP2(i,1);%Se hizo la
    resta en coordenada X
    Marcha.LFLT.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.LFoot.CoordenadasP2(i,2)-Marcha.LToe.CoordenadasP2(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.LFLT.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.LFoot.CoordenadasP2(i,3)-Marcha.LToe.CoordenadasP2(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.LFLT.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados
    Marcha.LFLT.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

%Este FOR es para obtener el segmento RFoot-RShin
for i = (1:n2)
    A=Marcha.RFoot.CoordenadasP2(i,1)-Marcha.RShin.CoordenadasP2(i,1);%Se hizo la
    resta en coordenada X
    Marcha.RFRS.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.RFoot.CoordenadasP2(i,2)-Marcha.RShin.CoordenadasP2(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.RFRS.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.RFoot.CoordenadasP2(i,3)-Marcha.RShin.CoordenadasP2(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.RFRS.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados
    Marcha.RFRS.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

%Este FOR es para obtener el segmento LFoot-LShin
for i = (1:n2)
    A=Marcha.LFoot.CoordenadasP2(i,1)-Marcha.LShin.CoordenadasP2(i,1);%Se hizo la
    resta en coordenada X
    Marcha.LFLS.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.LFoot.CoordenadasP2(i,2)-Marcha.LShin.CoordenadasP2(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.LFLS.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct

```

```

C=Marcha.LFoot.CoordenadasP2(i,3)-Marcha.LShin.CoordenadasP2(i,3);%Se hizo la
resta en coordenada Z
Marcha.LFLS.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
Marcha.LFLS.MagnitudP2(i,1)=D;%Se guardo en otro Struct
i=i+1;
end

```

%Ya teniendo los la magnitud de los segmentos ya hechos, seguimos con la
%formula multiplicando las magnitudes de los dos segmentos

```

%Este FOR es para obtener el producto de las magnitudes RSRF*RSRT
for i = (1:n2)
    A=Marcha.RSRF.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RSRT.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=A*B;%Multiplicacion de las dos magnitudes
    Marcha.R1PMP2(i,1)=C;%El resultado se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el producto de las magnitudes LSLF*LSLT
for i = (1:n2)
    A=Marcha.LSLF.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.LSLT.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=A*B;%Multiplicacion de las dos magnitudes
    Marcha.L1PMP2(i,1)=C;%El resultado se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el producto de las magnitudes RFRT*RFRS
for i = (1:n2)
    A=Marcha.RFRS.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RFRT.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=A*B;%Multiplicacion de las dos magnitudes
    Marcha.R2PMP2(i,1)=C;%El resultado se guardo en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener el producto de las magnitudes LFLT*LFLS

```

for i = (1:n2)
    A=Marcha.LFLS.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.LFLT.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=A*B;%Multiplicacion de las dos magnitudes
    Marcha.L2PMP2(i,1)=C;%El resultado se guardo en otro Struct
    i=i+1;
end

```

%Ya teniendo la multiplicacion de las magnitudes de los segmentos, seguimos con la
%formula realizando el producto punto entre los dos segmentos

```

%Este FOR es para obtener el producto punto entre los dos segmentos
%RSRT*.RSRF
for i = (1:n2)
    A=Marcha.RSRT.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RSRF.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
    Marcha.R1PPP2(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el pructo punto entre los dos segmentos
%LSLT*.LSLF
for i = (1:n2)
    A=Marcha.LSLT.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.LSLF.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
    Marcha.L1PPP2(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el pructo punto entre los dos segmentos
%RFRT*.RFRS
for i = (1:n2)
    A=Marcha.RFRT.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RFRS.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
    Marcha.R2PPP2(i,1)=C;%Se guarda el resultado en otro Struct

```

```
i=i+1;  
end
```

```
%Este FOR es para obtener el pructo punto entre los dos segmentos  
%LFLT*.LFLS  
for i = (1:n2)  
    A=Marcha.LFLT.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la  
operacion mas simple  
    B=Marcha.LFLS.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la  
operacion mas simple  
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot  
    Marcha.L2PPP2(i,1)=C;%Se guarda el resultado en otro Struct  
    i=i+1;  
end
```

```
%Una Vez ya teniendo el resultado de la multiplziacion de las magnitudes y  
%el producto punto de los segmentos seguimos con la formula que seria el  
%coseno inverzo de el producto punto de los segmentos entre el producto de  
%las magnitudes
```

```
%Este FOR es para obtener a en angulo en grados de la Pierna Derecha  
for i = (1:n2)  
    A=Marcha.R1PPP2(i,1);%Se guardo el estruct como una variable para hacer la operacion  
mas simple  
    B=Marcha.R1PMP2(i,1);%Se guardo el estruct como una variable para hacer la operacion  
mas simple  
    C=acosd(A/B);%Se caclula en angulo  
    Marcha.Angulo.RTRS RF2(i,1)=C;%Se guarda el resultado en otro Struct  
    i=i+1;  
end
```

```
%Este FOR es para obtener a en angulo en grados de la Pierna Izquierda  
for i = (1:n2)  
    A=Marcha.L1PPP2(i,1);%Se guardo el estruct como una variable para hacer la operacion  
mas simple  
    B=Marcha.L1PMP2(i,1);%Se guardo el estruct como una variable para hacer la operacion  
mas simple  
    C=acosd(A/B);%Se caclula en angulo  
    Marcha.Angulo.LTSLF2(i,1)=C;%Se guarda el resultado en otro Struct  
    i=i+1;  
end
```

```
%Este FOR es para obtener a en angulo en grados del pie derecho  
for i = (1:n2)  
    A=Marcha.R2PPP2(i,1);%Se guardo el estruct como una variable para hacer la operacion  
mas simple
```

```

    B=Marcha.R2PMP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    C=acosd(A/B);%Se caclula en angulo
    Marcha.Angulo.RTRFRS2(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener a en angulo en grados del pie izquierdo
for i = (1:n2)

```

```

    A=Marcha.L2PPP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    B=Marcha.L2PMP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    C=acosd(A/B);%Se caclula en angulo
    Marcha.Angulo.LTLFLS2(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

```

%% Grafica Marcha Control

```

```

figure('Name','Marcha Normal','NumberTitle','off');
fh = figure(1);
fh.WindowState = 'maximized';
t = 1:n;
subplot(2,2,1)
curve1 = animatedline('color','k','LineWidth', 2);
hold on
plot(t,Marcha.Angulo.RTRSRF1,'b--.')
```

%Se grafica la marcha del paciente 1 con una grafica de puntos de color azul

```

plot(t,Marcha.Angulo.RTRFRS1,'g--.')
```

%Se grafica la marcha del paciente 1 con una grafica de puntos de color verde

```

axis([0 80 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha Control')
legend({'Data','Rodilla Derecha','Pie Derecho'},'NumColumns',3)
legend('boxoff')
subplot(2,2,2)
curve2 = animatedline('color','k','LineWidth', 2);
hold on
plot(t,Marcha.Angulo.LTLSLF1,'b--.')
```

%Se grafica la marcha del paciente 1 con una grafica de puntos de color azul

```

plot(t,Marcha.Angulo.LTLFLS1,'g--.')
```

%Se grafica la marcha del paciente 1 con una grafica de puntos de color verde

```

axis([0 80 80 200])
grid on
grid minor
hold on
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha Control')
legend({'Data','Rodilla Izquierda','Pie Izquierdo'},'NumColumns',3)
legend('boxoff')
MusloRP1=plot3(0,0,0);
PiernaRP1=plot3(0,0,0);
MusloLP1=plot3(0,0,0);
PiernaLP1=plot3(0,0,0);
PieRP1=plot3(0,0,0);
PieLP1=plot3(0,0,0);

ib=0;
while ib<2
for ia=1:1:80
    pause(1/30)
    clearpoints(curve1)
    x=[ia ia];
    y=[0 ,270];
    addpoints(curve1, x, y);

    clearpoints (curve2)
    addpoints(curve2, x, y);

    subplot(2,2,[3,4])
    axis equal
    axis off
    hold on
    view(90,180)
    title('Marcha Control')

    delete(MusloRP1);
    delete(MusloLP1);
    delete(PiernaRP1);
    delete(PiernaLP1);
    delete(PieRP1);
    delete(PieLP1);

PiernaRP1=plot3([Marcha.RFoot.CoordenadasP1(ia,1),Marcha.RShin.CoordenadasP1(ia,1)]
,...
```

```

[Marcha.RFoot.CoordenadasP1(ia,3),Marcha.RShin.CoordenadasP1(ia,3)],...

[[Marcha.RFoot.CoordenadasP1(ia,2),Marcha.RShin.CoordenadasP1(ia,2)]*-1],...
    'color','y','linewidth',5);

PiernaLP1=plot3([Marcha.LFoot.CoordenadasP1(ia,1),Marcha.LShin.CoordenadasP1(ia,1)],.
..

[Marcha.LFoot.CoordenadasP1(ia,3),Marcha.LShin.CoordenadasP1(ia,3)],...

[[Marcha.LFoot.CoordenadasP1(ia,2),Marcha.LShin.CoordenadasP1(ia,2)]*-1],...
    'color','b','linewidth',5);
MusloRP1=plot3(
[Marcha.RShin.CoordenadasP1(ia,1),Marcha.RThigh.CoordenadasP1(ia,1)],...

[Marcha.RShin.CoordenadasP1(ia,3),Marcha.RThigh.CoordenadasP1(ia,3)],...

[[Marcha.RShin.CoordenadasP1(ia,2),Marcha.RThigh.CoordenadasP1(ia,2)]*-1],...
    'color','y','linewidth',6);
MusloLP1=plot3(
[Marcha.LShin.CoordenadasP1(ia,1),Marcha.LThigh.CoordenadasP1(ia,1)],...

[Marcha.LShin.CoordenadasP1(ia,3),Marcha.LThigh.CoordenadasP1(ia,3)],...

[[Marcha.LShin.CoordenadasP1(ia,2),Marcha.LThigh.CoordenadasP1(ia,2)]*-1],...
    'color','b','linewidth',6);

PieRP1=plot3([Marcha.RToe.CoordenadasP1(ia,1),Marcha.RFoot.CoordenadasP1(ia,1)],...

[Marcha.RToe.CoordenadasP1(ia,3),Marcha.RFoot.CoordenadasP1(ia,3)],...

[[Marcha.RToe.CoordenadasP1(ia,2),Marcha.RFoot.CoordenadasP1(ia,2)]*-1],...
    'color','k','linewidth',5);

PieLP1=plot3([Marcha.LToe.CoordenadasP1(ia,1),Marcha.LFoot.CoordenadasP1(ia,1)],...

[Marcha.LToe.CoordenadasP1(ia,3),Marcha.LFoot.CoordenadasP1(ia,3)],...

[[Marcha.LToe.CoordenadasP1(ia,2),Marcha.LFoot.CoordenadasP1(ia,2)]*-1],...
    'color','k','linewidth',5);

    end
    ib=ib+1;
end
%% Grafica Marcha del paciente semana 0

```



```

figure('Name','Marcha Anormal','NumberTitle','off');
fh = figure(2);
fh.WindowState = 'maximized';
t = 1:n2;
subplot(2,2,1)
curve1 = animatedline('color','k','LineWidth', 2);
hold on
plot(t,Marcha.Angulo.RTRSRF2,'r:o')%Se grafica la marcha del paciente 2 con una grafica
de puntos de color rojo
plot(t,Marcha.Angulo.RTRFRS2,'c:o')%Se grafica la marcha del paciente 2 con una grafica
de puntos de color celeste
axis([0 45 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha del Paciente Semana 0')
legend({'Data','Rodilla Derecha','Pie Derecho'},'NumColumns',3)
legend('boxoff')
subplot(2,2,2)
curve2 = animatedline('color','k','LineWidth', 2);
hold on
plot(t,Marcha.Angulo.LTLSLF2,'r:o')%Se grafica la marcha del paciente 2 con una grafica de
puntos de color rojo
plot(t,Marcha.Angulo.LTLFLS2,'c:o')%Se grafica la marcha del paciente 2 con una grafica de
puntos de color celeste
axis([0 45 80 200])
grid on
grid minor
hold on
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha del Paciente Semana 0')
legend({'Data','Rodilla Izquierda','Pie Izquierdo'},'NumColumns',3)
legend('boxoff')
PiernaRP2=plot3(0,0,0);
PantorrillaRP2=plot3(0,0,0);
PiernaLP2=plot3(0,0,0);
PantorrillaLP2=plot3(0,0,0);
PieRP2=plot3(0,0,0);
PieLP2=plot3(0,0,0);

ib=0;
while ib<2
    for ia=1:1:45

```

```

pause(1/30)
clearpoints(curve1)
x=[ia ia];
y=[0 ,270];
addpoints(curve1, x, y);

clearpoints (curve2)
addpoints(curve2, x, y);

subplot(2,2,[3,4])
axis equal
axis off
hold on
view(90,180)
title('Marcha del Paciente Semana 0')

delete(PiernaRP2);
delete(PiernaLP2);
delete(PantorillaRP2);
delete(PantorillaLP2);
delete(PieRP2);
delete(PieLP2);

```

```

PantorillaRP2=plot3([Marcha.RFoot.CoordenadasP2(ia,1),Marcha.RShin.CoordenadasP2(ia,1)],...

```

```

[Marcha.RFoot.CoordenadasP2(ia,3),Marcha.RShin.CoordenadasP2(ia,3)],...

```

```

[[Marcha.RFoot.CoordenadasP2(ia,2),Marcha.RShin.CoordenadasP2(ia,2)]*-1],...
'color','y','linewidth',5);

```

```

PantorillaLP2=plot3([Marcha.LFoot.CoordenadasP2(ia,1),Marcha.LShin.CoordenadasP2(ia,1)],...

```

```

[Marcha.LFoot.CoordenadasP2(ia,3),Marcha.LShin.CoordenadasP2(ia,3)],...

```

```

[[Marcha.LFoot.CoordenadasP2(ia,2),Marcha.LShin.CoordenadasP2(ia,2)]*-1],...
'color','b','linewidth',5);

```

```

PiernaRP2=plot3(
[Marcha.RShin.CoordenadasP2(ia,1),Marcha.RThigh.CoordenadasP2(ia,1)],...

```

```

[Marcha.RShin.CoordenadasP2(ia,3),Marcha.RThigh.CoordenadasP2(ia,3)],...

```

```

[[Marcha.RShin.CoordenadasP2(ia,2),Marcha.RThigh.CoordenadasP2(ia,2)]*-1],...
'color','y','linewidth',6);

```

```

    PiernaLP2=plot3(
[Marcha.LShin.CoordenadasP2(ia,1),Marcha.LThigh.CoordenadasP2(ia,1)],...

[Marcha.LShin.CoordenadasP2(ia,3),Marcha.LThigh.CoordenadasP2(ia,3)],...

[[Marcha.LShin.CoordenadasP2(ia,2),Marcha.LThigh.CoordenadasP2(ia,2)]*-1],...
    'color','b','linewidth',6);

PieRP2=plot3([Marcha.RToe.CoordenadasP2(ia,1),Marcha.RFoot.CoordenadasP2(ia,1)],...

[Marcha.RToe.CoordenadasP2(ia,3),Marcha.RFoot.CoordenadasP2(ia,3)],...

[[Marcha.RToe.CoordenadasP2(ia,2),Marcha.RFoot.CoordenadasP2(ia,2)]*-1],...
    'color','k','linewidth',5);

PieLP2=plot3([Marcha.LToe.CoordenadasP2(ia,1),Marcha.LFoot.CoordenadasP2(ia,1)],...

[Marcha.LToe.CoordenadasP2(ia,3),Marcha.LFoot.CoordenadasP2(ia,3)],...

[[Marcha.LToe.CoordenadasP2(ia,2),Marcha.LFoot.CoordenadasP2(ia,2)]*-1],...
    'color','k','linewidth',5);

    end
    ib=ib+1;
end
%% Comparacion poniendo las 4 graficas sin animaciones
figure('Name','Marcha Normal VS Marcha Anormal','NumberTitle','off');
fh = figure(3);
fh.WindowState = 'maximized';
t1 = 1:n;
t2 = 1:n2;
subplot(2,2,1)
hold on
plot(t1,Marcha.Angulo.RTRSRF1,'b--.')
hold on
plot(t1,Marcha.Angulo.RTRFRS1,'g--.')
axis([0 80 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha Control')
legend({'Rodilla Derecha','Pie Derecho'},'NumColumns',3)
legend('boxoff')
subplot(2,2,3)
plot(t2,Marcha.Angulo.RTRSRF2,'r:o')

```

```

hold on
plot(t2,Marcha.Angulo.RTRFRS2,'c:o')
axis([0 45 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha del Paciente Semana 0')
legend({'Rodilla Derecha','Pie Derecho'},'NumColumns',3)
legend('boxoff')
subplot(2,2,2)
plot(t1,Marcha.Angulo.LTSLF1,'b--.')
hold on
plot(t1,Marcha.Angulo.LTLFLS1,'g--.')
axis([0 80 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha Control')
legend({'Rodilla Izquierda','Pie Izquierdo'},'NumColumns',3)
legend('boxoff')
subplot(2,2,4)
plot(t2,Marcha.Angulo.LTSLF2,'r:o')
hold on
plot(t2,Marcha.Angulo.LTLFLS2,'c:o')
axis([0 45 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha del Paciente Semana 0')
legend({'Rodilla Izquierda','Pie Izquierdo'},'NumColumns',3)
legend('boxoff')
%Calculo de angulos entre segmentos con matriz importada del software IPI
%SOFT
clear all
clc
close all
load('caminandoground_export.mat')%se carga la matriz
n = length(JointsData.RThigh.Coordinate);%Cantidad de frames que se grabaron

%Este FOR es para guardar cada una de las coordenadas x,y,z del RShin en
%otro Struct
for i = 1:n

```

```

    JointsData.RShin.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RShin.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RShin.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RShin.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LShin en
%otro Struct
for i = 1:n

```

    JointsData.LShin.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LShin.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LShin.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LShin.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RThigh en
%otro Struct
for i = 1:n

```

    JointsData.RThigh.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X

```

```

    Marcha.RThigh.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RThigh.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RThigh.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LThigh en

%otro Struct

for i = 1:n

```

    JointsData.LThigh.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LThigh.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LThigh.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LThigh.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RFoot en

%otro Struct

for i = 1:n

```

    JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RFoot.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames

```

```

    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RFoot.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RFoot.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LFoot en
%otro Struct

```

for i = 1:n
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LFoot.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LFoot.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LFoot.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RToe en
%otro Struct

```

for i = 1:n
    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RToe.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RToe.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2

```

```

    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RToe.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LToe en
%otro Struct

```

for i = 1:n
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LToe.CoordenadasP1(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LToe.CoordenadasP1(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LToe.CoordenadasP1(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Para obtener el angulo entre dos segmentos primero tuvimos que contruir
%los segmentos utilizando las coordenadas, haciendo la operacion
%RShin-Rthigh,RShin-RFoot,LShin-Lthigh,LShin-LFoot y tambien obtener la magnitud de
estos
%segmentos

%Este FOR es para obtener el segmento Rshin-Rthigh

```

for i = (1:n)
    A=Marcha.RShin.CoordenadasP1(i,1)-Marcha.RThigh.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
    Marcha.RSRT.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.RShin.CoordenadasP1(i,2)-Marcha.RThigh.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
    Marcha.RSRT.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.RShin.CoordenadasP1(i,3)-Marcha.RThigh.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
    Marcha.RSRT.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct

```



```
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz cuadrada de la suma de los cuadrados
```

```
Marcha.RSRT.MagnitudP1(i,1)=D;%Se guardo en otro Struct  
i=i+1;  
end
```

```
%Este FOR es para obtener el segmento Lshin-Lthigh
```

```
for i = (1:n)
```

```
A=Marcha.LShin.CoordenadasP1(i,1)-Marcha.LThigh.CoordenadasP1(i,1);%Se hizo la resta en coordenada X
```

```
Marcha.LSLT.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
```

```
B=Marcha.LShin.CoordenadasP1(i,2)-Marcha.LThigh.CoordenadasP1(i,2);%Se hizo la resta en coordenada Y
```

```
Marcha.LSLT.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
```

```
C=Marcha.LShin.CoordenadasP1(i,3)-Marcha.LThigh.CoordenadasP1(i,3);%Se hizo la resta en coordenada Z
```

```
Marcha.LSLT.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
```

```
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz cuadrada de la suma de los cuadrados
```

```
Marcha.LSLT.MagnitudP1(i,1)=D;%Se guardo en otro Struct  
i=i+1;  
end
```

```
%Este FOR es para obtener el segmento Rshin-RFoot
```

```
for i = (1:n)
```

```
A=Marcha.RShin.CoordenadasP1(i,1)-Marcha.RFoot.CoordenadasP1(i,1);%Se hizo la resta en coordenada X
```

```
Marcha.RSRF.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
```

```
B=Marcha.RShin.CoordenadasP1(i,2)-Marcha.RFoot.CoordenadasP1(i,2);%Se hizo la resta en coordenada Y
```

```
Marcha.RSRF.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
```

```
C=Marcha.RShin.CoordenadasP1(i,3)-Marcha.RFoot.CoordenadasP1(i,3);%Se hizo la resta en coordenada Z
```

```
Marcha.RSRF.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
```

```
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz cuadrada de la suma de los cuadrados
```

```
Marcha.RSRF.MagnitudP1(i,1)=D;%Se guardo en otro Struct  
i=i+1;  
end
```

```
%Este FOR es para obtener el segmento LShin-LFoot
```

```
for i = (1:n)
```

```
A=Marcha.LShin.CoordenadasP1(i,1)-Marcha.LFoot.CoordenadasP1(i,1);%Se hizo la resta en coordenada X
```

```
Marcha.LSLF.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
```

```

    B=Marcha.LShin.CoordenadasP1(i,2)-Marcha.LFoot.CoordenadasP1(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.LSLF.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.LShin.CoordenadasP1(i,3)-Marcha.LFoot.CoordenadasP1(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.LSLF.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados
    Marcha.LSLF.MagnitudP1(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento RFoot-RToe
for i = (1:n)
    A=Marcha.RFoot.CoordenadasP1(i,1)-Marcha.RToe.CoordenadasP1(i,1);%Se hizo la
    resta en coordenada X
    Marcha.RFRT.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.RFoot.CoordenadasP1(i,2)-Marcha.RToe.CoordenadasP1(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.RFRT.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.RFoot.CoordenadasP1(i,3)-Marcha.RToe.CoordenadasP1(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.RFRT.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados
    Marcha.RFRT.MagnitudP1(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento LFoot-LToe
for i = (1:n)
    A=Marcha.LFoot.CoordenadasP1(i,1)-Marcha.LToe.CoordenadasP1(i,1);%Se hizo la
    resta en coordenada X
    Marcha.LFLT.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.LFoot.CoordenadasP1(i,2)-Marcha.LToe.CoordenadasP1(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.LFLT.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.LFoot.CoordenadasP1(i,3)-Marcha.LToe.CoordenadasP1(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.LFLT.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados
    Marcha.LFLT.MagnitudP1(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento RFoot-RShin
for i = (1:n)
    A=Marcha.RFoot.CoordenadasP1(i,1)-Marcha.RShin.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
    Marcha.RFRS.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.RFoot.CoordenadasP1(i,2)-Marcha.RShin.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
    Marcha.RFRS.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.RFoot.CoordenadasP1(i,3)-Marcha.RShin.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
    Marcha.RFRS.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
    Marcha.RFRS.MagnitudP1(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento LFoot-LShin
for i = (1:n)
    A=Marcha.LFoot.CoordenadasP1(i,1)-Marcha.LShin.CoordenadasP1(i,1);%Se hizo la
resta en coordenada X
    Marcha.LFLS.RestaP1(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.LFoot.CoordenadasP1(i,2)-Marcha.LShin.CoordenadasP1(i,2);%Se hizo la
resta en coordenada Y
    Marcha.LFLS.RestaP1(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.LFoot.CoordenadasP1(i,3)-Marcha.LShin.CoordenadasP1(i,3);%Se hizo la
resta en coordenada Z
    Marcha.LFLS.RestaP1(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
    Marcha.LFLS.MagnitudP1(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

%Ya teniendo los la magnitud de los segmentos ya hechos, seguimos con la
%formula multiplicando las magnitudes de los dos segmentos

```

%Este FOR es para obtener el producto de las magnitudes RSRF*RSRT
for i = (1:n)
    A=Marcha.RSRF.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RSRT.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=A*B;%Multiplicacion de las dos magnitudes
    Marcha.R1PMP1(i,1)=C;%El resultado se guardo en otro Struct
    i=i+1;

```

end

%Este FOR es para obtener el producto de las magnitudes LSLF*LSLT

for i = (1:n)

A=Marcha.LSLF.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la operacion mas simple

B=Marcha.LSLT.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la operacion mas simple

C=A*B;%Multiplicacion de las dos magnitudes

Marcha.L1PMP1(i,1)=C;%El resultado se guardo en otro Struct

i=i+1;

end

%Este FOR es para obtener el producto de las magnitudes RFRT*RFRS

for i = (1:n)

A=Marcha.RFRS.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la operacion mas simple

B=Marcha.RFRT.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la operacion mas simple

C=A*B;%Multiplicacion de las dos magnitudes

Marcha.R2PMP1(i,1)=C;%El resultado se guardo en otro Struct

i=i+1;

end

%Este FOR es para obtener el producto de las magnitudes LFLT*LFLS

for i = (1:n)

A=Marcha.LFLS.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la operacion mas simple

B=Marcha.LFLT.MagnitudP1(i,1);%Se guardo el estruct como una variable para hacer la operacion mas simple

C=A*B;%Multiplicacion de las dos magnitudes

Marcha.L2PMP1(i,1)=C;%El resultado se guardo en otro Struct

i=i+1;

end

%Ya teniendo la multiplicacion de las magnitudes de los segmentos, seguimos con la %formula realizando el producto punto entre los dos segmentos

%Este FOR es para obtener el pructo punto entre los dos segmentos

%RSRT*.RSRF

for i = (1:n)

A=Marcha.RSRT.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la operacion mas simple

B=Marcha.RSRF.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la operacion mas simple

C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot

```

    Marcha.R1PPP1(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener el pructo punto entre los dos segmentos

```
%LSLT*.LSLF
```

```
for i = (1:n)
```

```
    A=Marcha.LSLT.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    B=Marcha.LSLF.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
```

```
    Marcha.L1PPP1(i,1)=C;%Se guarda el resultado en otro Struct
```

```
    i=i+1;
```

```
end
```

%Este FOR es para obtener el pructo punto entre los dos segmentos

```
%RFRT*.RFRS
```

```
for i = (1:n)
```

```
    A=Marcha.RFRT.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    B=Marcha.RFRS.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
```

```
    Marcha.R2PPP1(i,1)=C;%Se guarda el resultado en otro Struct
```

```
    i=i+1;
```

```
end
```

%Este FOR es para obtener el pructo punto entre los dos segmentos

```
%LFLT*.LFLS
```

```
for i = (1:n)
```

```
    A=Marcha.LFLT.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    B=Marcha.LFLS.RestaP1(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
```

```
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
```

```
    Marcha.L2PPP1(i,1)=C;%Se guarda el resultado en otro Struct
```

```
    i=i+1;
```

```
end
```

%Una Vez ya teniendo el resultado de la multiplziacion de las magnitudes y

%el producto punto de los segmentos seguimos con la formula que seria el

%coseno inverzo de el producto punto de los segmentos entre el producto de

%las magnitudes

%Este FOR es para obtener a en angulo en grados de la Pierna Derecha

```

for i = (1:n)
    A=Marcha.R1PPP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    B=Marcha.R1PMP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    C=acosd(A/B);%Se caclula en angulo
    Marcha.Angulo.RTRSRF1(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener a en angulo en grados de la Pierna Izquierda

```

for i = (1:n)
    A=Marcha.L1PPP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    B=Marcha.L1PMP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    C=acosd(A/B);%Se caclula en angulo
    Marcha.Angulo.LTSLF1(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener a en angulo en grados del pie derecho

```

for i = (1:n)
    A=Marcha.R2PPP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    B=Marcha.R2PMP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    C=acosd(A/B);%Se caclula en angulo
    Marcha.Angulo.RTRFRS1(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener a en angulo en grados del pie izquierdo

```

for i = (1:n)
    A=Marcha.L2PPP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    B=Marcha.L2PMP1(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
    C=acosd(A/B);%Se caclula en angulo
    Marcha.Angulo.LTLFLS1(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

```

%%
% load('caminandomalground_export.mat')
% load('Carlos1_export.mat')

```

```

load('Carlos1_export1CICLO.mat')
n2 = length(JointsData.RThigh.Coordinate);%Cantidad de frames que se grabaron

%Este FOR es para guardar cada una de las coordenadas x,y,z del RShin en
%otro Struct
for i = 1:n2
    JointsData.RShin.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RShin.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RShin.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RShin.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LShin en
%otro Struct
for i = 1:n2
    JointsData.LShin.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LShin.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LShin.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LShin.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LShin.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RThigh en

```

```

%otro Struct
for i = 1:n2
    JointsData.RThigh.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RThigh.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.RThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RThigh.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RThigh.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LThigh en

```

%otro Struct
for i = 1:n2
    JointsData.LThigh.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LThigh.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LThigh.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LThigh.Coordinate(i,1);%Volvemos a cargar las coordenadas de cada uno de
los frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LThigh.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RFoot en

```

%otro Struct
for i = 1:n2
    JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames

```



```

A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
Marcha.RFoot.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
Marcha.RFoot.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
JointsData.RFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
Marcha.RFoot.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LFoot en
%otro Struct
for i = 1:n2
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LFoot.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LFoot.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LFoot.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LFoot.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del RToe en
%otro Struct
for i = 1:n2
    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.RToe.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1

```

```

    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.RToe.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.RToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.RToe.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Este FOR es para guardar cada una de las coordenadas x,y,z del LToe en
%otro Struct
for i = 1:n2

```

```

    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    A=ans.x(1,1);%De las coordenadas obtenidas solo toma la coordenada en X
    Marcha.LToe.CoordenadasP2(i,1)=A;%Guardamos la coordenada X en otro struct en la
columna 1
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    B=ans.y(1,1);%De las coordenadas obtenidas solo toma la coordenada en Y
    Marcha.LToe.CoordenadasP2(i,2)=B;%Guardamos la coordenada Y en otro struct en la
columna 2
    JointsData.LToe.Coordinate(i,1);%Adquiere las coordenadas x,y,z de cada uno de los
frames
    C=ans.z(1,1);%De las coordenadas obtenidas solo toma la coordenada en Z
    Marcha.LToe.CoordenadasP2(i,3)=C;%Guardamos la coordenada Z en otro struct en la
columna 3
    i=i+1;
end

```

```

%Para obtener el angulo entre dos segmentos primero tuvimos que contruir
%los segmentos utilizando las coordenadas, haciendo la operacion
%RShin-Rthigh,RShin-RFoot,LShin-Lthigh,LShin-LFoot y tambien obtener la magnitud de
estos
%segmentos

```

```

%Este FOR es para obtener el segmento Rshin-Rthigh
for i = (1:n2)

```

```

    A=Marcha.RShin.CoordenadasP2(i,1)-Marcha.RThigh.CoordenadasP2(i,1);%Se hizo la
resta en coordenada X
    Marcha.RSRT.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct

```

```

    B=Marcha.RShin.CoordenadasP2(i,2)-Marcha.RThigh.CoordenadasP2(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.RSRT.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.RShin.CoordenadasP2(i,3)-Marcha.RThigh.CoordenadasP2(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.RSRT.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados
    Marcha.RSRT.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento Lshin-Lthigh
for i = (1:n2)

```

```

    A=Marcha.LShin.CoordenadasP2(i,1)-Marcha.LThigh.CoordenadasP2(i,1);%Se hizo la
    resta en coordenada X
    Marcha.LSLT.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.LShin.CoordenadasP2(i,2)-Marcha.LThigh.CoordenadasP2(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.LSLT.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.LShin.CoordenadasP2(i,3)-Marcha.LThigh.CoordenadasP2(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.LSLT.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados
    Marcha.LSLT.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento Rshin-RFoot
for i = (1:n2)

```

```

    A=Marcha.RShin.CoordenadasP2(i,1)-Marcha.RFoot.CoordenadasP2(i,1);%Se hizo la
    resta en coordenada X
    Marcha.RSRF.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.RShin.CoordenadasP2(i,2)-Marcha.RFoot.CoordenadasP2(i,2);%Se hizo la
    resta en coordenada Y
    Marcha.RSRF.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.RShin.CoordenadasP2(i,3)-Marcha.RFoot.CoordenadasP2(i,3);%Se hizo la
    resta en coordenada Z
    Marcha.RSRF.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
    cuadrada de la suma de los cuadrados
    Marcha.RSRF.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento Lshin-LFoot
for i = (1:n2)
    A=Marcha.LShin.CoordenadasP2(i,1)-Marcha.LFoot.CoordenadasP2(i,1);%Se hizo la
resta en coordenada X
    Marcha.LSLF.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.LShin.CoordenadasP2(i,2)-Marcha.LFoot.CoordenadasP2(i,2);%Se hizo la
resta en coordenada Y
    Marcha.LSLF.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.LShin.CoordenadasP2(i,3)-Marcha.LFoot.CoordenadasP2(i,3);%Se hizo la
resta en coordenada Z
    Marcha.LSLF.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
    Marcha.LSLF.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento RFoot-RToe
for i = (1:n2)
    A=Marcha.RFoot.CoordenadasP2(i,1)-Marcha.RToe.CoordenadasP2(i,1);%Se hizo la
resta en coordenada X
    Marcha.RFRT.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.RFoot.CoordenadasP2(i,2)-Marcha.RToe.CoordenadasP2(i,2);%Se hizo la
resta en coordenada Y
    Marcha.RFRT.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.RFoot.CoordenadasP2(i,3)-Marcha.RToe.CoordenadasP2(i,3);%Se hizo la
resta en coordenada Z
    Marcha.RFRT.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
    D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz
cuadrada de la suma de los cuadrados
    Marcha.RFRT.MagnitudP2(i,1)=D;%Se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el segmento LFoot-LToe
for i = (1:n2)
    A=Marcha.LFoot.CoordenadasP2(i,1)-Marcha.LToe.CoordenadasP2(i,1);%Se hizo la
resta en coordenada X
    Marcha.LFLT.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
    B=Marcha.LFoot.CoordenadasP2(i,2)-Marcha.LToe.CoordenadasP2(i,2);%Se hizo la
resta en coordenada Y
    Marcha.LFLT.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
    C=Marcha.LFoot.CoordenadasP2(i,3)-Marcha.LToe.CoordenadasP2(i,3);%Se hizo la
resta en coordenada Z
    Marcha.LFLT.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct

```

```
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz cuadrada de la suma de los cuadrados
```

```
Marcha.LFLT.MagnitudP2(i,1)=D;%Se guardo en otro Struct  
i=i+1;  
end
```

```
%Este FOR es para obtener el segmento RFoot-RShin
```

```
for i = (1:n2)
```

```
A=Marcha.RFoot.CoordenadasP2(i,1)-Marcha.RShin.CoordenadasP2(i,1);%Se hizo la resta en coordenada X
```

```
Marcha.RFRS.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
```

```
B=Marcha.RFoot.CoordenadasP2(i,2)-Marcha.RShin.CoordenadasP2(i,2);%Se hizo la resta en coordenada Y
```

```
Marcha.RFRS.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
```

```
C=Marcha.RFoot.CoordenadasP2(i,3)-Marcha.RShin.CoordenadasP2(i,3);%Se hizo la resta en coordenada Z
```

```
Marcha.RFRS.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
```

```
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz cuadrada de la suma de los cuadrados
```

```
Marcha.RFRS.MagnitudP2(i,1)=D;%Se guardo en otro Struct
```

```
i=i+1;  
end
```

```
%Este FOR es para obtener el segmento LFoot-LShin
```

```
for i = (1:n2)
```

```
A=Marcha.LFoot.CoordenadasP2(i,1)-Marcha.LShin.CoordenadasP2(i,1);%Se hizo la resta en coordenada X
```

```
Marcha.LFLS.RestaP2(i,1)=A;%Se guardo el resultado en otro Struct
```

```
B=Marcha.LFoot.CoordenadasP2(i,2)-Marcha.LShin.CoordenadasP2(i,2);%Se hizo la resta en coordenada Y
```

```
Marcha.LFLS.RestaP2(i,2)=B;%Se guardo el resultado en otro Struct
```

```
C=Marcha.LFoot.CoordenadasP2(i,3)-Marcha.LShin.CoordenadasP2(i,3);%Se hizo la resta en coordenada Z
```

```
Marcha.LFLS.RestaP2(i,3)=C;%Se guardo el resultado en otro Struct
```

```
D=sqrt((A^2)+(B^2)+(C^2));%Se obtuvo la magnitud del segmento utilizando la raiz cuadrada de la suma de los cuadrados
```

```
Marcha.LFLS.MagnitudP2(i,1)=D;%Se guardo en otro Struct
```

```
i=i+1;  
end
```

```
%Ya teniendo los la magnitud de los segmentos ya hechos, seguimos con la %formula multiplicando las magnitudes de los dos segmentos
```

```
%Este FOR es para obtener el producto de las magnitudes RSRF*RSRT
```

```
for i = (1:n2)
```

```

    A=Marcha.RSRF.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RSRT.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=A*B;%Multiplicacion de las dos magnitudes
    Marcha.R1PMP2(i,1)=C;%El resultado se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el producto de las magnitudes LSLF*LSLT
for i = (1:n2)
    A=Marcha.LSLF.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.LSLT.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=A*B;%Multiplicacion de las dos magnitudes
    Marcha.L1PMP2(i,1)=C;%El resultado se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el producto de las magnitudes RFRT*RFRS
for i = (1:n2)
    A=Marcha.RFRS.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RFRT.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=A*B;%Multiplicacion de las dos magnitudes
    Marcha.R2PMP2(i,1)=C;%El resultado se guardo en otro Struct
    i=i+1;
end

```

```

%Este FOR es para obtener el producto de las magnitudes LFLT*LFLS
for i = (1:n2)
    A=Marcha.LFLS.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.LFLT.MagnitudP2(i,1);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=A*B;%Multiplicacion de las dos magnitudes
    Marcha.L2PMP2(i,1)=C;%El resultado se guardo en otro Struct
    i=i+1;
end

```

%Ya teniendo la multiplicacion de las magnitudes de los segmentos, seguimos con la
%formula realizando el producto punto entre los dos segmentos

%Este FOR es para obtener el producto punto entre los dos segmentos

```

%RSRT*.RSRF
for i = (1:n2)
    A=Marcha.RSRT.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RSRF.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
    Marcha.R1PPP2(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener el pructo punto entre los dos segmentos

```

%LSLT*.LSLF
for i = (1:n2)
    A=Marcha.LSLT.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.LSLF.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
    Marcha.L1PPP2(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener el pructo punto entre los dos segmentos

```

%RFRT*.RFRS
for i = (1:n2)
    A=Marcha.RFRT.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.RFRS.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
    Marcha.R2PPP2(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

%Este FOR es para obtener el pructo punto entre los dos segmentos

```

%LFLT*.LFLS
for i = (1:n2)
    A=Marcha.LFLT.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    B=Marcha.LFLS.RestaP2(i,:);%Se guardo el estruct como una variable para hacer la
operacion mas simple
    C=dot(A,B);%Se realiza la operacion utilizanco la funcion dot
    Marcha.L2PPP2(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

```

```
%Una Vez ya teniendo el resultado de la multiplzacion de las magnitudes y
%el producto punto de los segmentos seguimos con la formula que seria el
%coseno inverzo de el producto punto de los segmentos entre el producto de
%las magnitudes
```

```
%Este FOR es para obtener a en angulo en grados de la Pierna Derecha
```

```
for i = (1:n2)
```

```
    A=Marcha.R1PPP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
```

```
    B=Marcha.R1PMP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
```

```
    C=acosd(A/B);%Se caclula en angulo
```

```
    Marcha.Angulo.RTRSRF2(i,1)=C;%Se guarda el resultado en otro Struct
```

```
    i=i+1;
```

```
end
```

```
%Este FOR es para obtener a en angulo en grados de la Pierna Izquierda
```

```
for i = (1:n2)
```

```
    A=Marcha.L1PPP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
```

```
    B=Marcha.L1PMP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
```

```
    C=acosd(A/B);%Se caclula en angulo
```

```
    Marcha.Angulo.LTLSLF2(i,1)=C;%Se guarda el resultado en otro Struct
```

```
    i=i+1;
```

```
end
```

```
%Este FOR es para obtener a en angulo en grados del pie derecho
```

```
for i = (1:n2)
```

```
    A=Marcha.R2PPP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
```

```
    B=Marcha.R2PMP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
```

```
    C=acosd(A/B);%Se caclula en angulo
```

```
    Marcha.Angulo.RTRFRS2(i,1)=C;%Se guarda el resultado en otro Struct
```

```
    i=i+1;
```

```
end
```

```
%Este FOR es para obtener a en angulo en grados del pie izquierdo
```

```
for i = (1:n2)
```

```
    A=Marcha.L2PPP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
```

```
    B=Marcha.L2PMP2(i,1);%Se guardo el estruct como una variable para hacer la operacion
mas simple
```

```
    C=acosd(A/B);%Se caclula en angulo
```



```

    Marcha.Angulo.LTLFLS2(i,1)=C;%Se guarda el resultado en otro Struct
    i=i+1;
end

%% Grafica Marcha Control

figure('Name','Marcha Normal','NumberTitle','off');
fh = figure(1);
fh.WindowState = 'maximized';
t = 1:n;
subplot(2,2,1)
curve1 = animatedline('color','k','LineWidth', 2);
hold on
plot(t,Marcha.Angulo.RTRSRF1,'b--.')
```

%Se grafica la marcha del paciente 1 con una grafica de puntos de color azul

```

plot(t,Marcha.Angulo.RTRFRS1,'g--.')
```

%Se grafica la marcha del paciente 1 con una grafica de puntos de color verde

```

axis([0 80 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha Control')
legend({'Data','Rodilla Derecha','Pie Derecho'},'NumColumns',3)
legend('boxoff')
subplot(2,2,2)
curve2 = animatedline('color','k','LineWidth', 2);
hold on
plot(t,Marcha.Angulo.LTLSLF1,'b--.')
```

%Se grafica la marcha del paciente 1 con una grafica de puntos de color azul

```

plot(t,Marcha.Angulo.LTLFLS1,'g--.')
```

%Se grafica la marcha del paciente 1 con una grafica de puntos de color verde

```

axis([0 80 80 200])
grid on
grid minor
hold on
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha Control')
legend({'Data','Rodilla Izquierda','Pie Izquierdo'},'NumColumns',3)
legend('boxoff')
MusloRP1=plot3(0,0,0);
PiernaRP1=plot3(0,0,0);
MusloLP1=plot3(0,0,0);
PiernaLP1=plot3(0,0,0);
PieRP1=plot3(0,0,0);
```

```

PieLP1=plot3(0,0,0);

ib=0;
while ib<2
for ia=1:1:80
    pause(1/30)
    clearpoints(curve1)
    x=[ia ia];
    y=[0 ,270];
    addpoints(curve1, x, y);

    clearpoints (curve2)
    addpoints(curve2, x, y);

    subplot(2,2,[3,4])
    axis equal
    axis off
    hold on
    view(90,180)
    title('Marcha Control')

    delete(MusloRP1);
    delete(MusloLP1);
    delete(PiernaRP1);
    delete(PiernaLP1);
    delete(PieRP1);
    delete(PieLP1);

PiernaRP1=plot3([Marcha.RFoot.CoordenadasP1(ia,1),Marcha.RShin.CoordenadasP1(ia,1)]
,...
[Marcha.RFoot.CoordenadasP1(ia,3),Marcha.RShin.CoordenadasP1(ia,3)],...
[[Marcha.RFoot.CoordenadasP1(ia,2),Marcha.RShin.CoordenadasP1(ia,2)]*-1],...
    'color','y','linewidth',5);

PiernaLP1=plot3([Marcha.LFoot.CoordenadasP1(ia,1),Marcha.LShin.CoordenadasP1(ia,1)],.
..
[Marcha.LFoot.CoordenadasP1(ia,3),Marcha.LShin.CoordenadasP1(ia,3)],...
[[Marcha.LFoot.CoordenadasP1(ia,2),Marcha.LShin.CoordenadasP1(ia,2)]*-1],...
    'color','b','linewidth',5);
    MusloRP1=plot3(
[Marcha.RShin.CoordenadasP1(ia,1),Marcha.RThigh.CoordenadasP1(ia,1)],...

```

```

[Marcha.RShin.CoordenadasP1(ia,3),Marcha.RThigh.CoordenadasP1(ia,3)],...

[[Marcha.RShin.CoordenadasP1(ia,2),Marcha.RThigh.CoordenadasP1(ia,2)]*-1],...
    'color','y','linewidth',6);
    MusloLP1=plot3(
[Marcha.LShin.CoordenadasP1(ia,1),Marcha.LThigh.CoordenadasP1(ia,1)],...

[Marcha.LShin.CoordenadasP1(ia,3),Marcha.LThigh.CoordenadasP1(ia,3)],...

[[Marcha.LShin.CoordenadasP1(ia,2),Marcha.LThigh.CoordenadasP1(ia,2)]*-1],...
    'color','b','linewidth',6);

PieRP1=plot3([Marcha.RToe.CoordenadasP1(ia,1),Marcha.RFoot.CoordenadasP1(ia,1)],...

[Marcha.RToe.CoordenadasP1(ia,3),Marcha.RFoot.CoordenadasP1(ia,3)],...

[[Marcha.RToe.CoordenadasP1(ia,2),Marcha.RFoot.CoordenadasP1(ia,2)]*-1],...
    'color','k','linewidth',5);

PieLP1=plot3([Marcha.LToe.CoordenadasP1(ia,1),Marcha.LFoot.CoordenadasP1(ia,1)],...

[Marcha.LToe.CoordenadasP1(ia,3),Marcha.LFoot.CoordenadasP1(ia,3)],...

[[Marcha.LToe.CoordenadasP1(ia,2),Marcha.LFoot.CoordenadasP1(ia,2)]*-1],...
    'color','k','linewidth',5);

    end
    ib=ib+1;
end
%% Grafica Marcha del paciente semana 0

figure('Name','Marcha Anormal','NumberTitle','off');
fh = figure(2);
fh.WindowState = 'maximized';
t = 1:n2;
subplot(2,2,1)
curve1 = animatedline('color','k','LineWidth', 2);
hold on
plot(t,Marcha.Angulo.RTRSRF2,'r:o')%Se grafica la marcha del paciente 2 con una grafica
de puntos de color rojo
plot(t,Marcha.Angulo.RTRFRS2,'c:o')%Se grafica la marcha del paciente 2 con una grafica
de puntos de color celeste
axis([0 45 80 200])
grid on
grid minor

```

```

xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha del Paciente Semana 0')
legend({'Data','Rodilla Derecha','Pie Derecho'},'NumColumns',3)
legend('boxoff')
subplot(2,2,2)
curve2 = animatedline('color','k','LineWidth', 2);
hold on
plot(t,Marcha.Angulo.LTSLF2,'r:o')%Se grafica la marcha del paciente 2 con una grafica de
puntos de color rojo
plot(t,Marcha.Angulo.LTLFLS2,'c:o')%Se grafica la marcha del paciente 2 con una grafica de
puntos de color celeste
axis([0 45 80 200])
grid on
grid minor
hold on
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha del Paciente Semana 0')
legend({'Data','Rodilla Izquierda','Pie Izquierdo'},'NumColumns',3)
legend('boxoff')
PiernaRP2=plot3(0,0,0);
PantorrillaRP2=plot3(0,0,0);
PiernaLP2=plot3(0,0,0);
PantorrillaLP2=plot3(0,0,0);
PieRP2=plot3(0,0,0);
PieLP2=plot3(0,0,0);

ib=0;
while ib<2
    for ia=1:1:45
        pause(1/30)
        clearpoints(curve1)
        x=[ia ia];
        y=[0 ,270];
        addpoints(curve1, x, y);

        clearpoints (curve2)
        addpoints(curve2, x, y);

        subplot(2,2,[3,4])
        axis equal
        axis off
        hold on
        view(90,180)
        title('Marcha del Paciente Semana 0')
    end
    ib=ib+1;
end

```

```
delete(PiernaRP2);
delete(PiernaLP2);
delete(PantorillaRP2);
delete(PantorillaLP2);
delete(PieRP2);
delete(PieLP2);
```

```
PantorillaRP2=plot3([Marcha.RFoot.CoordenadasP2(ia,1),Marcha.RShin.CoordenadasP2(ia,1)],...
```

```
[Marcha.RFoot.CoordenadasP2(ia,3),Marcha.RShin.CoordenadasP2(ia,3)],...
```

```
[[Marcha.RFoot.CoordenadasP2(ia,2),Marcha.RShin.CoordenadasP2(ia,2)]*-1],...
'color','y','linewidth',5);
```

```
PantorillaLP2=plot3([Marcha.LFoot.CoordenadasP2(ia,1),Marcha.LShin.CoordenadasP2(ia,1)],...
```

```
[Marcha.LFoot.CoordenadasP2(ia,3),Marcha.LShin.CoordenadasP2(ia,3)],...
```

```
[[Marcha.LFoot.CoordenadasP2(ia,2),Marcha.LShin.CoordenadasP2(ia,2)]*-1],...
'color','b','linewidth',5);
```

```
PiernaRP2=plot3(
[Marcha.RShin.CoordenadasP2(ia,1),Marcha.RThigh.CoordenadasP2(ia,1)],...
```

```
[Marcha.RShin.CoordenadasP2(ia,3),Marcha.RThigh.CoordenadasP2(ia,3)],...
```

```
[[Marcha.RShin.CoordenadasP2(ia,2),Marcha.RThigh.CoordenadasP2(ia,2)]*-1],...
'color','y','linewidth',6);
```

```
PiernaLP2=plot3(
[Marcha.LShin.CoordenadasP2(ia,1),Marcha.LThigh.CoordenadasP2(ia,1)],...
```

```
[Marcha.LShin.CoordenadasP2(ia,3),Marcha.LThigh.CoordenadasP2(ia,3)],...
```

```
[[Marcha.LShin.CoordenadasP2(ia,2),Marcha.LThigh.CoordenadasP2(ia,2)]*-1],...
'color','b','linewidth',6);
```

```
PieRP2=plot3([Marcha.RToe.CoordenadasP2(ia,1),Marcha.RFoot.CoordenadasP2(ia,1)],...
```

```
[Marcha.RToe.CoordenadasP2(ia,3),Marcha.RFoot.CoordenadasP2(ia,3)],...
```

```
[[Marcha.RToe.CoordenadasP2(ia,2),Marcha.RFoot.CoordenadasP2(ia,2)]*-1],...
'color','k','linewidth',5);
```

```

PieLP2=plot3([Marcha.LToe.CoordenadasP2(ia,1),Marcha.LFoot.CoordenadasP2(ia,1)],...

[Marcha.LToe.CoordenadasP2(ia,3),Marcha.LFoot.CoordenadasP2(ia,3)],...

[[Marcha.LToe.CoordenadasP2(ia,2),Marcha.LFoot.CoordenadasP2(ia,2)]*-1],...
    'color','k','linewidth',5);

    end
    ib=ib+1;
end
%% Comparacion poniendo las 4 graficas sin animaciones
figure('Name','Marcha Normal VS Marcha Anormal','NumberTitle','off');
fh = figure(3);
fh.WindowState = 'maximized';
t1 = 1:n;
t2 = 1:n2;
subplot(2,2,1)
hold on
plot(t1,Marcha.Angulo.RTRSRF1,'b--.')
hold on
plot(t1,Marcha.Angulo.RTRFRS1,'g--.')
axis([0 80 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha Control')
legend({'Rodilla Derecha','Pie Derecho'},'NumColumns',3)
legend('boxoff')
subplot(2,2,3)
plot(t2,Marcha.Angulo.RTRSRF2,'r:o')
hold on
plot(t2,Marcha.Angulo.RTRFRS2,'c:o')
axis([0 45 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha del Paciente Semana 0')
legend({'Rodilla Derecha','Pie Derecho'},'NumColumns',3)
legend('boxoff')
subplot(2,2,2)
plot(t1,Marcha.Angulo.LTSLF1,'b--.')
hold on
plot(t1,Marcha.Angulo.LTLFLS1,'g--.')

```

```

axis([0 80 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha Control')
legend({'Rodilla Izquierda','Pie Izquierdo'},'NumColumns',3)
legend('boxoff')
subplot(2,2,4)
plot(t2,Marcha.Angulo.LTSLF2,'r:o')
hold on
plot(t2,Marcha.Angulo.LTLFLS2,'c:o')
axis([0 45 80 200])
grid on
grid minor
xlabel('Cuadros por segundo')
ylabel('Angulo(grados)')
title('Marcha del Paciente Semana 0')
legend({'Rodilla Izquierda','Pie Izquierdo'},'NumColumns',3)
legend('boxoff')

```