



UNIVERSIDAD POLITÉCNICA DE PUEBLA
Organismo Público Descentralizado del Gobierno de Puebla



INGENIERÍA MECATRÓNICA
Trabajo Práctico como requisito parcial para obtener el título de Ingeniero en Mecatrónica

PODOSCOPIO DIGITAL

Presenta:
Mario Alberto Xique Salazar

Asesor en la universidad:
Dr. José Pedro Sánchez Santana

Asesor en la empresa:
Dr. Obed Cortés Arturo

Juan C. Bonilla, Puebla a 03 Diciembre de 2018.

**DIRECCIÓN DEL PROGRAMA ACADÉMICO DE INGENIERÍA
MECATRÓNICA****ACTA DE EXAMEN**

En el Municipio Juan C. Bonilla, Puebla a 07 de DICIEMBRE del año 2018 siendo las 10:30 horas, se reunieron en el aula D3-206 de esta Universidad, los integrantes del jurado:

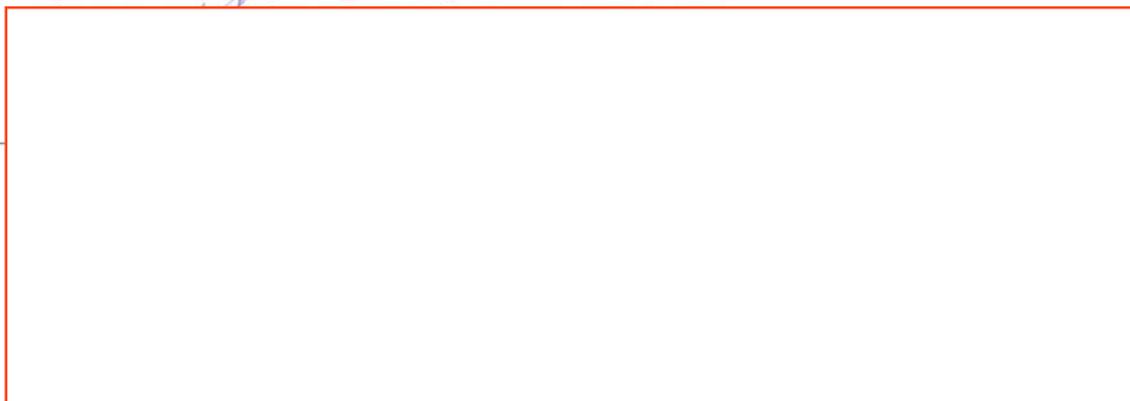
Presidente: DRA. RITA MARINA ACEVEZ PEREZ
Secretario: MTRA. LAISEL TOXQUI HUITZIL
Vocal: DR. MARCO ANTONIO CANCHOLA CHAVEZ

Y de acuerdo a las disposiciones reglamentarias en vigor se procedió a efectuar el examen que para obtener el título de Ingeniero(a) Mecatrónico(a) presenta el/la C. MARIO ALBERTO XIQUE SALAZAR con matrícula número 131400231.

Tomando en cuenta el contenido del trabajo cuyo título es: PODOSCOPIO DIGITAL

que fue dirigido por DR. BAFAEL ROJAS RODRIGUEZ y codirigido por DR. MARCO ANTONIO CANCHOLA CHAVEZ, una vez concluida la presentación oral se decidió que fuera: APROBADO.

El/La presidente del jurado hizo saber al sustentante el resultado obtenido, el código de ética y le tomó la protesta de ley, dándose por terminado el acto a las 11:45 horas y una vez leída y aprobada la presente fue firmada por las personas que en el acto intervinieron.





SECRETARÍA DE EDUCACIÓN PÚBLICA
GOBIERNO DE PUEBLA



SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA



Carta de Presentación/Aceptación para Realizar Estancia/Estadía

Juan C. Bonilla, Puebla a 17 de Septiembre de 2018

Dr. Obed Cortés Aburto
Profesor de Tiempo Completo del
PE de Ingeniería Mecatrónica UPPue
Presente:

Los estudiantes de la Universidad Politécnica de Puebla, como parte de su formación académica y profesional, deben realizar de manera obligatoria su Estadía dentro de una empresa o institución relacionada con algún área de especialización de sus estudios respectivos, con la intención de adquirir pertinencia y experiencia laboral en cada ciclo formativo.

Estas actividades se desarrollarán durante un cuatrimestre, que comprende 600 horas de trabajo, distribuidas de acuerdo al convenio al que se llegue con la empresa.

Es importante destacar que los estudiantes tienen la obligación de mantener la confidencialidad de la información derivada de la Estadía, y además, durante el desarrollo de ésta, no generarán relación laboral alguna con la Unidad Productiva o Social, ya que ellos cuentan con seguro social facultativo que les cubre la atención médica.

Agradecemos las facilidades brindadas al estudiante:

Nombre: Xique Salazar Mario Alberto
Número de matrícula: 131400231
Programa académico: Ingeniería en Mecatrónica
Actividades a desarrollar: Estadía en Mecatrónica
Duración: 600 Hrs.
Fecha de inicio: 13 de Septiembre de 2018 Fecha de término: 7 de Diciembre de 2018
Asesor por parte de la Unidad Productiva o Social: Dr. Obed Cortés Aburto
Asesor por parte de la Universidad: José Pedro Sánchez Santana

De conformidad, las partes se comprometen a cumplir con lo mencionado anteriormente.

Por la Universidad

Por la Unidad Productiva o

Estudiante



UPPue
Universidad Politécnica de Puebla

"Generamos Ciencia y Tecnología"
Tercer Carril del Ejido Serrano S/N, San Mateo Cuanalá,
Mpio. Juan C. Bonilla, Puebla, México .
C.P. 72640 Tels: (222) 774 66 40 al 46

UTP
COORDINACIÓN GENERAL DE UNIVERSIDADES
TECNOLÓGICAS y POLITÉCNICAS

Certificada en ISO 9001:2015 Evaluada en el Nivel 1 por CIEES
Certificada en NMX-R-025-SCFI-2015 en Igualdad Laboral y no Discriminación



UNIVERSIDAD POLITÉCNICA DE PUEBLA
Organismo Público Descentralizado del Estado de Puebla
ENCUESTA DE SEGUIMIENTO, EVALUACIÓN DE SATISFACCIÓN DEL EMPLEADOR Y LIBERACIÓN DE ESTANCIA/ESTADIA.

Nombre de la Empresa:	Universidad Politécnica de Puebla.			Fecha:	25/10/18.
Tamaño de la Empresa:	Micro (1-10)		Pequeña (11-50)		
	Mediana (51-250)		Grande (Más de 251)		
Sector de la Empresa:	Público	<input checked="" type="checkbox"/>	Privado		
Nombre del Evaluador:	DR. Obed Cortes Aburto.				
Teléfono del Evaluador:	222746656		E-mail:	obed.cortes@uppuebla.edu.mx	
Nombre del Estudiante:	Mario Alberto Xique Salazar.				
Programa Académico:	Ingeniería mecatrónica.		Área asignada:	Proyectos.	
	Estancia 1		Estancia 2	Estadia	<input checked="" type="checkbox"/>
	Seguimiento	<input checked="" type="checkbox"/>	Evaluación		

Favor de evaluar el nivel de satisfacción de 1 a 5, de acuerdo al desempeño del estudiante, con base en la siguiente escala de valores:

**1 Insatisfecho = 0%, 2 Poco satisfecho 25%, 3 Regular satisfacción 50%,
4 Buena satisfacción 75%, 5 Muy satisfecho 100%**

1	¿Aplica razonamiento lógico y analítico?	5
2	¿Es puntual y asiste?	5
3	¿Asume responsabilidades?	5
4	Domina alguna lengua extranjera?	4
5	¿Es hábil para relacionarse?	5
6	¿Su presentación personal es adecuada?	5
7	¿Desarrolla habilidad para trabajar en equipo?	5
8	¿Muestra habilidad de dirección y liderazgo?	4
9	¿Se interesa por la búsqueda de información pertinente y actualizada?	4
10	¿Manifiesta disposición para aprender constantemente?	5
11	¿Adquirió competencias durante el proyecto asignado?	5
	Total	52

¿Considera liberada la estancia/estadia del estudiante por su desempeño? Sí ___ No X

¿Qué aspectos sugiere usted, para lograr una adecuada pertinencia con su empresa?



UNIVERSIDAD POLITÉCNICA DE PUEBLA
 Organismo Público Descentralizado del Estado de Puebla
**ENCUESTA DE SEGUIMIENTO, EVALUACIÓN DE SATISFACCIÓN
 DEL EMPLEADOR Y LIBERACIÓN DE ESTANCIA/ESTADIA.**

Nombre de la Empresa:	Universidad Politécnica de Puebla			Fecha:	03/12/18
Tamaño de la Empresa:	Micro(1-10)		Pequeña(11-50)		
	Mediana (51-250)		Grande (Más de 251)	X	
Sector de la Empresa:	Público	X	Privado		
Nombre del Evaluador:	Dr. Obed Cortes Aburto.				
Teléfono del Evaluador:	2227746656		E-mail:	obed.cortes@uppuebla.edu.mx	
Nombre del Estudiante:	Mario Alberto Xique Salazar.				
Programa Académico:	Mecatrónica.		Área asignada:	Proyectos.	
	Estancia 1		Estancia 2		Estadia X
	Seguimiento		Evaluación	X	

Favor de evaluar el nivel de satisfacción de 1 a 5, de acuerdo al desempeño del estudiante, con base en la siguiente escala de valores:

**1 Insatisfecho = 0%, 2 Poco satisfecho 25%, 3 Regular satisfacción 50%,
 4 Buena satisfacción 75%, 5 Muy satisfecho 100%**

- | | | |
|----|---|-----------|
| 1 | ¿Aplica razonamiento lógico y analítico? | 5 |
| 2 | ¿Es puntual y asiste? | 5 |
| 3 | ¿Asume responsabilidades? | 5 |
| 4 | Domina alguna lengua extranjera? | 4 |
| 5 | ¿Es hábil para relacionarse? | 5 |
| 6 | ¿Su presentación personal es adecuada? | 5 |
| 7 | ¿Desarrolla habilidad para trabajar en equipo? | 5 |
| 8 | ¿Muestra habilidad de dirección y liderazgo? | 5 |
| 9 | ¿Se interesa por la búsqueda de información pertinente y actualizada? | 5 |
| 10 | ¿Manifiesta disposición para aprender constantemente? | 5 |
| 11 | ¿Adquirió competencias durante el proyecto asignado? | 5 |
| | Total | 54 |

¿Considera liberada la estancia/estadia del estudiante por su desempeño? Sí No

¿Qué aspectos sugiere usted, para lograr una adecuada pertinencia con su empresa?



SECRETARÍA DE EDUCACIÓN PÚBLICA
GOBIERNO DE PUEBLA



SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA



ACTA DE REVISIÓN DE DOCUMENTO DE ESTADÍA

En el Mpio. de Juan C. Bonilla, Puebla, a 29 de noviembre de 2018, se designó a los miembros de la Comisión Revisora de la Estadía por parte de la Academia de Profesores de Ingeniería Mecatrónica de la Universidad Politécnica de Puebla para examinar el documento del proyecto de Estadía intitulado:

Podoscopio Digital

Presentado por el alumno:

Mario Alberto Xique Salazar

con número de matrícula 131400231, aspirante al grado de

Licenciado en Ingeniería en Mecatrónica

Después de satisfacer los requisitos señalados por las disposiciones reglamentarias vigentes, los miembros de la Comisión manifestaron **APROBAR** el documento del proyecto de Estadía.

LA COMISIÓN REVISORA

Dr. José Pedro Sánchez Santana
Asesor de Estadía

Dra. Rit

MGIT. Grisel Toxqui Huitzil
Secretario

Marco Antonio Canchola Chávez
Vocal

Dra
Direct

érez
trónica

UPPue
Universidad Politécnica de Puebla

UNIVERSIDAD POLITÉCNICA DE PUEBLA
"Generamos Ciencia y Tecnología"
Tercer Carril del Ejido Serrano S/N, San Mateo Cuanalá,
Mpio. Juan C. Bonilla, Puebla, México ·
C.P. 72640 Tels: (222) 774 66 40 al 46

UTP
COORDINACIÓN GENERAL DE UNIVERSIDADES
TECNOLÓGICAS Y POLITÉCNICAS



SECRETARÍA DE
EDUCACIÓN PÚBLICA
GOBIERNO DE PUEBLA



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



CARTA CESIÓN DE DERECHOS

En el Mpio. de Juan C. Bonilla, Puebla, el día 5 de diciembre del 2018, el que suscribe Mario Alberto Xique Salazar alumno del Programa Académico de Ingeniería Mecatrónica con número de matrícula 131400231, manifiesta que es autor intelectual del presente trabajo documental de Estadía bajo la dirección del Dr. José Pedro Sánchez Santana y cede los derechos del trabajo intitulado "Podoscopio digital", a la Universidad Politécnica de Puebla para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección jose.sanchez@uppuebla.edu.mx Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

[Redacted signature box]

Mario Alberto Xique Salazar

Certificada en ISO 9001:2015 Evaluada en el Nivel 1 por CIEES Certificada en NMX-R-025-SCFL-2015 en Igualdad Laboral y no Discriminación

UPPue
Universidad Politécnica de Puebla

UNIVERSIDAD POLITÉCNICA DE PUEBLA
"Generamos Ciencia y Tecnología"
Tercer Carril del Ejido Serrano S/N, San Mateo Cuanalá,
Mpio. Juan C. Bonilla, Puebla, México ·
C.P. 72640 Tels: (222) 774 66 40 al 46

UTP
COORDINACIÓN GENERAL DE UNIVERSIDADES
TECNOLÓGICAS Y POLITÉCNICAS

RESUMEN

El presente trabajo muestra el diseño e implementación de un sistema basado en el procesamiento de imágenes el cual determina de manera automática si un pie presenta algún tipo de deformidad. Se han investigado los diferentes protocolos existentes para la determinación del tipo de pie, y se ha elegido el índice de clasificación propuesto por Hernández-Corvo debido a que por su proceso gráfico basado en líneas rectas y tangentes, es ideal para la creación de un algoritmo que detecte de manera automática las mediciones de la huella plantar, además de ser un método preciso y que clasifica en 6 tipos de deformaciones del pie, como resultados se logró la construcción de un dispositivo no invasivo al paciente con una interfaz gráfica sencilla e intuitiva y que crea un archivo STL para la fabricación de una plantilla plantar en impresión 3D ideal para cada tipo de pie.

ABSTRAC

The present work shows the design and implementation of a system based on image processing which automatically determines if a foot presents some kind of deformity. The different existing protocols for the determination of the foot type have been investigated, and the classification index proposed by Hernández-Corvo has been chosen because its graphical process based on straight lines and tangents is ideal for the creation of an algorithm. that automatically detects the measurements of the footprint, in addition to being an accurate method and classified into 6 types of foot deformations, as results were achieved the construction of a non-invasive device to the patient with a simple and intuitive graphic interface and that creates an STL file for the manufacture of a template in 3D printing ideal for each foot type.

ÍNDICE GENERAL

RESUMEN	I
ABSTRAC	I
Índice de Figuras	IV
Índice de Tablas	V
Capítulo 1.....	1
Introducción	1
1.1 Universidad Politécnica de Puebla.....	1
1.2 Antecedentes	3
1.2.1 Pedígrafo	3
1.2.2 Fotopodograma.....	3
1.2.3 Podoscopio.....	4
1.2.4 Baropodometría	4
1.3 Trabajos previos	5
1.3.1 Sistema de análisis de las presiones plantares.....	5
1.3.2 Podoscopio digital.....	5
1.3.3 Detección de deformaciones provocadas por el pie diabético.....	6
1.3.4 Tipificación de la huella plantar de escolares entre 6 y 8 años.....	6
1.4 Definición del problema.....	7
1.5 Justificación.....	8
1.6 Objetivos.....	9
1.6.1 Objetivo General:	9
1.6.2 Objetivos Específicos:.....	9
1.7 Cronograma de actividades.....	10
Capítulo 2	11
Marco Teórico	11
2.1 Fundamentos básicos del pie humano	11
2.1.1 El pie y su estructura	11
2.1.2 Funciones del pie.....	11
2.1.3 Bóveda plantar	12
2.2 Las huellas o impresiones plantares.	13
2.3 Clasificación de las huellas o impresiones plantares.....	13
2.3.1 Pie plano.....	14
2.3.2 Pie plano normal	14

2.3.3	Pie normal	14
2.3.4	Pie normal cavo	15
2.3.5	Pie cavo	15
2.3.6	Pie cavo fuerte	15
2.3.7	Pie cavo extremo	16
2.4	Métodos de análisis y obtención de la huella plantar	16
2.5	El método de Hernández-Corvo	17
Capítulo 3.....		19
Desarrollo		19
3.1	Diseño estructural de la plataforma	19
3.1.1	Simulación de la base (SolidWorks)	20
3.5	Diseño del sistema de captura de imagen.....	21
3.3.3	Proceso físico	21
3.3.7	Proceso de refracción en las imágenes	21
3.3.6	Kinect.....	22
3.2.4	Espejo	23
3.2.5	Vidrio	23
3.3	Conexión Kinect Matlab.....	24
3.4	Iluminación	25
3.5	Desarrollo de la interfaz.....	26
3.6	Descripción de la Interfaz gráfica de usuario.....	26
3.6.1	Modo espejo	27
3.6.2	Recorte de imagen	27
3.6.3	División de color	28
3.6.4	Convertir imagen en binario	29
3.6.5	Relleno de huecos	30
3.6.6	Remover fondo de la imagen.	31
3.6.7	Mediciones en la imagen.	31
3.6.8	Resultados de tipo de pie.	33
3.7.9	Modelado de plantilla.	33
Capítulo 4.....		34
4.1	Pruebas del hardware y software.....	34
4.1.1	Paciente 1.....	34
4.1.2	Paciente 2	36
4.2	Conclusiones	37
Referencias		38
Anexos		39

1. Plano de la estructura.....	39
2. Lamina	40
3. Vidrio	41
4. Espejo	42
5. Código Matlab.....	43

Índice de Figuras

Figura 1. Mapa de la ubicación de la UPPue.	2
Figura 2. Pedígrafo. [1]	3
Figura 3. Fotopodograma. [1].....	3
Figura 4. Podoscopio. [2].....	4
Figura 5. Análisis plantar baropodómetro electrónico. [3]	4
Figura 6. Sistema de análisis de las presiones plantares. [10]	5
Figura 7. Mediciones tomadas por el podoscopio digital. [11]	5
Figura 8. Mediciones tomadas dispositivo detector de deformaciones. [12].....	6
Figura 9. Distribución por tipificación de Pie en ambos Miembros Inferiores. [13]	6
Figura 10. Diagrama a bloques del proceso. [Fuente autores]	7
Figura 11. El pie humano la composición. [4]	12
Figura 12. Estructura de la bóveda plantar. [4].....	12
Figura 13. Pie plano [1]	14
Figura 14. Pie plano normal [1]	14
Figura 15. Pie normal. [1]	14
Figura 16. Pie normal cavo [1].....	15
Figura 17. Pie cavo normal [1].....	15
Figura 18. Pie cavo fuerte. [1]	15
Figura 19. Pie cavo extremo. [1]	16
Figura 20. Protocolo de Hernández- Corvo [6].....	18
Figura 21. Diseño estructural. [SolidWorks].....	19
Figura 22. Análisis de tensiones de la base. [SolidWorks].....	20
Figura 23. Análisis de desplazamiento. [SolidWorks].....	20
Figura 24. Representación de los principios físicos. [7]	22
Figura 25. Kinect componentes. [9]	22
Figura 26. Resistencia a la flexión. [7]	23
Figura 27. Adaptador usb Kinect. [9].....	24
Figura 28. Drivers SDK v1.8. [9]	24
Figura 29. Tira de LED. [Fuente autores].....	25
Figura 30. Interfaz gráfica de usuario. [Fuente autores].....	26
Figura 31. Toma de captura. [Fuente autores].....	27

Figura 32. Modo espejo. [Fuente autores].....	27
Figura 33. Distancia x (Pie derecho) [Fuente autores]	28
Figura 34. Distancia x (Pie izquierdo) [Fuente autores]	28
Figura 35. Distancia y (Pie derecho) [Fuente autores]	28
Figura 36. Distancia y (Pie izquierdo) [Fuente autores]	28
Figura 37. Extracción de color. [Fuente autores]	29
Figura 38. Imagen negativa. [Fuente autores]	29
Figura 39. Imagen en binario. [Fuente autores].....	30
Figura 40. Rellenado de pixeles [Fuente autores].....	30
Figura 41. Eliminación del fondo. [Fuente autores]	31
Figura 42. Resultados de las mediciones. [Fuente autores].....	33
Figura 43. Plantilla sugerida. [Fuente autores]	33
Figura 44. Ubicación del paciente. [Fuente autores]	34
Figura 45. Forma en la que se ubican los pies. [Fuente autores].....	35
Figura 46. Procesamiento de la huella plantar. [Fuente autores].....	35
Figura 47. Resultados del paciente 2. [Fuente autores].....	36

Índice de Tablas

Tabla 1. Cronograma	10
Tabla 2. Métodos de análisis y obtención de la huella plantar	16
Tabla 3. Valoración del tipo de pie según el índice de Hernández-Corvo ¡Error! Marcador no definido.	

Introducción

1.1 Universidad Politécnica de Puebla

La Universidad Politécnica de Puebla, pertenece a un subsistema de educación superior de la Secretaría de Educación Pública (SEP). Este subsistema está orientado por la Coordinación General de Universidades Politécnicas y tiene a bien desarrollar la investigación aplicada al desarrollo tecnológico bajo programas educativos a nivel ingeniería, licenciatura y estudios de posgrado con base en el Modelo Educativo Basado en Competencias, al mismo tiempo, llevan una colaboración estrecha con organizaciones de los sectores productivo, público y social.

La universidad a partir de Agosto de 2006 se encuentra ubicada en Tercer Carril del Ejido Serrano S/N, San Mateo Cuanalá, con instalaciones pertinentes para su operatividad: edificios de docencia, laboratorios, talleres, biblioteca y espacios para el deporte y la cultura.

La ubicación favorece tanto a la comunidad educativa como a la empresarial (ver Figura 1 para identificar el plano de ubicación), por la cercanía a los corredores industriales de Huejotzingo, Quetzalcóatl, San Martín, Parque Industrial FINSA, Corredor Industrial de Estado de Tlaxcala y el Aeropuerto Internacional Hermanos Serdán.

A través de una Política de calidad y de mejora continua, la UPPue ha concretado una misión y visión en su quehacer universitario, a través de su misión y visión institucional. Cuenta con la Misión de formar integralmente a profesionales competentes que atienden necesidades de los sectores productivo y social, mediante el desarrollo tecnológico, la innovación y la investigación aplicada. Su Visión al 2030 es ser una Universidad acreditada nacional e internacionalmente por su modelo educativo, sus servicios y la calidad de sus egresados, en un marco de sustentabilidad y que consolida permanentemente alianzas estratégicas con los sectores productivo y social.

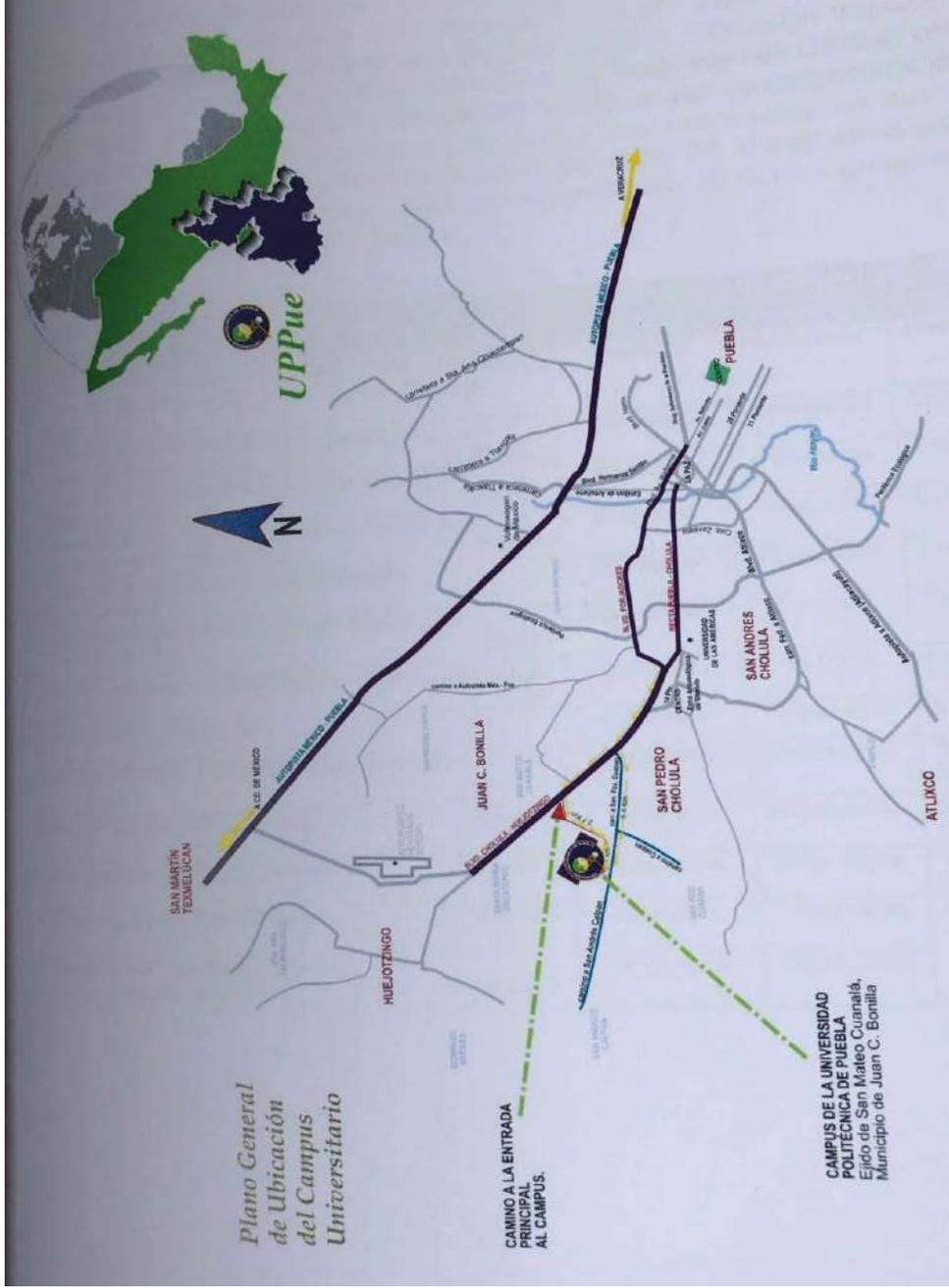


Figura 1. Mapa de la ubicación de la UPPue.

1.2 Antecedentes

1.2.1 Pedígrafo

Consiste en pisar sobre un dispositivo de goma impregnado en tinta Figura 2, bajo el cual hay una hoja de papel que tras la pisada se embebe de la tinta y señala la huella plantar para posteriormente realizar las debidas mediciones, el inconveniente con este método es que la tinta mal aplicada puede ocasionar que la huella no se mida correctamente además de ser desagradable e incómodo al paciente. [1].



Figura 2. Pedígrafo. [1]

1.2.2 Fotopodograma

Similar al pedígrafo solo que en este caso se realiza a través de la impresión de la planta del pie en un papel fotográfico que, al resultar saturado con líquido revelador, nos ofrece información sobre el contacto o la presión que realiza el pie sobre una superficie plana Figura 3. Esto sirve para completar un diagnóstico y para valorar la evolución de un tratamiento. Las imágenes que se consiguen con este método proporcionan un perímetro nítido y claro de la porción del pie que se apoya [1].



Figura 3. Fotopodograma. [1]

1.2.3 Podoscopio

El podoscopio Figura 4, es un equipo sencillo, de muy fácil utilización es de gran apoyo en las evaluaciones de las formas del soporte plantar. Las formas iniciales del podoscopio constituían la simple observación de las formas del soporte plantar; era posible incorporar algunas mediciones no invasivas; pero en general solo se trataba de una visión “espejo” de los rasgos plantares del sujeto motivo de análisis. [2]



Figura 4. Podoscopio. [2]

1.2.4 Baropodometría

El baropodómetro electrónico Figura 5 es un equipo compuesto por una plataforma con sensores de presión que analizan la distribución de las cargas en los pies de forma estática y durante el ciclo de la marcha mediante del análisis del apoyo plantar, este equipo representa lo último en tecnología para el estudio del pie y la marcha ya que mediante este análisis se puede detectar posibles alteraciones en la postura y la biomecánica del pie [3].



Figura 5. Análisis plantar baropodómetro electrónico. [3]

1.3 Trabajos previos

1.3.1 Sistema de análisis de las presiones plantares.

Creado por Carreño J. C. Y Cabrera P. T. Figura 6. Se trata de un sistema que obtiene la impresión de la huella plantar del pie para luego aplicar un procesamiento y posteriormente obtener las distribuciones de las presiones sobre la superficie plantar de pie de esta manera se obtiene un índice de cavidad y de esta manera determinar el tipo de pie. [10]



Figura 6. Sistema de análisis de las presiones plantares. [10]

1.3.2 Podoscopio digital.

Está basado en la captura de la imagen de la huella plantar a través de un escáner convencional de uso común Figura 7, y en el desarrollo de un algoritmo para el procesamiento de las imágenes, y una representación en una consola de visualización en el monitor de la computadora, proporciona ciertas herramientas de medición y análisis de la huella en la misma interfaz. Asimismo, incluye la construcción de una plataforma de acrílico transparente e iluminada que permite al paciente colocar su huella plantar para ser capturada óptimamente. El software base utilizado para el análisis, procesamiento y presentación de las imágenes plantares estudiadas es Matlab. [11]



Figura 7. Mediciones tomadas por el podoscopio digital. [11]

1.3.3 Detección de deformaciones provocadas por el pie diabético.

Creado por se trata de un sistema que permite la detección temprana de deformidades en el pie ocasionadas por el pie diabético, el sistema hace uso de una WebCam, una pc y un algoritmo desarrollado en Matlab, el estudio hace uso del método de Hernández corvo y un clasificador propuesto, entre la información que aporta este dispositivo están mediciones del pie, índice corporal del pie, clasificación del tipo de pie y el grado de deformación Figura 8, según sus autores el dispositivo tiene un porcentaje de error del 7.99%. [12]

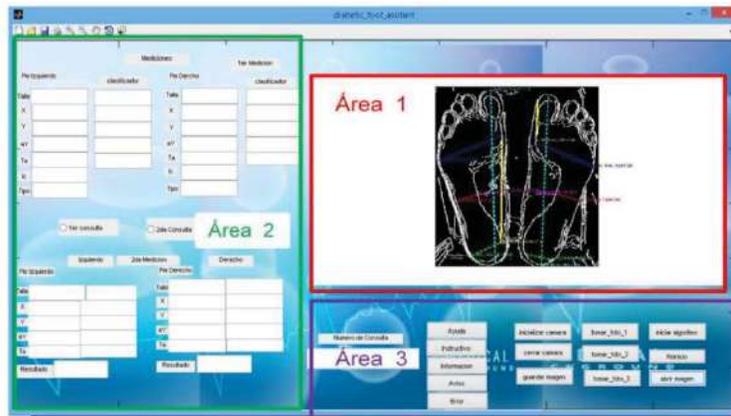


Figura 8. Mediciones tomadas dispositivo detector de deformaciones. [12]

1.3.4 Tipificación de la huella plantar de escolares entre 6 y 8 años.

Trinidad Z. Ha hecho un trabajo de investigación en el que se ha determinado las características de la huella plantar de niños y niñas entre los seis y los ocho años de edad de población urbana de en el cual caracterizo las variaciones ocurridas en las mismas; con el fin de identificar las posibles causas de estas variaciones y las pautas de corrección que al respecto puedan realizarse Se utilizó test de valoración y antropometría predictiva para analizar el tipo de pie según el descenso de la bóveda plantar de acuerdo al método de impresión plantar con carga normal tomado con una unidad de podometría especializada, usando MatLab para el procesamiento de las imágenes y bajo el método de Hernández Corvo, los resultados de su investigación se observan en la Figura 9. [13]

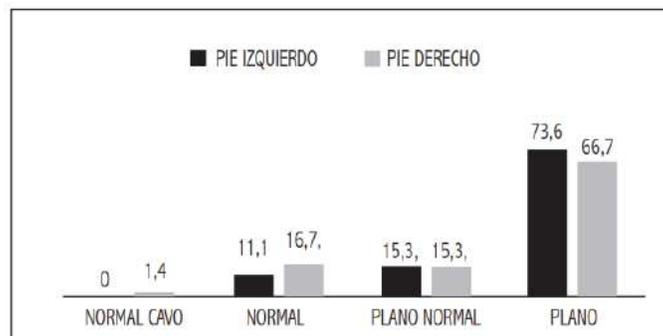
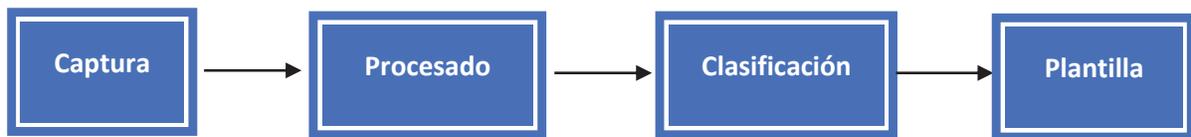


Figura 9. Distribución por tipificación de Pie en ambos Miembros Inferiores. [13]

1.4 Definición del problema

Durante años las técnicas usadas por podólogos para la identificación del tipo de pie han sido anticuadas, métodos como el pedigrafo y el fotopodograma son económicos pero tienen diversas desventajas entre las cuales se encuentran características como ser invasivos, dependiendo de la tinta esta puede complicar la acción de tomar mediciones en la huella, además de ser un acto incomodo al paciente, y se deben de realizar procedimientos gráficos que resultan tediosos al médico; debido a esto se ha tomado la decisión de crear un proyecto que consiste en el desarrollo de un dispositivo fundado en el principio de un Podoscopio, el cual estará basado en la captura de una imagen de las huellas plantares, para después realizar un procesamiento de la misma como se puede observar en el diagrama de la Figura 9, y utilizando el método de clasificación del tipo del pie propuesto por Hernández Corvo el cual ha sido seleccionado por usar un método grafico basado en líneas rectas y tangentes lo que facilita su incorporación en procesamiento, y posteriormente representarlas en una Interfaz Gráfica de Usuario mejorando así su interpretación al médico. Con una plataforma vidrio que permite al paciente colocar sus pies para obtener la huella plantar, lo que creara conformidad y seguridad al paciente.



*Figura 10. Diagrama a bloques del proceso.
[Fuente autores]*

1.5 Justificación

El pie plano es un padecimiento frecuente en la población preescolar y escolar esto genera problemas cuando se desea tomar un estilo de vida deportista, esta deformidad es susceptible a la corrección integral, siempre y cuando se establezca vigilancia y/o tratamiento oportuno. En otros casos las secuelas son irreversibles y en un futuro potencialmente podrán deteriorar la auto-imagen del individuo adulto, además de otros problemas como alteraciones en la coordinación motora, precisión, disminución del movimiento o alteraciones del equilibrio generando así deficiencias en el deporte.

Aunque en condiciones normales el pie plano tiende a una corrección espontánea, es deber del médico de primer nivel tener la pericia suficiente para la detección temprana, así también los factores de riesgo asociados. Dar seguimiento y derivación oportuna a segundo nivel, de esta manera limitar el daño que pudiese generar una mala alineación corporal.

La finalidad del presente trabajo es la identificación temprana de los factores predisponentes asociados al pie plano además que puede condicionar alteraciones diversas, que afectarían la actividad y la calidad de vida del individuo en su vida futura.

1.6 Objetivos

1.6.1 Objetivo General:

Diseñar e implementar un podoscopio, que hará uso de un procesamiento de imágenes el cual determinará si un pie es normal, cavo o plano de manera automática que facilitará el trabajo del ortopedista.

1.6.2 Objetivos Específicos:

- a) Investigar los diferentes métodos de clasificación del tipo de pie con el propósito de elegir el mejor método que se base en la huella plantar.
- b) Diseñar y construir un podoscopio no invasivo con las especificaciones necesarias para la correcta adquisición de imágenes.
- c) Evaluar el sensor Kinect como dispositivo de escaneo para determinar si la precisión y las propiedades del sensor serán suficientes para la adquisición de imagen.
- d) Crear una interfaz gráfica de usuario (GUI) que será usada como herramienta visual para la interpretación y evaluación de la huella plantar.
- e) Crear un algoritmo de reconocimiento automático de medias en una huella plantar y clasificación de ésta, usando el método que se adecue para él procesamiento.
- f) Prototipar una plantilla para un paciente con deformidad en el pie, usando la tecnología de Impresión 3D.

1.7 Cronograma de actividades

Actividad	Octubre						Noviembre	
	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8
Recabar información.								
Diseño y Creación de estructura.								
Implementación de dispositivos electrónicos.								
Creación del código fuente para la adquisición de datos.								
Etapas de pruebas.								
Resultados.								
Tesis								

Tabla 1. Cronograma

Marco Teórico

2.1 Fundamentos básicos del pie humano

2.1.1 El pie y su estructura

El pie humano es uno de los órganos más complejos que tiene nuestro cuerpo, por tal motivo posee varias funciones, una de ellas es la de ser gestora de la marcha, que hace de recepción, amortiguación, sostén de propulsión. Además de ser un órgano de vital importancia en las reacciones de equilibrio, capaz de ser el informante perfecto de nuestros movimientos con respecto al suelo gracias a las medidas de presión.

El pie es el elemento clave en la evolución hacia la verticalidad, lo que ha permitido ser el único mamífero que vive realmente una postura de forma bípeda y mantenerse en equilibrio, hacer cosas fabulosas desde correr y saltar con toda seguridad, ha permitido al hombre vivir en toda la plenitud y gozar de la libertad.

El pie es la extremidad más distal del miembro inferior, se le considera como una estructura perfecta diseñada para aguantar nuestro peso y poder desplazarnos con toda seguridad por diversos terrenos y en circunstancias desfavorables para el hombre [4].

2.1.2 Funciones del pie

Para asimilar la gestión de los pies, debemos analizar la función la cual se fundamenta en ser el soporte primordial para poder desplazarnos de forma bípeda, para todo aquello se debe cumplir con 4 funciones básicas:

- De absorción constante a los impactos.
- De generar las suficientes fuerzas de propulsión para caminar.
- De traslación las fuerzas de rotación generados por las caderas.
- De adaptación a superficies irregulares.

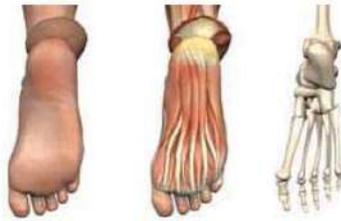


Figura 11. El pie humano la composición. [4]

2.1.3 Bóveda plantar

En la Figura 8 se puede apreciar la estructura como una especie de arco que se forma en la parte interna del pie con el fin de soportar de mejor manera las fuerzas de presión y de carga, esta de arco presenta varias particularidades la primera que es cóncava hacia abajo presentando tres puntos de apoyo, los mismos que son; un apoyo posterior (hueso calcáneo) y dos apoyos anteriores, por dentro la cabeza del primer metatarsiano y por fuera la cabeza del quinto, en fin el sistema para que el pie soporte carga y presión, se da a través de los tres puntos de apoyo, es decir mediante los cuales es sustentado por el también llamado trípode plantar [4].

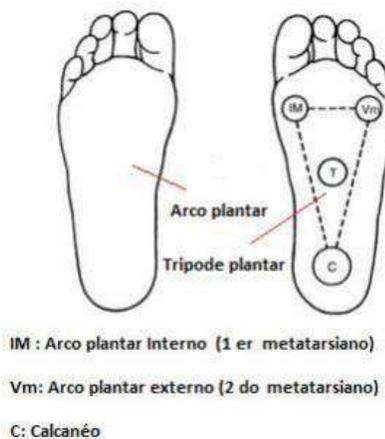


Figura 12. Estructura de la bóveda plantar. [4]

El trípode plantar que forma al pie, requiere que los puntos de apoyo estén conectados de forma que se genere entre ellos músculos (arcos plantares), que están unidos por bandas aponeuróticas, de aquí se generaran tres arcos importantes, el arco plantar interno, arco plantar externo y por último el arco anterior.

- a. **Arco plantar Interno:** Como su nombre lo dice se genera en la parte interna del pie, une el punto de apoyo del calcáneo con la cabeza del primer metatarsiano.
- b. **Arco plantar Externo:** une el apoyo del calcáneo con la cabeza del quinto metatarsiano, formándose así el arco principal externo.
- c. **Arco Anterior:** Une los puntos de apoyo de los miembros que componen del primer al quinto metatarsiano.

2.2 Las huellas o impresiones plantares.

La Podometría es la técnica que tiene como objetivo medir las presiones que se aplican sobre la planta del pie o sobre una región específica de la misma. Los podoscopios sobre todo los más recientes, representan las posibilidades interpretativas que ofrecen las impresiones o huellas plantares, ligadas por varias metodologías resultan un inapreciable aliado en los estudios y valoraciones las de entintado, utilizan tintas dactilografías solubles en mayor o menor grado, las impresiones de los fotopodogramas son un poco más caras ya que utilizan papel fotográfico y por último las denominadas de altas resoluciones que son mediante plataformas con múltiples sensores integrados donde se puede apreciar mediante escalas de colores las regiones de mayor o menor carga, en la Figura 8 se puede observar los tres tipos de impresiones plantares en la actualidad [2].

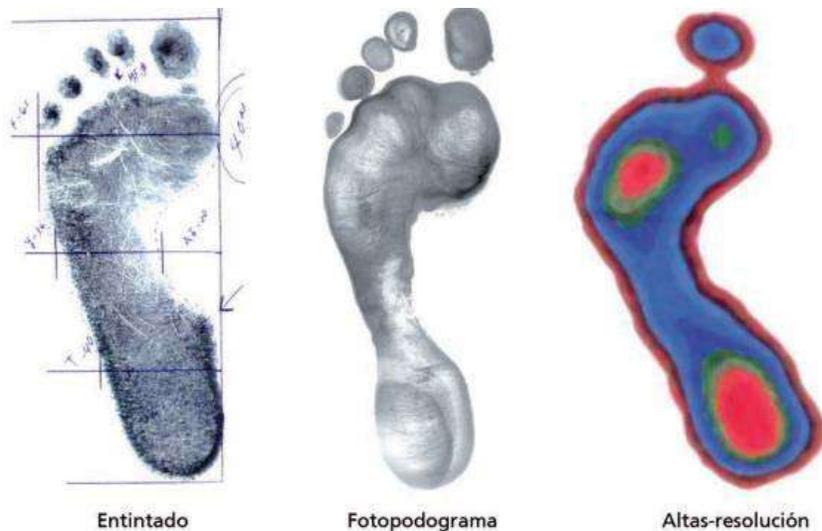


Figura 13. Impresiones plantares. [4]

2.3 Clasificación de las huellas o impresiones plantares

Inicialmente las impresiones plantares fueron catalogadas en planas, normales y cavas; posteriormente se comprobó que resultaban insuficientes y que entre ellas podían establecerse criterios intermedios; llegando finalmente al establecimiento de dos categorías para las planas; dos para las denominadas como normales y tres para las conocidas como impresiones cavas [2].

- Planas; Planas-Normales.
- Normales; Normal-Cavo.
- Cavo; Cavo-Fuerte; Cavo-Extremo.

2.3.1 Pie plano

Corresponde a la estructura del pie en donde se observa que carece de la bóveda plantar, transformando el arco plantar en un aplanamiento de la superficie plantar, por lo tanto el pie no se apoya sobre los tres puntos del trípede plantar sino sobre toda la planta. En este caso la forma de pisar es hacia el interior de forma pronadora [5].

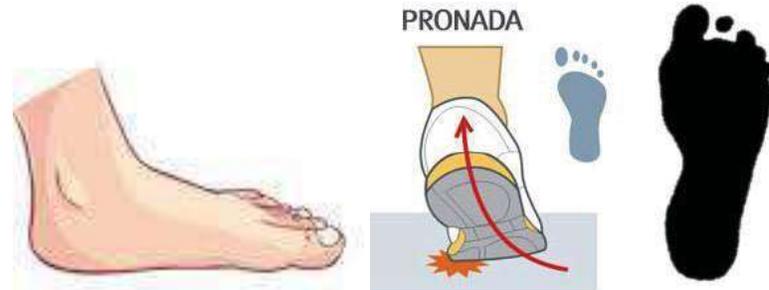


Figura 13. Pie plano [1]

2.3.2 Pie plano normal

Este tipo de pie se observa parte de la bóveda plantar, es decir se presenta en la impresión plantar un arca mínimo en la zona de la bóveda plantar presentándose de igual manera área mayoritaria de pisada. En este caso igual la pisada sigue siendo hacia dentro, es decir la pisada interna [5].



Figura 14. Pie plano normal [1]

2.3.3 Pie normal

El pie normal se presenta de la forma más natural es decir se realiza una pisada de adentro hacia afuera es decir primero los metatarsos y después la zona del calcáneo, en este caso el desgaste del zapato es uniforme [5].

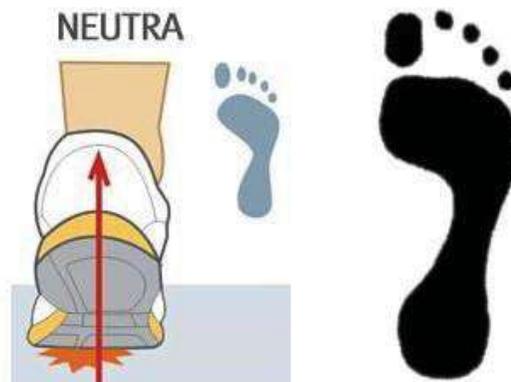


Figura 15. Pie normal. [1]

2.3.4 Pie normal cavo

Este tipo de pie tiene una cualidad que el arco plantar empieza a disminuir su medida horizontalmente, es decir se presenta desgaste en la forma exterior del zapato haciendo que se pise más hacia esta zona, logrando disminuir la zona de contacto en la parte central del pie [5].



Figura 16. Pie normal cavo [1]

2.3.5 Pie cavo

Consiste en una elevación atípica de la bóveda plantar, los dedos quedan agarrotados hacia adentro, dando origen a que se presente desgaste en el zapato en la parte exterior hacia afuera [5].



Figura 17. Pie cavo normal [1]

2.3.6 Pie cavo fuerte

Como su nombre lo indica es que el pie tiene sus dedos flexionados a tal punto que llega a disminuir el tamaño del pie, haciéndose más notorio el desgaste del zapato y la forma de la pisada más hacia afuera [5].



Figura 18. Pie cavo fuerte. [1]

2.3.7 Pie cavo extremo

Aquí el pie se apoya sobre el borde externo, mientras que en el borde interno se encuentra elevado, es decir es un defecto de posición estructural del pie. Al ser esta patología no muy común es fácil identificar este tipo ya que el pie no deja huella en la parte central de la planta, ni en los dedos [5].



Figura 19. Pie cavo extremo. [1]

2.4 Métodos de análisis y obtención de la huella plantar

Las mediciones del pie se suelen hacer con la intención de prescribir órtesis o ayudar en el descubrimiento de factores de riesgo y lesiones deportivas o de otro tipo. Existe una gran cantidad de métodos para cuantificar la huella plantar y establecer una clasificación del tipo de pie (Tabla 2).

Tabla 2. Métodos de análisis y obtención de la huella plantar. [3]

MÉTODOS DE ANÁLISIS	AUTORES (AÑO/S)	PARÁMETRO MEDIDO	VENTAJAS	INCONVENIENTES
Inspección visual no cuantitativa	Cavanagh y Rodgers (1987).	Forma del pie por observación visual.	Rápido y útil.	Subjetivo, necesidad mucha experiencia examinador.
Ángulo tibio-calcáneo (ATC) con goniómetro	Viladot (2000) y Albert (2009).	Ángulo entre la tibia y el calcáneo.	Sencillez, determina pie valgo o varo.	Subjetivo, necesidad mucha experiencia examinador.
Altura del dorso del pie	Cowan, et al. (1993); Williams y McClay (2000).	Longitud más alta del dorso del pie tomada al 50% de la longitud real del pie.	Válido.	Subjetividad.
Altura del escafoides: palpación directa	Chu et al. (1995); Saltzman et al. (1995).	Medición tubérculo escafoides hasta el suelo para determinar medida ángulo interno.	Rápido, sencillo, no riesgo para los participantes.	Subjetivo, basado en medida estática.
Caída del escafoides	Sachthanandam y Joseph (1995); Shrader et al. (2005); Howard y Briggs (2006); Billis et al. (2007); Nielsen et al. (2008); Razeghi y Batt (2000).	Medida desde tuberosidad escafoides en descarga y después en apoyo bipodal.	Rápido, sencillo, no riesgo para los participantes.	Subjetivo, basado en medida estática.
Altura del escafoides normalizada	Williams y McClay (2000); Menz y Munteanu (2005); Scott et al. (2007); Nielsen et al. (2008).	Altura del escafoides entre longitud total del pie.	Rápido, sencillo, no riesgo para los participantes.	Subjetivo, basado en medida estática.
Altura del escafoides truncada	Cowan et al. (1993); Saltzman et al. (1995); Kaufman et al. (1999); Williams y MacClay (2000); Menz y Munteanu (2005); Murley et al. (2009).	Altura del escafoides entre la longitud truncada de la huella.	Proporciona la representación más válida del esqueleto óseo del pie.	Subjetividad.
Índice del Arco (IA)	Cavanagh y Rodgers (1987); Hamil et al. (1989); McCrory et al. (1997); Nikolaidou y Boudolos (2006).	Proporción entre las áreas de contacto de las diferentes partes de la huella plantar excluyendo los dedos.	Medida útil y un predictor válido de la altura del arco interno del pie.	Errores al determinar la superficie del pie.
Índice de la impresión del pie	Redmon et al. (2006); Redmon et al. (2008).	Método observacional de clasificación de la postura estática del pie.	Rápido y sencillo, toma datos de la huella plantar impresa.	Necesidad de experiencia previa por parte del examinador, destreza manual para la palpación y subjetividad.
Método de Hernández Corvo	Hernández (1989); Sirgo y Aguado (1991); Sirgo et al. (1997); Abián et al. (2005); López et al. (2006); Zurita et al. (2006); Abián et al. (2008); Aguilar et al. (2009).	Tipificar el pie según unas medidas que se realizan en base a la impresión plantar.	Buena precisión, tanto en la realización como en la clasificación del tipo de pie.	Basado en medida estática.

2.5 El método de Hernández-Corvo

El Índice de Hernández-Corvo permite determinar el tipo de pie mediante un protocolo establecido, así el pie se tipifica según las medidas obtenidas a través de la imagen plantar, dando como resultado seis posibilidades que abarcan desde el pie plano hasta el pie cavo extremo. Este índice presenta una buena precisión, tanto en la realización como en la clasificación del tipo de pie. Sobre cada huella se aplica el protocolo de valoración que se describe en los puntos (a-o). Este método ha sido elegido por ser un método gráfico que usa líneas rectas y tangentes lo que facilita la creación de un algoritmo que detecte automáticamente las medidas en cualquier tipo de pie. En la Figura 16 se puede observar el proceso que se lleva a cabo para poder obtener la clasificación según el método. [6]

- a) Marcación de los puntos 1 y 1' en las prominencias internas del antepié y del retropié respectivamente, figura 2A.
- b) Unión de los puntos 1 y 1' para formar el trazo inicial, figura 2B.
- c) Marcación de los puntos 2 y 2' en el extremo anterior y posterior de la huella respectivamente, figura 2C.
- d) Trazado de dos líneas perpendiculares al trazo inicial que pasen por 2 y por 2'.
- e) La distancia entre la línea que pasa por 2 y el punto 1 se llama medida fundamental (mf), 2D.
- f) Trazado de tres líneas perpendiculares al trazo inicial que pasen por las divisiones de la medida fundamental (se les denomina de arriba abajo 3, 4 y 5), figura 2E.
- g) Trazado de una línea entre 3 y 4 perpendicular a 3 (y paralela al trazo inicial), que pase por el punto más externo del pie. Se llama línea 6, Figura 2F.
- h) Medición del valor X, que es la distancia entre el trazo inicial y la línea 6 y que corresponde a la anchura del metatarso.
- i) Trazado de la línea 7, paralela al trazo inicial, que pasa por el punto más externo de la línea 4, Figura 2F.
- j) Trazado de la línea 8, paralela al trazo inicial, que pasa por el punto más externo del pie de la línea 5, 2F.
- k) Medición de la distancia entre la línea 8 y el trazo inicial.
- l) Trazado de la línea 9 paralela al trazo inicial y que pasa por el punto más externo de la zona interna entre 4 y 5, Figura 2F.
- m) Medición de la distancia Y, entre 9 y 7.
- n) Medición de la distancia entre la línea 9 y el trazo inicial.
- o) Se calcula el %X según la Ecuación 1.

$$\%X = \frac{(X-Y)}{X} * 100 \quad (1)$$

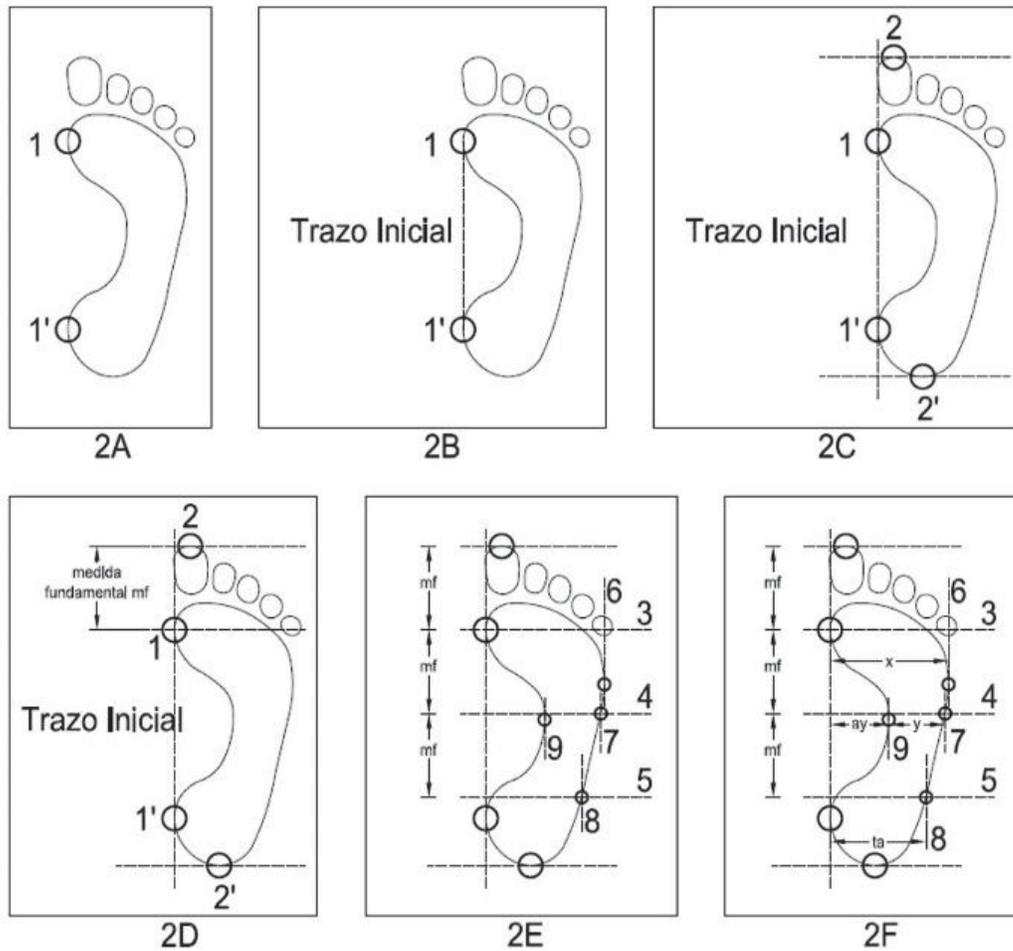


Figura 20. Protocolo de Hernández- Corvo [6]

La valoración del pie se da según el valor obtenido en la ecuación 1, el porcentaje se muestra en la Tabla 3 [6].

Tabla 3. Valoración del tipo de pie según el índice de Hernández-Corvo

%X	TIPO DE PIE
0-34	Plano
35-39	Plano - Normal
40-54	Normal
55-59	Normal - Cavo
60-74	Cavo
75-84	Cavo fuerte
85-100	Cavo extremo

3.1 Diseño estructural de la plataforma

Para realizar el diseño estructural de la plataforma, se toma en consideración varios aspectos en donde dicha estructura tiene que ser lo más confiable posible y que sea lo más compacta, siendo la misma un equipo que brinde seguridad a la persona que requiera el análisis de la superficie plantar, dicho diseño requerirá de consideraciones iniciales para garantizar su funcionamiento. A continuación, se presentan dichas consideraciones:

- Como medida general, el sistema debe tener un ancho de 400 mm y largo de 700mm.
- El sistema debe soportar un peso de 100 Kg
- Para el proceso físico de la captura de la imagen, nuestro sistema debe contar con una cámara la misma que va a estar colocado debajo de la superficie de descanso enfocando a un vidrio, la misma que ira colocada de forma inclinada por debajo de la superficie de pisada
- Debe constar de dos peldaños de ancho de 390mm y largo de 150 mm.
- La superficie en donde va a permanecer el paciente en etapa de bipedestación debe tener las siguientes medidas de ancho de 390 mm y largo de 400 mm.
- Como material refractante se utiliza vidrio de 10 mm.

La importancia de este tipo de análisis previo al diseño definitivo radica en la necesidad de realizar una valoración fiable y ordenada del aparato locomotor, que en nuestro caso se centra en la valoración del pie. El presente método de evaluación se aplicara después que la persona a evaluar este por lo menos 60 segundos, en la posición de bipedestación.

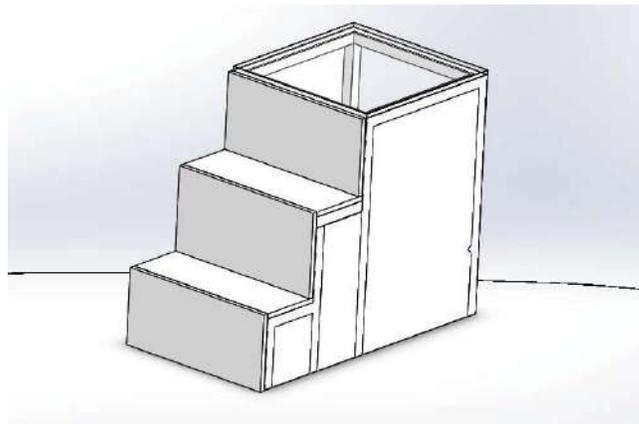


Figura 21. Diseño estructural. [SolidWorks]

3.1.1 Simulación de la base (SolidWorks)

SolidWorks es un programa de diseño y simulación de estructuras y mecanismos. El modo de análisis que se escogió dentro del programa, para realizar el análisis estructural fue en modo estático en donde se puede visualizar los resultados de forma gráfica y la posibilidad de reflejar estos comportamientos de forma numérica, los mismos que reflejan los esfuerzos o deflexiones que se somete la estructura de acuerdo a las exigencias cargadas en el programa.

- Se realiza el boceto de la estructura en el programa tomando en cuenta el tipo de material a usar.
- Se colocan las cargas en los perfiles donde va a estar alojado la superficie refractante (vidrio), la misma donde va a ser soportado el peso completo de la persona.

En la Figura 19 se muestra el análisis de tensiones y en la Figura 20 se muestra el análisis de desplazamiento que tendrá la base con las consideraciones antes mencionadas.

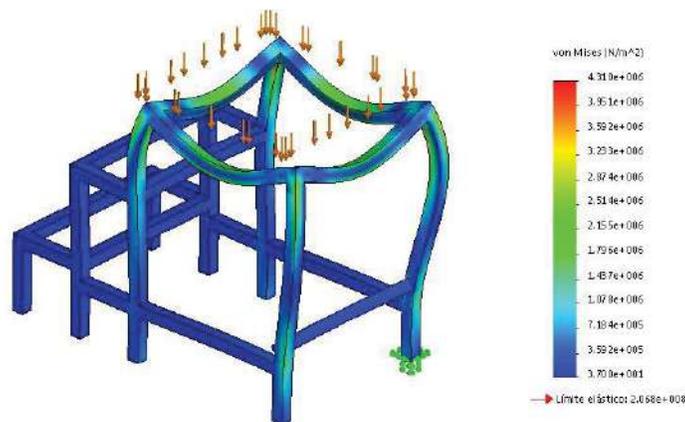


Figura 22. Análisis de tensiones de la base. [SolidWorks]

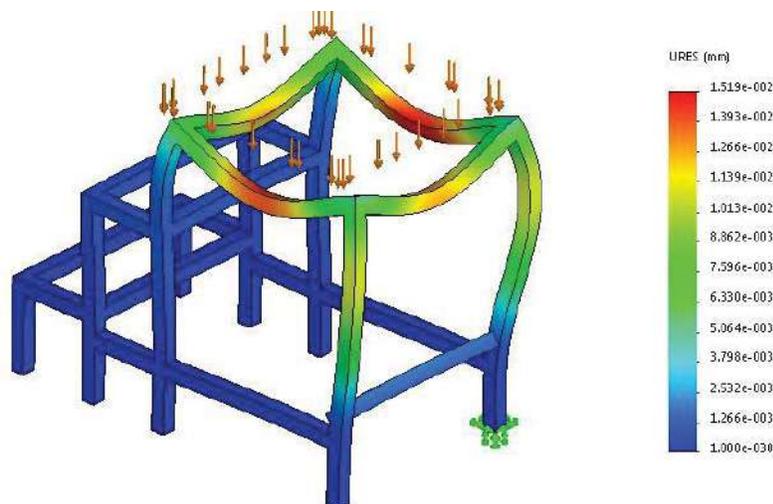


Figura 23. Análisis de desplazamiento. [SolidWorks]

3.5 Diseño del sistema de captura de imagen

Para emitir una valoración del pie la misma que sea confiable, repetible, y no ser invasivo, el paciente deberá estar sobre la superficie refractante y sobre esta se realizara la captura de la imagen, ya que de esta acción dependerá una buena lectura de los pies, es por ello que nos valemos del principio físico descrito por Betts, mediante la relación presión – intensidad con la ayuda de un podoscopio.

Como se requiere un análisis estático de las presiones plantares para hacer una valoración de este y determinar alguna patología, que luego con el procesamiento morfológico de imágenes, se llegara a un diagnostico confiable. El equipo deberá llevar en el interior de su estructura un sistema de captura de imagen la misma que constara con los siguientes elementos:

- Cámara (Kinect).
- Espejo.

3.3.3 Proceso físico

Para el presente proceso físico nos valemos de la construcción del dispositivo, en donde el análisis plantar se centra exclusivamente en la zona de pisada que es refractada, en donde dicho dispositivo mide la distribución de la presión de cada zona de la planta del pie, mediante la relación presión – intensidad, principio fundamental para el análisis plantar como el que maneja el podoscopio.

El proceso físico aplicado en nuestro dispositivo se basa en el principio del podoscopio, el cual consiste en una placa transparente (acrílico, vidrio), iluminadas por luces fluorescentes que canaliza la luz a través de ellas y que en la parte superior se encuentra un material transductor, en el cual se aplica un fenómeno físico descrito por Betts.

3.3.7 Proceso de refracción en las imágenes.

Betts nos dice: la luz entra por los bordes de la placa acrílica es reflejada totalmente entre la superficie superior en inferior de dicha placa. La reflexión total solo ocurre cuando lo rayos de luz viajan desde un medio con un índice de refracción bajo, como en este caso del acrílico al aire. Sin embargo, en aquellas zonas donde el plástico está en contacto íntimo con el acrílico, la reflexión total interna no ocurre, ya que el plástico tiene un índice de refracción mayor que la del acrílico. En estas zonas de contacto los rayos de luz son refractados fuera del acrílico y dispersados en todas las direcciones desde la superficie del plástico [7].

En la siguiente figura se describe:

- μ_a : índice de refracción del aire.
- μ_{ac} : índice de refracción del acrílico.

- μ_p : índice de refracción plástico.
- θ_i : ángulo de incidencia.
- θ_c : ángulo crítico.

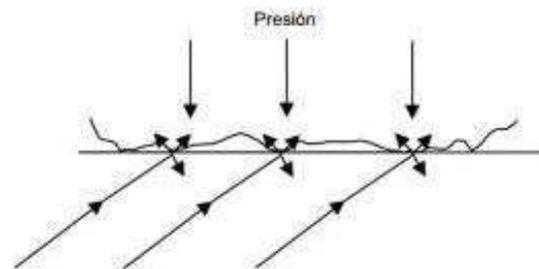
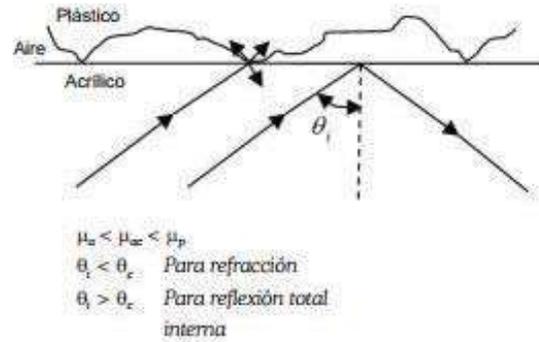


Figura 24. Representación de los principios físicos. [7]

Luego de ver el proceso físico descrito por betts, podemos decir que nuestro equipo se basará en el mismo principio con una particularidad que no se utilizará el plástico sino que más bien el proceso físico se dará directamente entre la planta del pie y el material refractante, para lograr el resultado final, la impresión plantar requerida para el tratamiento mediante software.

3.3.6 Kinect

Como nuestro sistema requiere de una imagen RGB para poder separar la luz verde reflejada y una resolución de 640x480 Mega píxeles para realizar un procesamiento más rápido de las imágenes, se ha decidido usar la cámara de Kinect la cual cumple con estas características.



Figura 25. Kinect componentes. [9]

3.2.4 Espejo

La cámara del Kinect, tiene una particularidad ya que la misma nos debe brindar el enfoque hacia el espejo el mismo que estará fijado a la base y a la pared lateral estará colocado exactamente a 45°, de tal manera que abarque toda la superficie de la zona de pisada.

3.2.5 Vidrio

Para el presente proyecto lo que se empleó como superficie y material refractante es el vidrio claro de 10 mm, el mismo que nos brinda la seguridad y confiabilidad por su dureza y resistencia.

El uso de este material refractante nos brinda excelentes resultados para lograr la refracción total interna frustrada, que para dicho efecto el vidrio al estar sometido a la luz fluorescente a los costados, y así de esta manera lograr el fenómeno de la refracción de la luz a través de la superficie del vidrio.

En la figura 19 se muestra el material refractante el mismo que es de material de vidrio claro templado de 10 mm, se determina la Zona de pisada que es la zona de registro de la superficie plantar, de igual manera tenemos a la zona auxiliar la misma que ayuda al paso de la luz entre las pared superior e inferior de la placa del vidrio contribuyendo así al proceso físico de reflexión total interna. La fuerza de ruptura por flexión de un cristal normal es de 3.5 Kg/cm^2



Figura 26. Resistencia a la flexión. [7]

3.3 Conexión Kinect Matlab.

En las primeras versiones de Kinect o era necesario el adaptador con eliminador debido a que la consola tenía un puerto USB especial que suministraba la potencia requerida para hacer funcionar el Kinect, pero en versiones posteriores fue cambiado el plug de conexión al Xbox 360 debido a esto era imposible conectar el Kinect al ordenador, por esto fue necesario la creación de un adaptador Figura 24 el cual provee de el voltaje necesario para realizar la correcta conexión entre Kinect y PC.



Figura 27. Adaptador usb Kinect. [9]

Para que el ordenador pueda recibir datos de Kinect es necesario instalar Kinect SDK el cual contiene los drivers necesarios para poder usar Kinect en el pc, es importante aclarar que para usar Kinect para Xbox 360 es necesario Kinect SDK v1.8 y para Kinect para Xbox one es necesario Kinect SDK v2.0.

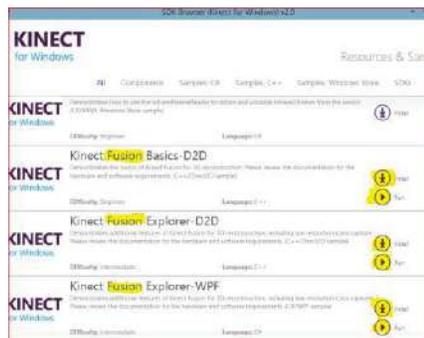


Figura 28. Drivers SDK v1.8. [9]

En Matlab es necesario descargar el toolbox de Kinect for Windows v1 esta versión es necesaria para usar el Kinect para Xbox 360 y solo se admite hasta MATLAB 2015 si se quisiera usar el Kinect para Xbox one es necesario la versión Kinect for Windows v2 y MATLAB 2016 o posterior con Windows 8 o posterior.

Para el procesamiento de imagen es necesario instalar en Matlab Image Acquisition Toolbox el cual contiene todas las funciones necesarias para realizar el procesamiento de imagen tales como medición, descomposición por colores, imagen en binario etc. Para el proyecto que a continuación se describe se usaron las siguientes condiciones:

- Windows 7 Ultimate 64 bit.
- Kinect para Xbox 360.
- Matlab 2015a.
- Kinect SDK v1.8. (Windows)
- Kinect for Windows v1 Toolbox. (MATLAB)
- Image Acquisition Toolbox. (MATLAB)

3.4 Iluminación

La iluminación existe en el entorno, no siempre es la adecuada ya que se obtienen imágenes con bajo contraste, es decir, poco o demasiado iluminadas. Esto conlleva a problemas con sombras no deseadas, que afectan directamente la complejidad del algoritmo. Para definir el definir claramente el tipo de iluminación es necesario identificar las propiedades de los objetos que intervienen en la escena, pueden ocurrir tres situaciones cuando un haz luminoso incide sobre un material; que se refleje, se absorba o se transmita a través de él. En nuestro caso utilizaremos el caso tres queremos que se transmita el haz de luz en el vidrio, para cuando se haga presión sobre el vidrio se marque solo la huella plantar en este caso utilizaremos una tira de Led de color verde.



Figura 29. Tira de LED. [Fuente autores]

3.5 Desarrollo de la interfaz

Se procede a crear una Interfaz gráfica en Matlab que permita la interacción del usuario con el Kinect al iniciar se tomara la captura de las huellas del paciente. En la Figura 27 se puede observar la interfaz de usuario realizando las primeras capturas en las siguientes imágenes se describe a fondo cada parte de la interfaz.

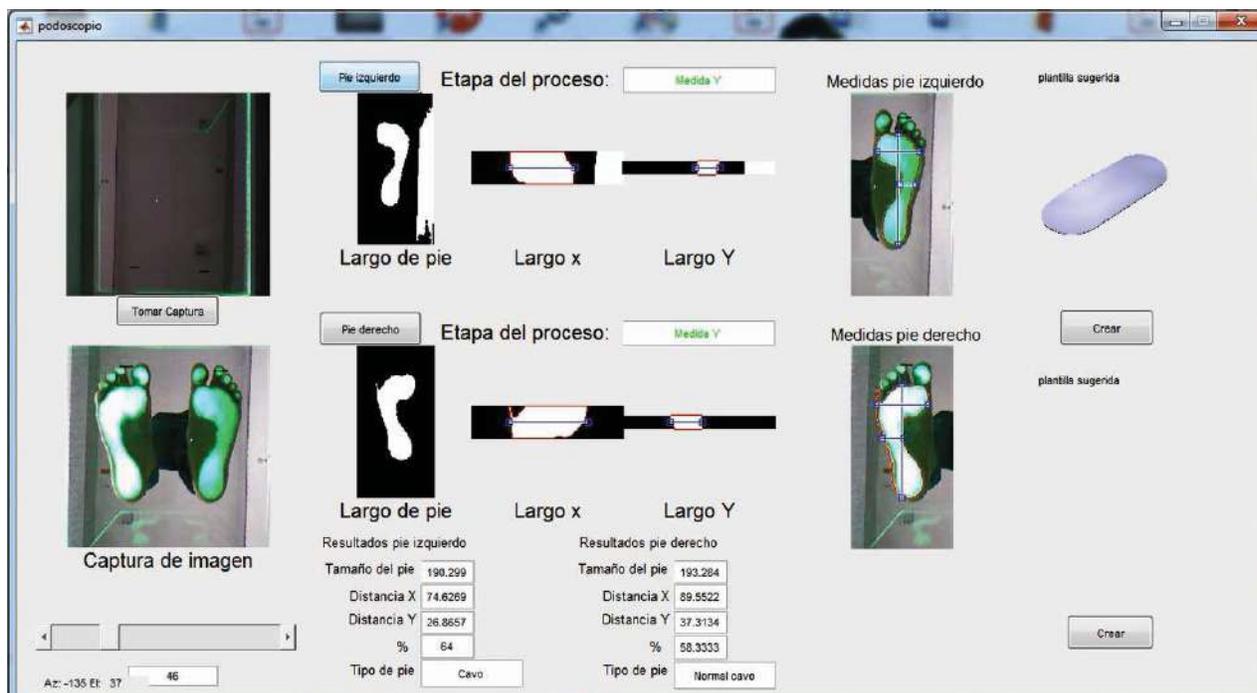


Figura 30. Interfaz gráfica de usuario. [Fuente autores]

3.6 Descripción de la Interfaz gráfica de usuario

Como primer paso se procede a tomar una captura de ambos pies, el paciente se debe colocar por encima del podoscopio colocando los pies en medio del cristal y ligeramente separados, el paciente debe tener una postura recta en todo momento, entonces se procede a tomar la captura esto se realiza por medio de una interfaz creada en MATLAB. Se tiene que asignar al Kinect como una entrada de video y posteriormente tomar la captura a continuación se muestra el código utilizado para esta tarea, si se han instalado los controladores correctamente no se debería tener mayor problema:

```
//asigna una entrada de video.  
Video = videoinput('Kinect',1,'RGB_640x480');  
//toma una captura del video.  
CapturaOriginal = getsnapshot (Video);
```



Figura 31. Toma de captura. [Fuente autores]

3.6.1 Modo espejo

Originalmente Kinect tiene una cámara RGB estándar que a diferencia con la perspectiva de visión de una persona, en Kinect toma una captura del video como espejo, esto ocurre con todas las cámaras de cualquier dispositivo, pero se cambian automáticamente, en Matlab no es así por lo que es necesario la creación de una imagen en modo espejo esto para relacionarse mejor con la imagen en la pantalla. Para esto se usa el siguiente código:

```
//modo espejo  
ImagenEspejo = flipdim(CapturaOriginal,2);
```

La intención de esto es la de tener una perspectiva más clara de la imagen que será procesada debido a que se puede confundir entre el pie derecho o izquierdo y con esto se tiene una imagen como si se estuviera viendo frontalmente.



Figura 32. Modo espejo. [Fuente autores]

3.6.2 Recorte de imagen

A continuación, se recortará la imagen con el propósito de crear siete imágenes que serán procesadas de manera independiente. El primer recorte prepara la captura de video para ser procesada, eliminando el contorno, en esta captura el usuario identificará la correcta posición de los pies del paciente y la posición del Kinect, no deberá poder observarse ninguna luz en el contorno de la imagen.

Después se recorta la imagen anterior la cual contiene ambos pies, para separarla en pie derecho y pie izquierdo Figuras 30,31 esto con el propósito de medir individualmente la Longitud del pie a nivel de la línea axial. Con esto se generan 2 imágenes. Los siguientes recortes se realizan en el ante pie de ambas imágenes pie derecho y pie izquierdo con el propósito de medir la dista X en cada una de ellas, que será usada en la ecuación de Hernández corvo. Esto genera 2 imágenes.

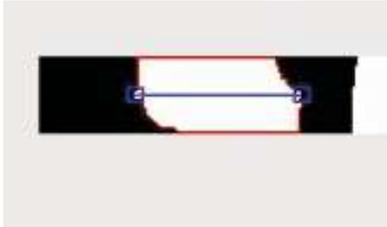


Figura 34. Distancia x (Pie izquierdo) [Fuente autores]



Figura 33. Distancia x (Pie derecho) [Fuente autores]

Los últimos recortes se realizan en el pie medio de las imágenes pie derecho y pie izquierdo con el propósito de medir la distancia Y que será usada en la ecuación de Hernández corvo. Esto genera 2 imágenes.

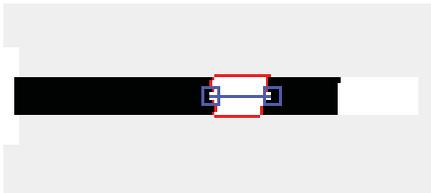


Figura 36. Distancia y (Pie izquierdo) [Fuente autores]

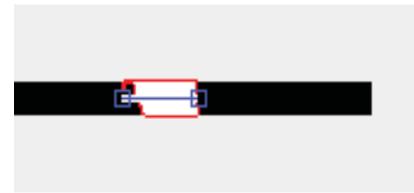


Figura 35. Distancia y (Pie derecho) [Fuente autores]

3.6.3 División de color

El podoscopio utiliza un cristal de 10 mm de espesor para soportar el peso de una persona promedio con iluminación verde en uno de sus lados, el objetivo de esto es iluminar la huella del pie (la piel que toca la superficie) en luz verde, a continuación se obtiene los tres colores RGB de cada una de las seis imágenes previamente cortadas, se usa el siguiente código:

```
redChannel = rgbImage(:, :, 1); //extrae el Rojo.
greenChannel = rgbImage(:, :, 2); //extrae el Verde.
blueChannel = rgbImage(:, :, 3); //extrae el Azul.
```



Figura 37. Extracción de color. [Fuente autores]

Al extraer el color verde de cada imagen se genera una nueva imagen en la cual los pixeles verdes adquieren una escala de grises más clara y los pixeles rojos y azules adquieren una escala de grises más oscuro esto hace resaltar la huella en un color blanco y el fondo negro cuando se pasa a una imagen binaria, pero la detección de formas de MATLAB solo detecta colores negros, entonces las imágenes generadas se deben cambiar a negativo para que la huella adquiera un tono negro y el fondo un tono blanco el código es el siguiente:

```
negativo=255-greenChannel;
```

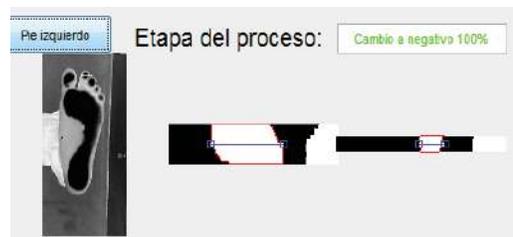


Figura 38. Imagen negativa. [Fuente autores]

3.6.4 Convertir imagen en binario

Una vez teniendo las seis imágenes en negativo, y teniendo en escala de grises oscura la huella y escala de grises clara el fondo se procede a transformar las seis imágenes nuevas a una imagen binaria la cual crea pixeles blancos y negros a partir de una imagen en escala de grises el nivel en el que se desea que los pixeles se vuelvan blancos o negros se puede ajustar con el siguiente código:

```
binaryImage = negativo < x;//ajuste.
```

Donde x es el valor en el que se desea que los pixeles adquieran un valor de 0 ò 1 para ambientes donde existe demasiada luz el valor debe cambiarse a un valor superior a 150 y en ambientes de poca luz se debe cambiar a un valor de menos de 150.



Figura 39. Imagen en binario. [Fuente autores]

3.6.5 Relleno de huecos

Como se puede esperar de una fotografía no siempre se toma una captura perfecta, debido al ambiente pueden existir pixeles que varían de color en toda la imagen y hasta este punto pueden existir zonas más claras u oscuras dentro y fuera de la huella, haciendo así un difícil procesamiento de la imagen esto se resuelve rellenando los agujeros en la imagen binaria en el área de la huella, Matlab cuenta con la función `imfill` que resuelve de manera automática este detalle rellenando con pixeles negros los pixeles blancos menos significativos dentro de la huella creando así un área de pixeles negros uniforme.



Figura 40. Relleno de pixeles [Fuente autores]

3.6.6 Remover fondo de la imagen.

Lo ideal sería que la luz verde solo fuera incidente en la huella del pie, la realidad es que siempre se escapa un poco de luz, de esta manera son iluminados objetos no deseados por ejemplo el arco plantar en el pie, el pantalón, o manchas en el cristal esto debido a sudoración o polvo en los pies los cuales manchan el cristal y la luz verde en consecuencia los ilumina.

Por esto es necesario eliminar el fondo en las imágenes binarias para este caso todos los pixeles negros fuera de la huella, Matlab posee la función `bwareaopen`, la cual elimina el fondo dependiendo de los pixeles conectados (área en pixeles) la cantidad de pixeles se puede modificar en el código.



Figura 41. Eliminación del fondo. [Fuente autores]

3.6.7 Mediciones en la imagen.

El Kinect siempre estará a una determinada distancia del área que se quiere medir por lo que es posible crear una ecuación que determine la distancia en mm por cada pixel de la cámara. La resolución de la cámara de Kinect es de 640x480 pixeles y la distancia entre el área que se desea medir y la cámara del Kinect es de 300mm, para la calibración de la distancia se usa una banda verde de 100mm de largo y se toma una captura desde la interfaz, el programa medirá automáticamente la distancia en pixeles, con esto es posible realizar una relación la cual se describe a continuación:

$$dpm = \left(\frac{dpb}{b}\right) * 1 \quad (2)$$

donde:

b = distancia de la banda (para este caso 100mm).

d_{pb} = distancia en pixeles de la banda b .

d_{pm} = distancia en pixeles por mm.

Con esto se obtiene la distancia en pixeles por milímetro, la cual será usada para realizar las medidas correspondientes. Ahora que las seis imágenes están perfectamente delimitadas se procede a realizar las medidas en pixeles correspondientes Matlab posee la función `regionprops` la cual regresa medidas como área, centroide, enmarcación de figuras, longitud máxima, longitud mínima entre otras.

La función usada para realizar la medida axial de longitud del pie es `MajorAxisLength` esta función mide la longitud más larga en una elipse esta herramienta es ideal para realizar esta medición debido a que se puede considerar al pie como una elipse y dependiendo del tamaño calculado en esta etapa se decide dónde cortar las imágenes del pie medio y ante pie.

Para realizar la medición de las distancia X y Y, solicitadas en la ecuación de Hernández corvo se usa la función `BoundingBox` la cual encuadra a una figura y devuelve la posición de un punto de origen en coordenadas (x,y) siendo el punto cero de la imagen la esquina superior izquierda; seguida por la anchura y altura del cuadro y las devuelve en una matriz de 1x4. Tomando solo la medida del ancho en cada imagen es posible determinar las distancias X y Y solicitadas, todas las medidas obtenidas anteriormente están dadas en pixeles entonces se procede a pasarlos a milímetros usando la relación anterior se obtuvo la distancia en pixeles por milímetro con esto se puede formular las siguientes relaciones:

$$dm = \frac{dpx}{dpm} , \quad (3)$$

$$dm = \frac{dpy}{dpm} , \quad (4)$$

donde:

dm = distancia en milímetros.

dpx = distancia en pixeles de X.

dpy = distancia en pixeles de Y.

3.6.8 Resultados de tipo de pie.

Se muestran los resultados obtenidos en la interfaz con los siguientes datos para cada pie valor en X, valor en Y, distancia axial de la longitud del pie, porcentaje obtenido con la ecuación de Hernández corvo y el tipo de pie a partir de este porcentaje.

Resultados pie izquierdo		Resultados pie derecho	
Tamaño del pie	190.299	Tamaño del pie	193.284
Distancia X	73.8806	Distancia X	87.3134
Distancia Y	26.1194	Distancia Y	37.3134
%	64.6465	%	57.265
Tipo de pie	Cavo	Tipo de pie	Normal cavo

Figura 42. Resultados de las mediciones. [Fuente autores]

3.7.9 Modelado de plantilla.

Por ultimo una vez obtenido el tipo de pie mediante el procesamiento de imagen y utilizando el método de Hernández-Corvo nuestra interfaz nos proporciona una sugerencia de la plantilla que deberíamos de usar en la Figura 40 se puede mostrar el modelado de la plantilla para posteriormente poder imprimirla en 3D.



Figura 43. Plantilla sugerida. [Fuente autores]

4.1 Pruebas del hardware y software

Como proceso de pruebas lo que se realiza es el diagnóstico del equipo frente a las exigencias de funcionamiento, las mismas que es soportar el peso de las personas que lo requieran el análisis plantar y que dicha prueba sea la más amigable con el paciente. A continuación, se presenta el prototipo final que de acuerdo al análisis inicial propuesto cumple con las exigencias del cual obtenemos los resultados esperados, los mismos que la imagen plantar del paciente.

Luego de tener el equipo listo, procedemos a indicar como se prepara la escena para el proceso de captura de la imagen, la misma que a la superficie refractante se le ilumina mediante una tira de led que se encuentra a un extremo y se logra la refracción total de la superficie plantar frente al vidrio.

Una vez que se ha realizado la colocación de todos los elementos en el equipo como son vidrio, espejo, Kinect y tira de led procedemos a verificar el funcionamiento y la colocación inicial según corresponda para garantizar la funcionalidad del equipo, en la Figura 41 se indica el paciente listo a ser analizado la superficie plantar.

4.1.1 Paciente 1

Nombre: Marco Antonio González Saucedo

Peso: 64 Kg

Estatura: 1.60m

Ocupación: Estudiante

Edad: 23 años



Figura 44.
Ubicación del
paciente.
[Fuente autores]

Una vez preparado el hardware se conecta al computador, y se inicia el programa podoscopio.m, el mismo que es extensión de Matlab 2015a, ahora elegimos la opción tomar captura, en la siguiente Figura 42 vamos a observar la imagen plantar del paciente.

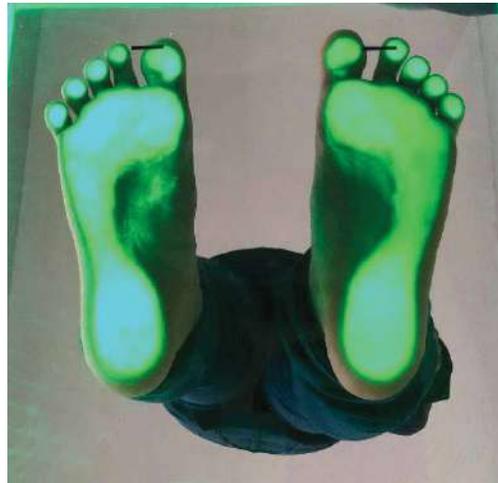


Figura 45. Forma en la que se ubican los pies. [Fuente autores]

Teniendo la imagen correcta en la posición correcta, procedemos a realizar el procesamiento de imagen, como se muestra en la Figura 43, se puede observar que el paciente tiene en el pie izquierdo un tipo de pie cavo y el pie derecho es normal cavo, también se puede observar la sugerencia de la plantilla.

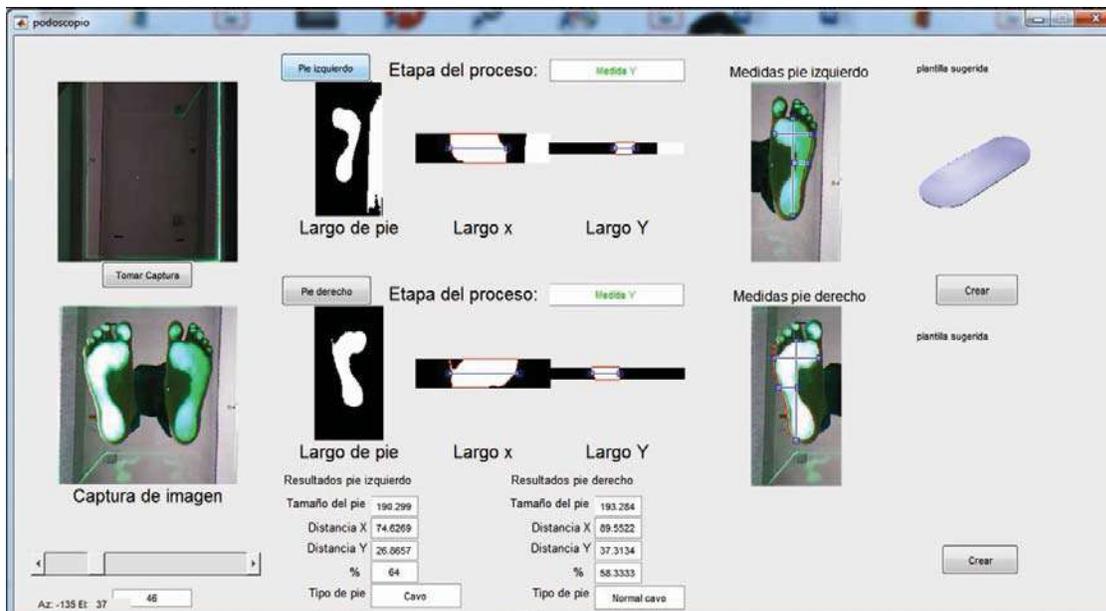


Figura 46. Procesamiento de la huella plantar. [Fuente autores]

4.1.2 Paciente 2

Nombre: Fernando Flores Clara

Peso: 75 Kg

Estatura: 1.65m

Ocupación: Estudiante

Edad: 23 años

En la Figura 44 se muestra los resultados obtenidos del segundo paciente como se puede observar en este caso los dos pies presentan pie plano ya que en el caso anterior se observa lo contrario.

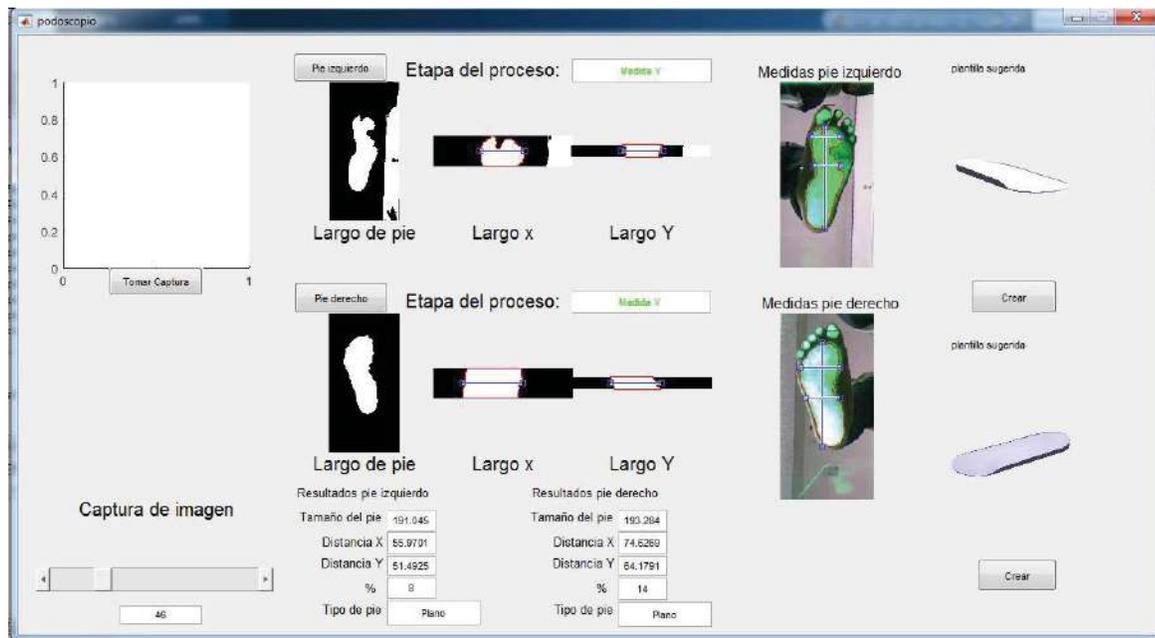


Figura 47. Resultados del paciente 2. [Fuente autores]

4.2 Conclusiones

Se ha logrado con éxito el diseño e implementación de un podoscopio digital que de manera automática valora el tipo de pie de un paciente en seis tipos de deformación del pie facilitando el trabajo del ortopedista.

Durante la investigación de los diferentes métodos de clasificación del pie, se ha seleccionado el método de Hernández Corvo el cual ha resultado ser ideal para la creación del algoritmo de análisis de la huella plantar por su método gráfico basado en líneas rectas y tangentes.

Se ha logrado un diseño de un podoscopio no invasivo al paciente el cual genera una confianza en el paciente, aunque para trabajos futuros se espera cambiar el vidrio usado por vidrio templado con el motivo de soportar mayores cargas.

Se ha creado una interfaz gráfica amable e intuitiva al usuario lo que genera que el trabajo de análisis de la huella plantar sea más fácil de visualizar.

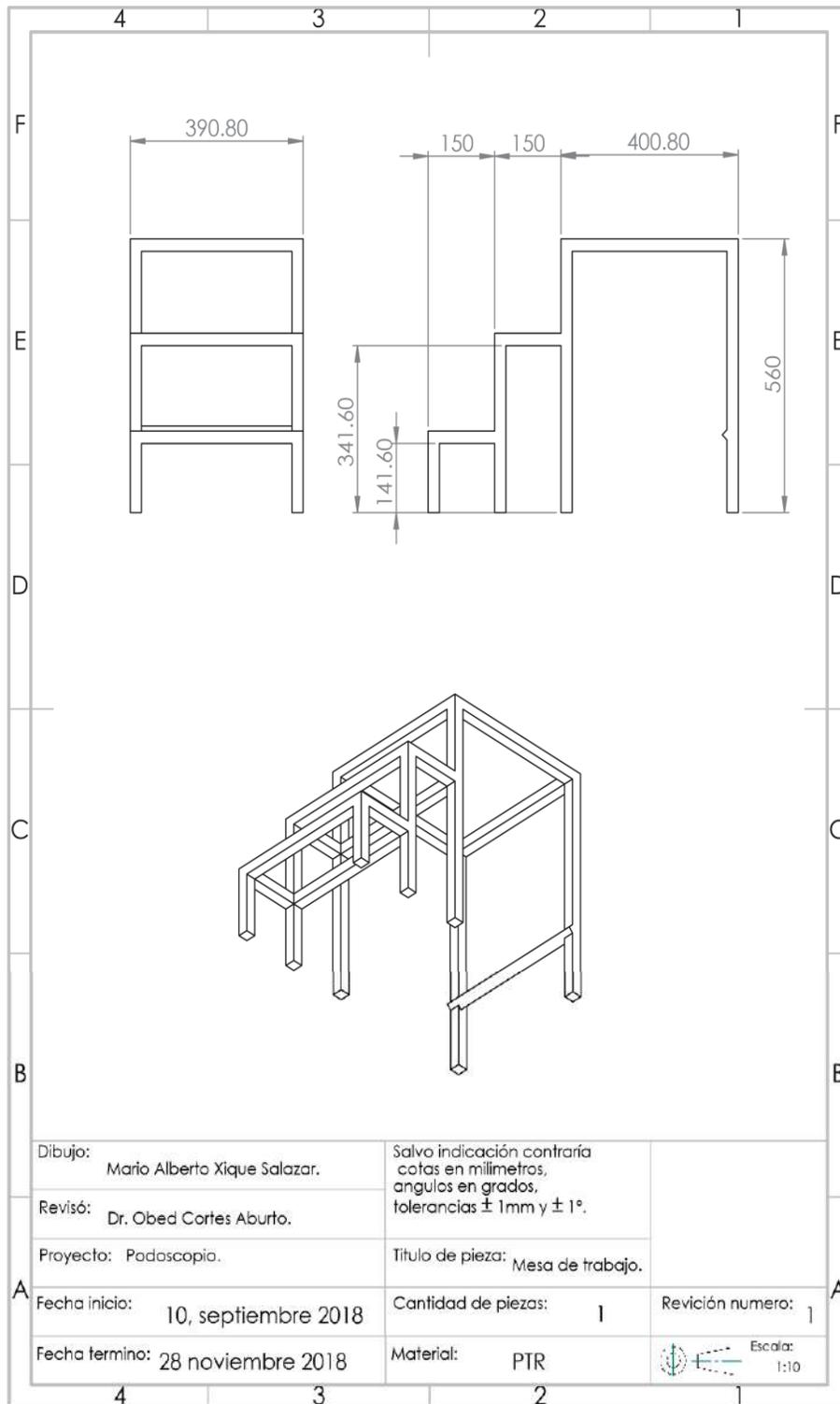
Se ha creado un algoritmo de reconocimiento de la huella plantar en la que es posible obtener datos como largo del pie, distancia en X y distancia en Y datos útiles para la clasificación del tipo de pie usados en la ecuación de Hernández Corvo, así también genera un archivo STL de una plantilla ideal para cada tipo de pie lista para imprimir por medio de la tecnología de impresión 3D.

Referencias

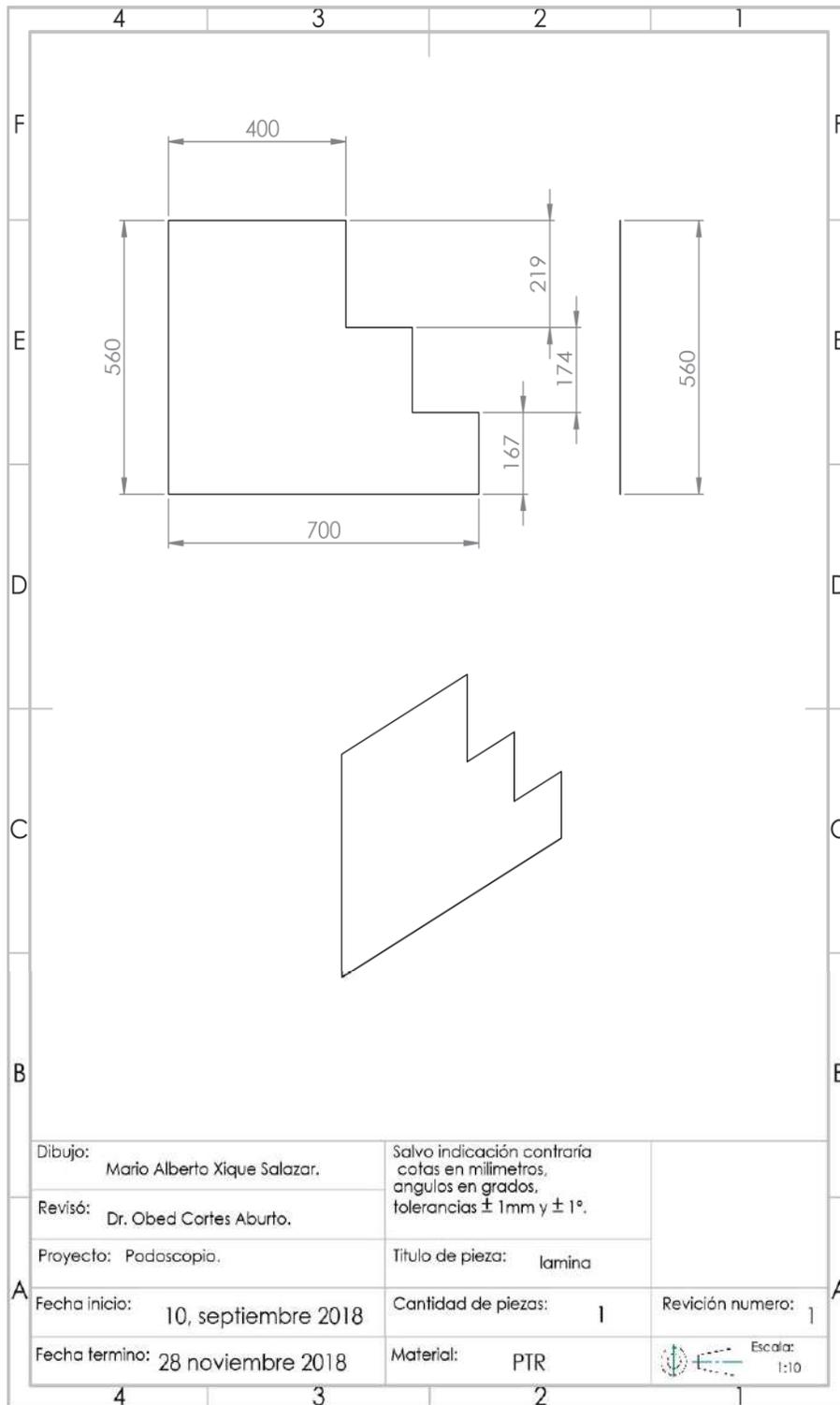
- [1] Morfología de manos y pies, *Descripción del pie*, Febrero 2007, pág. 11.
<http://www.mcgraw-hill.es/bcv/guide/capitulo/8448180747.pdf>
- [2] R. Hernández Corvo, Temas de estabilografía. Comunidad de Madrid: 2007
- [3] Baropodometría electrónica, Descripción del equipo, Enero 2007
<http://www.bioingenieria.edu.ar/grupos/geic/biblioteca/archivos/Trabypres/T08TCPe07.pdf>
- [4] Anatomía de la Pierna Humana, Capítulo 2, Octubre 2011, pág. 13.
http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/hernandez_s_f/capitulo2.pdf
- [5] C. García, “Pie plano flexible” 2012. [Online]. Available:
www.traumatologiainfantil.com/es/pie/pies_planos. [Accessed: 15-Dec-2015].
- [6] R. Hernández Corvo, Morfología funcional deportiva: sistema locomotor. Madrid: Paidotribo, 1989.
- [7] Betts, R. P.; Duckworth, T. and Austin, L. G. (1980). Critical light reflection at a plastic/glass interface and its application to foot pressure measurements, *Medical and Biological Engineering and Computing*, Vol. 4, pt 3, p. 136-142.
- [8] “Síntomas y tratamiento del pie cavo” [Online]. Available:
<http://runningtherapycenter.com/sintomas-y-tratamiento-del-pie-cavo/>.
- [9] “Centro de descargas Microsoft” [Online]. Available: <https://www.microsoft.com/en-us/download/details.aspx?id=40278>
- [10] Carreño J. C. Y Cabrera P. T. “Diseño e implementación de un sistema de análisis de las presiones plantares en estática basado en procesamiento de imágenes” 2014.
- [11] Hernández L. “Podoscopio Digital” 2014. [Online]. Available:
http://rd.udb.edu.sv:8080/jspui/bitstream/11715/940/1/podoscopio_digital.pdf
- [12] Vargas E. D. “detección de deformaciones provocadas por el pie diabético” 2017.
[Online]. Available:
http://rd.udb.edu.sv:8080/jspui/bitstream/11715/940/1/podoscopio_digital.pdf

Anexos

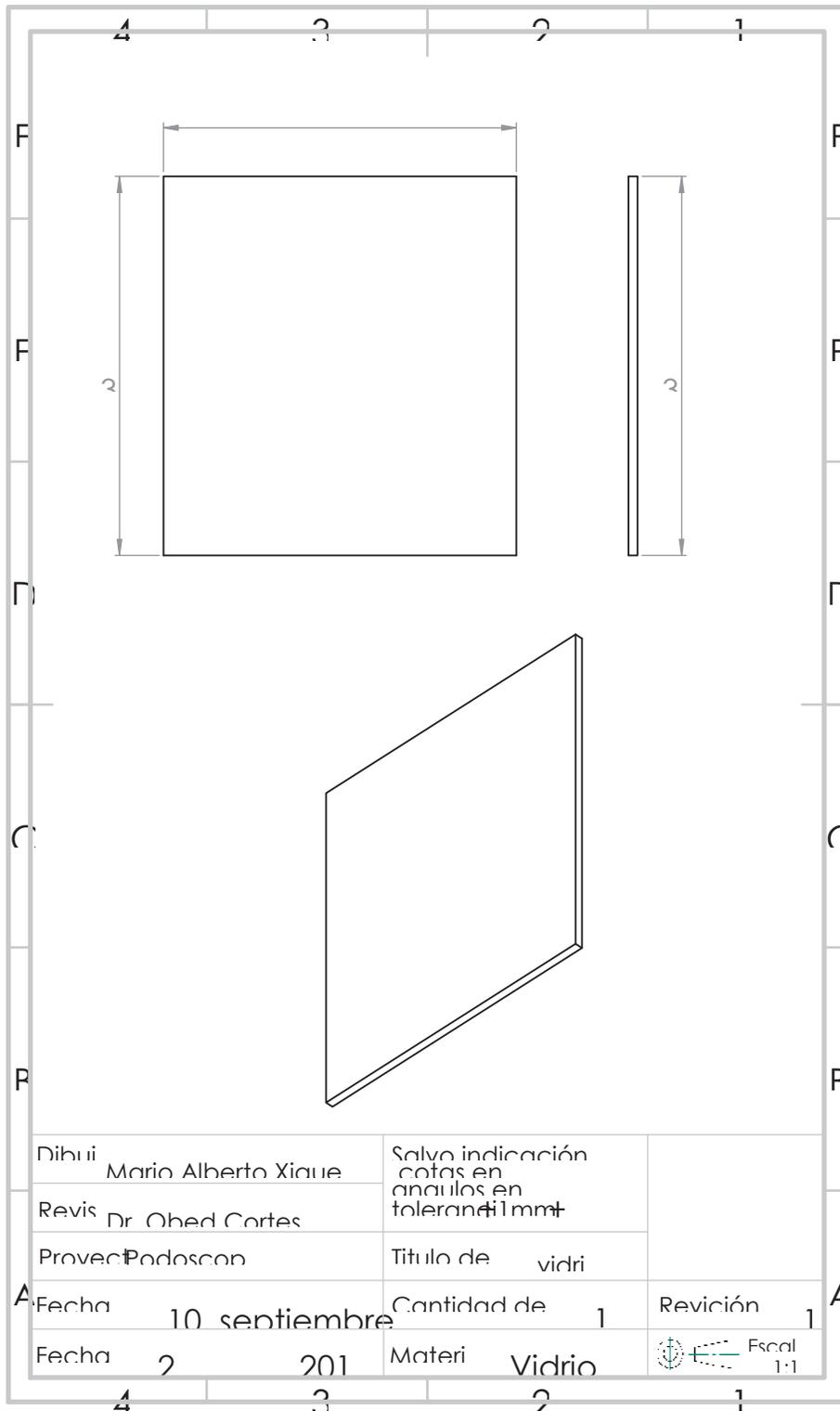
1. Plano de la estructura



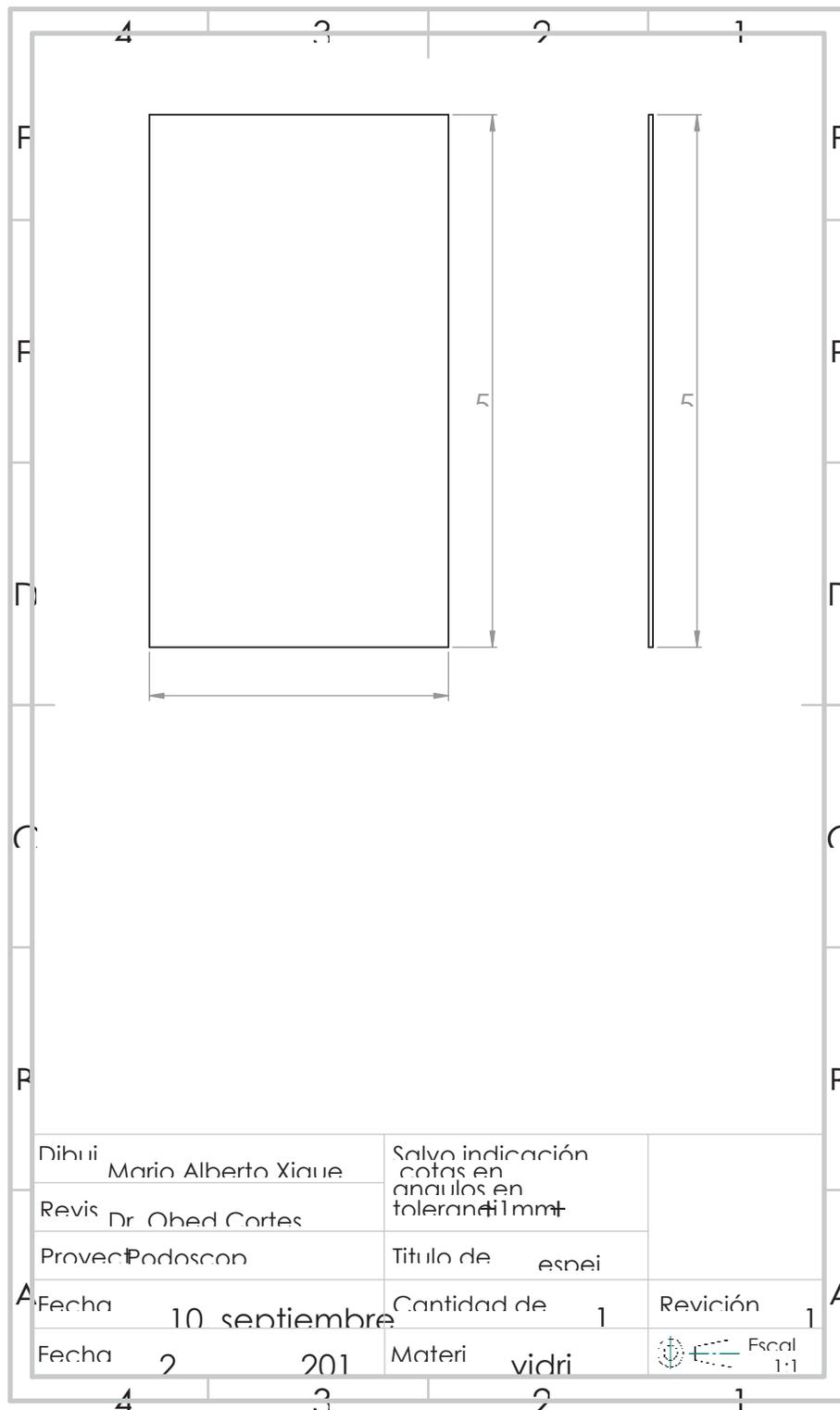
2. Lamina



3. Vidrio



4. Espejo



5. Código Matlab

```
function varargout = podoscopio(varargin)
% PODOSCOPIO MATLAB code for podoscopio.fig
%   PODOSCOPIO, by itself, creates a new PODOSCOPIO or raises the existing
%   singleton*.
%
%   H = PODOSCOPIO returns the handle to a new PODOSCOPIO or the handle to
%   the existing singleton*.
%
%   PODOSCOPIO('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in PODOSCOPIO.M with the given input arguments.
%
%   PODOSCOPIO('Property','Value',...) creates a new PODOSCOPIO or raises
the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before podoscopio_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to podoscopio_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help podoscopio

% Last Modified by GUIDE v2.5 28-Nov-2018 22:49:27

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @podoscopio_OpeningFcn, ...
                  'gui_OutputFcn',  @podoscopio_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before podoscopio is made visible.
function podoscopio_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```

% varargin    command line arguments to podoscopio (see VARARGIN)

% Choose default command line output for podoscopio
handles.output = hObject;
axes(handles.graambpre);
axis off
axes(handles.graderlar);
axis off
axes(handles.graderx);
axis off
axes(handles.gradery);
axis off
axes(handles.gradercom);
axis off
axes(handles.graizqlar);
axis off
axes(handles.graizqx);
axis off
axes(handles.graizqy);
axis off
axes(handles.graizqcom);
axis off
axes(handles.graizqpla);
axis off
axes(handles.graderpla);
axis off
handles.video = videoinput('Kinect',1,'RGB_640x480');
    start(handles.video);
    guidata(hObject,handles);
    vidRes = get(handles.video,'VideoResolution');
    nBands = get(handles.video,'NumberOfBands');
    hImage = image(zeros(vidRes(2), vidRes(1),nBands),'Parent',...
        handles.graambvid);
    preview(handles.video,hImage);

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes podoscopio wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = podoscopio_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function texizqy_Callback(hObject, eventdata, handles)
% hObject    handle to texizqy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of texizqy as text
%         str2double(get(hObject,'String')) returns contents of texizqy as a
double

% --- Executes during object creation, after setting all properties.
function texizqy_CreateFcn(hObject, eventdata, handles)
% hObject     handle to texizqy (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function texdereta_Callback(hObject, eventdata, handles)
% hObject     handle to texdereta (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of texdereta as text
%         str2double(get(hObject,'String')) returns contents of texdereta as a
double

% --- Executes during object creation, after setting all properties.
function texdereta_CreateFcn(hObject, eventdata, handles)
% hObject     handle to texdereta (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function texizqeta_Callback(hObject, eventdata, handles)
% hObject     handle to texizqeta (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of texizqeta as text
%         str2double(get(hObject,'String')) returns contents of texizqeta as a
double

```

```

% --- Executes during object creation, after setting all properties.
function texizqeta_CreateFcn(hObject, eventdata, handles)
% hObject    handle to texizqeta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function texizqlar_Callback(hObject, eventdata, handles)
% hObject    handle to texizqlar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of texizqlar as text
%       str2double(get(hObject,'String')) returns contents of texizqlar as a
double

% --- Executes during object creation, after setting all properties.
function texizqlar_CreateFcn(hObject, eventdata, handles)
% hObject    handle to texizqlar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function texizqtip_Callback(hObject, eventdata, handles)
% hObject    handle to texizqtip (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of texizqtip as text
%       str2double(get(hObject,'String')) returns contents of texizqtip as a
double

% --- Executes during object creation, after setting all properties.
function texizqtip_CreateFcn(hObject, eventdata, handles)

```

```

% hObject    handle to texizqtip (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.

```

```

%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function texizqx_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to texizqx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of texizqx as text
%     str2double(get(hObject,'String')) returns contents of texizqx as a
double

```

```

% --- Executes during object creation, after setting all properties.

```

```

function texizqx_CreateFcn(hObject, eventdata, handles)

```

```

% hObject    handle to texizqx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.

```

```

%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function texizqher_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to texizqher (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of texizqher as text
%     str2double(get(hObject,'String')) returns contents of texizqher as a
double

```

```

% --- Executes during object creation, after setting all properties.

```

```

function texizqher_CreateFcn(hObject, eventdata, handles)

```

```

% hObject    handle to texizqher (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.

```

```

%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in botambpre.
function botambpre_Callback(hObject, eventdata, handles)
% hObject     handle to botambpre (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
handles.rgb = getsnapshot(handles.video);
espejoAA = flipdim(handles.rgb,1);
espejoID = flipdim(espejoAA,2);
PieRecorte1 = imcrop(espejoID,[70 10 465 465]);
imwrite(PieRecorte1, 'C:\Users\Fernando\Documents\Podoscopio\PieRecorte1.jpg')
axes(handles.graambpre);
imshow(PieRecorte1);

function texdery_Callback(hObject, eventdata, handles)
% hObject     handle to texdery (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of texdery as text
%       str2double(get(hObject,'String')) returns contents of texdery as a
double

% --- Executes during object creation, after setting all properties.
function texdery_CreateFcn(hObject, eventdata, handles)
% hObject     handle to texdery (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function texderlar_Callback(hObject, eventdata, handles)
% hObject     handle to texderlar (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of texderlar as text
%       str2double(get(hObject,'String')) returns contents of texderlar as a
double

```

```

% --- Executes during object creation, after setting all properties.
function texderlar_CreateFcn(hObject, eventdata, handles)
% hObject    handle to texderlar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function texdertip_Callback(hObject, eventdata, handles)
% hObject    handle to texdertip (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of texdertip as text
%         str2double(get(hObject,'String')) returns contents of texdertip as a
double

% --- Executes during object creation, after setting all properties.
function texdertip_CreateFcn(hObject, eventdata, handles)
% hObject    handle to texdertip (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function texderx_Callback(hObject, eventdata, handles)
% hObject    handle to texderx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of texderx as text
%         str2double(get(hObject,'String')) returns contents of texderx as a
double

% --- Executes during object creation, after setting all properties.
function texderx_CreateFcn(hObject, eventdata, handles)
% hObject    handle to texderx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function texderher_Callback(hObject, eventdata, handles)
% hObject     handle to texderher (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of texderher as text
%     str2double(get(hObject,'String')) returns contents of texderher as a
double

% --- Executes during object creation, after setting all properties.
function texderher_CreateFcn(hObject, eventdata, handles)
% hObject     handle to texderher (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in botderini.
function botderini_Callback(hObject, eventdata, handles)
% hObject     handle to botderini (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
t=0.5;
barra = get(handles.barra,'Value')
barral = barra*40;
set(handles.texambbar,'string', num2str(barral))
set(handles.texdereta,'ForegroundColor','red')
set(handles.texdereta,'string','Imagen RGB 30%')
fullFileName = imread('fer.jpg');
pieDerecho = imcrop(fullFileName,[10 5 232 465]);
axes(handles.graderlar);
imshow(pieDerecho);
pause(t);
set(handles.texdereta,'string','Imagen RGB 60%')
pause(t);
set(handles.texdereta,'ForegroundColor','green')
set(handles.texdereta,'string','Imagen RGB 100%')
pause(t);

```

```

set(handles.texdereta,'ForegroundColor','red')
set(handles.texdereta,'string','Extraccion color verde 30%')
redChannel = pieDerecho(:, :, 1);
greenChannel = pieDerecho(:, :, 2);
blueChannel = pieDerecho(:, :, 3);
axes(handles.graderlar);
imshow(greenChannel);
pause(t);
set(handles.texdereta,'string','Extraccion color verde 60%')
pause(t);
set(handles.texdereta,'ForegroundColor','green')
set(handles.texdereta,'string','Extraccion color verde 100%')
pause(t);

```

```

set(handles.texdereta,'ForegroundColor','red')
set(handles.texdereta,'string','Cambio a negativo 30%')
negativo=255-greenChannel;
axes(handles.graderlar);
imshow(negativo);
pause(t);
set(handles.texdereta,'string','Cambio a negativo 60%')
pause(t);
set(handles.texdereta,'ForegroundColor','green')
set(handles.texdereta,'string','Cambio a negativo 100%')
pause(t);

```

```

set(handles.texdereta,'ForegroundColor','red')
set(handles.texdereta,'string','Cambio a Binario 33%')
pause(t);
binaryImage = negativo < barra1;
axes(handles.graderlar);
imshow(binaryImage);
pause(t);
set(handles.texdereta,'string','Cambio a Binario 45%')
pause(t);
set(handles.texdereta,'ForegroundColor','green')
set(handles.texdereta,'string','Cambio a Binario 100%')
pause(t);

```

```

set(handles.texdereta,'ForegroundColor','red')
set(handles.texdereta,'string','rellenando huecos 43%')
pause(t);
binaryImage = imfill(binaryImage, 'holes');
axes(handles.graderlar);
imshow(binaryImage);
set(handles.texdereta,'string','rellenando huecos 70%')
pause(t);
set(handles.texdereta,'ForegroundColor','green')
set(handles.texdereta,'string','rellenando huecos 100%')
pause(t);

```

```

set(handles.texdereta,'ForegroundColor','red')
set(handles.texdereta,'string','eliminando fondo 12%')
pause(t);
binaryImage = bwareaopen(binaryImage, 5000);
binaryImage = imclose(binaryImage, true(5));

```

```

axes(handles.graderlar);
imshow(binaryImage);
set(handles.texdereta,'string','eliminando fondo 30%')
pause(t);
set(handles.texdereta,'ForegroundColor','green')
set(handles.texdereta,'string','eliminando fondo 100%')
pause(t);

Box = regionprops(binaryImage,'BoundingBox');
vect1=Box.BoundingBox;
corx=vect1(1);
cory=vect1(2);
anchox=vect1(3);
anchoy=vect1(4);
boundaries = bwboundaries(binaryImage);
x = boundaries{1}(:, 2);
y = boundaries{1}(:, 1);
axes(handles.gradercom);
imshow(pieDerecho);
hold on;
plot(x, y, 'r-', 'LineWidth', 1);
d=imdistanline(gca,[corx+anchox/2 corx+anchox/2],[cory cory+anchoy]);
setLabelVisible(d,false);
regla = (getDistance(d))/1.34;
set(handles.texderlar,'string',regla);
pause(t);
%-----
    if regla >= 195 && regla <= 230
        disx = imcrop(binaryImage,[0 120 232 50]);
        disy = imcrop(binaryImage,[0 250 232 20]);
        lx=105;
        ly=250;
    end
    if regla >= 150 && regla <= 194
        disx = imcrop(binaryImage,[0 115 232 50]);
        disy = imcrop(binaryImage,[0 200 232 20]);
        lx=110;
        ly=200;
    end
    if regla >= 110 && regla <= 149
        disx = imcrop(binaryImage,[0 110 232 50]);
        disy = imcrop(binaryImage,[0 150 232 20]);
        lx=100;
        ly=150;
    end
set(handles.texdereta,'string','Medida X')
Boxx = regionprops(disx,'BoundingBox');
vecx=Boxx.BoundingBox;
corxx=vecx(1);
coryx=vecx(2);
anchoxx=vecx(3);
anchoyx=vecx(4);
boundariesx = bwboundaries(disx);
x = boundariesx{1}(:, 2);
y = boundariesx{1}(:, 1);
axes(handles.graderx);
imshow(disx);

```

```

hold on;
plot(x, y, 'r', 'LineWidth', 1);
dx=imdistline(gca,[corxx corxx+anchoox],[coryx+anchoyx/2 coryx+anchoyx/2]);
setLabelVisible(dx,false);
reglax = getDistance(dx)/1.34;
set(handles.texderx,'string',reglax)

pause(t);

set(handles.texdereta,'string','Medida Y')
Boxy = regionprops(disY,'BoundingBox');
vecy=Boxy.BoundingBox;
corxy=vecy(1);
coryy=vecy(2);
anchoxy=vecy(3);
anchoyy=vecy(4);
boundariesy = bwboundaries(disY);
x = boundariesy{1}(:, 2);
y = boundariesy{1}(:, 1);
axes(handles.gradery);
imshow(disY);
hold on;
plot(x, y, 'r', 'LineWidth', 1);
dy=imdistline(gca,[corxy corxy+anchoxy],[coryy+anchoyy/2 coryy+anchoyy/2]);
setLabelVisible(dy,false);
reglay = getDistance(dy)/1.34;
set(handles.texdery,'string',reglay)

pause(t);

boundariesy = bwboundaries(binaryImage);
x = boundariesy{1}(:, 2);
y = boundariesy{1}(:, 1);
axes(handles.gradercom);
imshow(pieDerecho);
hold on;
plot(x, y, 'r', 'LineWidth', 1);
dl=imdistline(gca,[corx+anchox/2 corx+anchox/2],[cory cory+anchoy]);
dx=imdistline(gca,[corxx corxx+anchoox],[ (coryx+anchoyx/2)+lx
(coryx+anchoyx/2)+lx]);
dy=imdistline(gca,[corxy corxy+anchoxy],[ (coryy+anchoyy/2)+ly
(coryy+anchoyy/2)+ly]);
setLabelVisible(dl,false);
setLabelVisible(dx,false);
setLabelVisible(dy,false);
%-----Ecuacion Hernández corvo
hcd= ((reglax-reglay)/reglax)*100;
set(handles.texderher,'string',hcd)
if hcd >= 0 && hcd <= 34.9
    set(handles.texdertip,'string','Plano')
    axes(handles.graderpla);
plader = stlread('C:\Users\Fernando\Documents\Podoscopio\plader.stl');
handles.plader=plader;
guidata(hObject,handles)
patch(plader,'FaceColor',[0.8 0.8 1.0], ...
      'EdgeColor','none', ...
      'FaceLighting','gouraud', ...)

```

```

        'AmbientStrength', 0.15);
camlight('headlight');
material('dull');
axis('image');
view([-135 35]);

end
if hcd >= 35 && hcd <= 39.9
    set(handles.texdertip,'string','Plano normal')
    axes(handles.graderpla);
plader = stlread('C:\Users\Fernando\Documents\Podoscopio\plader.stl');
handles.plader=plader;
guidata(hObject,handles)
patch(plader,'FaceColor', [0.8 0.8 1.0], ...
    'EdgeColor', 'none', ...
    'FaceLighting', 'gouraud', ...
    'AmbientStrength', 0.15);
camlight('headlight');
material('dull');
axis('image');
view([-135 35]);
rotate3d on
end
if hcd >= 40 && hcd <= 54.9
    set(handles.texdertip,'string','Normal')
end

if hcd >= 55 && hcd <= 59.9
    set(handles.texdertip,'string','Normal cavo')
end
if hcd >= 60 && hcd <= 74.9
    set(handles.texdertip,'string','Cavo')
    axes(handles.graderpla);
plader = stlread('C:\Users\Fernando\Documents\Podoscopio\plader.stl');
handles.plader=plader;
guidata(hObject,handles)
patch(plader,'FaceColor', [0.8 0.8 1.0], ...
    'EdgeColor', 'none', ...
    'FaceLighting', 'gouraud', ...
    'AmbientStrength', 0.15);
camlight('headlight');
material('dull');
axis('image');
view([-135 35]);
rotate3d on
end
if hcd >= 75 && hcd <= 84.9
    set(handles.texdertip,'string','Cavo fuerte')
    axes(handles.graderpla);
plader = stlread('C:\Users\Fernando\Documents\Podoscopio\pladerc.stl');
handles.plader=plader;
guidata(hObject,handles)
patch(plader,'FaceColor', [0.8 0.8 1.0], ...
    'EdgeColor', 'none', ...
    'FaceLighting', 'gouraud', ...
    'AmbientStrength', 0.15);
camlight('headlight');

```

```

material('dull');
axis('image');
view([-135 35]);
rotate3d on
end
if hcd >= 85 && hcd <= 100
    set(handles.texdertip,'string','Cavo extremo')
    axes(handles.graderpla);
plader = stlread('C:\Users\Fernando\Documents\Podoscopio\pladerc.stl');
handles.plader=plader;
guidata(hObject,handles)
patch(plader,'FaceColor', [0.8 0.8 1.0], ...
      'EdgeColor', 'none', ...
      'FaceLighting', 'gouraud', ...
      'AmbientStrength', 0.15);
camlight('headlight');
material('dull');
axis('image');
view([-135 35]);
rotate3d on
end
%-----

% --- Executes on button press in botizqini.
function botizqini_Callback(hObject, eventdata, handles)
% hObject handle to botizqini (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%folder = 'C:\Users\TRABAJO\Documents\MATLAB\podoscopio\';
barra = get(handles.barra,'Value')
barral = (barra*40);
set(handles.texambbar,'string', num2str(barral))
t=0.75;
set(handles.texizqeta,'ForegroundColor','red')
set(handles.texizqeta,'string','Imagen RGB 20%')
fullFileName = imread('fer.jpg');
pieIzquierdo = imcrop(fullFileName,[233 0 232 470]);
axes(handles.graizqlar);
imshow(pieIzquierdo);
pause(t);
set(handles.texizqeta,'string','Imagen RGB 66%')
pause(t);
set(handles.texizqeta,'ForegroundColor','green')
set(handles.texizqeta,'string','Imagen RGB 100%')
pause(t);

set(handles.texizqeta,'ForegroundColor','red')
set(handles.texizqeta,'string','Extraccion color verde 29%')
redChannel = pieIzquierdo(:, :, 1);
greenChannel = pieIzquierdo(:, :, 2);
blueChannel = pieIzquierdo(:, :, 3);
axes(handles.graizqlar);
imshow(greenChannel);
pause(t);
set(handles.texizqeta,'string','Extraccion color verde 76%')
pause(t);
set(handles.texizqeta,'ForegroundColor','green')

```

```
set(handles.texizqeta,'string','Extraccion color verde 100%')
pause(t);
```

```
set(handles.texizqeta,'ForegroundColor','red')
set(handles.texizqeta,'string','Cambio a negativo 12%')
negativo=255-greenChannel;
axes(handles.graizqlar);
imshow(negativo);
pause(t);
set(handles.texizqeta,'string','Cambio a negativo 75%')
pause(t);
set(handles.texizqeta,'ForegroundColor','green')
set(handles.texizqeta,'string','Cambio a negativo 100%')
pause(t);
```

```
set(handles.texizqeta,'ForegroundColor','red')
set(handles.texizqeta,'string','Cambio a Binario 26%')
binaryImage = negativo < barral;
axes(handles.graizqlar);
imshow(binaryImage);
pause(t);
set(handles.texizqeta,'string','Cambio a Binario 59%')
pause(t);
set(handles.texizqeta,'ForegroundColor','green')
set(handles.texizqeta,'string','Cambio a Binario 100%')
pause(t);
```

```
set(handles.texizqeta,'ForegroundColor','red')
set(handles.texizqeta,'string','rellenando huecos 56%')
binaryImage = imfill(binaryImage, 'holes');
axes(handles.graizqlar);
imshow(binaryImage);
pause(t);
set(handles.texizqeta,'string','rellenando huecos 80%')
pause(t);
set(handles.texizqeta,'ForegroundColor','green')
set(handles.texizqeta,'string','rellenando huecos 100%')
pause(t);
```

```
set(handles.texizqeta,'ForegroundColor','red')
set(handles.texizqeta,'string','Removiendo fondo 47%')
binaryImage = bwareaopen(binaryImage, 5000);
binaryImage = imclose(binaryImage, true(5));
axes(handles.graizqlar);
imshow(binaryImage);
pause(t);
set(handles.texizqeta,'string','Removiendo fondo 72%')
pause(t);
set(handles.texizqeta,'ForegroundColor','green')
set(handles.texizqeta,'string','Removiendo fondo 100%')
pause(t);
```

```

Box = regionprops(binaryImage, 'BoundingBox');
vect1=Box.BoundingBox;
corx=vect1(1);
cory=vect1(2);
anchox=vect1(3);
anchoy=vect1(4);
boundaries = bwboundaries(binaryImage);
x = boundaries{1}(:, 2);
y = boundaries{1}(:, 1);
axes(handles.graizqcom);
imshow(pieIzquierdo);
hold on;
plot(x, y, 'r', 'LineWidth', 1);
d=imdistline(gca, [corx+anchox/2 corx+anchox/2], [cory cory+anchoy]);
setLabelVisible(d, false);
regla = (getDistance(d))/1.34;
set(handles.texizqlar, 'string', regla);

    if regla >= 195 && regla <= 230
        disx = imcrop(binaryImage, [0 120 232 50]);
        disy = imcrop(binaryImage, [0 250 232 20]);
        lx=105;
        ly=250;
    end
    if regla >= 150 && regla <= 194
        disx = imcrop(binaryImage, [0 115 232 50]);
        disy = imcrop(binaryImage, [0 200 232 20]);
        lx=110;
        ly=200;
    end
    if regla >= 110 && regla <= 149
        disx = imcrop(binaryImage, [0 110 232 50]);
        disy = imcrop(binaryImage, [0 150 232 20]);
        lx=100;
        ly=150;
    end
set(handles.texizqeta, 'string', 'Medida X')
Boxx = regionprops(disx, 'BoundingBox');
vecx=Boxx.BoundingBox;
corxx=vecx(1);
coryx=vecx(2);
anchoxx=vecx(3);
anchoyx=vecx(4);
boundariesx = bwboundaries(disx);
x = boundariesx{1}(:, 2);
y = boundariesx{1}(:, 1);
axes(handles.graizqx);
imshow(disx);
hold on;
plot(x, y, 'r', 'LineWidth', 1);
dx=imdistline(gca, [corxx corxx+anchoxx], [coryx+anchoyx/2 coryx+anchoyx/2]);
setLabelVisible(dx, false);
reglax = getDistance(dx)/1.34;
set(handles.texizqx, 'string', reglax)

pause(t);

```

```

set(handles.texizqeta, 'string', 'Medida Y')
Boxy = regionprops(dis, 'BoundingBox');
vecy=Boxy.BoundingBox;
corxy=vecy(1);
coryy=vecy(2);
anchoxy=vecy(3);
anchoyy=vecy(4);
boundariesy = bwboundaries(dis);
x = boundariesy{1}(:, 2);
y = boundariesy{1}(:, 1);
axes(handles.graizqy);
imshow(dis);
hold on;
plot(x, y, 'r', 'LineWidth', 1);
dy=imdistline(gca, [corxy corxy+anchoxy], [coryy+anchoyy/2 coryy+anchoyy/2]);
setLabelVisible(dy, false);
reglay = getDistance(dy)/1.34;
set(handles.texizqy, 'string', reglay)

boundariesy = bwboundaries(binaryImage);
x = boundariesy{1}(:, 2);
y = boundariesy{1}(:, 1);
axes(handles.graizqcom);
imshow(pieIzquierdo);
hold on;
plot(x, y, 'r', 'LineWidth', 1);
dl=imdistline(gca, [corx+anchox/2 corx+anchox/2], [cory cory+anchoy]);
dx=imdistline(gca, [corxx corxx+anchoxx], [(coryx+anchoyx/2)+lx
(coryx+anchoyx/2)+lx]);
dy=imdistline(gca, [corxy corxy+anchoxy], [(coryy+anchoyy/2)+ly
(coryy+anchoyy/2)+ly]);
setLabelVisible(dl, false);
setLabelVisible(dx, false);
setLabelVisible(dy, false);

%-----Ecuacion Hernández corvo
hcd=((reglax-reglay)/reglax)*100;
set(handles.texizqher, 'string', hcd)
if hcd >= 0 && hcd <= 34.9
    set(handles.texizqtip, 'string', 'Plano')
    axes(handles.graizqpla);
    plaizq = stlread('C:\Users\Fernando\Documents\Podoscopio\plaizq.stl');
    handles.plaizq=plaizq;
    guidata(hObject, handles)
    patch(plaizq, 'FaceColor', [0.8 0.8 1.0], ...
        'EdgeColor', 'none', ...
        'FaceLighting', ' gouraud', ...
        'AmbientStrength', 0.15);
    camlight('headlight');
    material('dull');
    axis('image');
    view([-135 35]);
    rotate3d on
end

if hcd >= 35 && hcd <= 39.9

```

```

        set(handles.texizqtip,'string','Plano normal')
        axes(handles.graizqpla);
    plaizq = stlread('C:\Users\Fernando\Documents\Podoscopio\plaizq.stl');
    handles.plaizq=plaizq;
    guidata(hObject,handles)
    patch(plaizq,'FaceColor',[0.8 0.8 1.0], ...
        'EdgeColor','none', ...
        'FaceLighting','gouraud', ...
        'AmbientStrength', 0.15);
    camlight('headlight');
    material('dull');
    axis('image');
    view([-135 35]);
    rotate3d on
end

if hcd >= 40 && hcd <= 54.9
    set(handles.texizqtip,'string','Normal')
end

if hcd >= 55 && hcd <= 59.9
    set(handles.texizqtip,'string','Normal cavo')
end
if hcd >= 60 && hcd <= 74.9
    set(handles.texizqtip,'string','Cavo')
    axes(handles.graizqpla);
    plaizq = stlread('C:\Users\Fernando\Documents\Podoscopio\plaizq.stl');
    handles.plaizq=plaizq;
    guidata(hObject,handles)
    patch(plaizq,'FaceColor',[0.8 0.8 1.0], ...
        'EdgeColor','none', ...
        'FaceLighting','gouraud', ...
        'AmbientStrength', 0.15);
    camlight('headlight');
    material('dull');
    axis('image');
    view([-135 35]);
    rotate3d on
end
if hcd >= 75 && hcd <= 84.9
    set(handles.texizqtip,'string','Cavo fuerte')
    axes(handles.graizqpla);
    plaizq = stlread('C:\Users\Fernando\Documents\Podoscopio\plaizqc.stl');
    handles.plaizq=plaizq;
    guidata(hObject,handles)
    patch(plaizq,'FaceColor',[0.8 0.8 1.0], ...
        'EdgeColor','none', ...
        'FaceLighting','gouraud', ...
        'AmbientStrength', 0.15);
    camlight('headlight');
    material('dull');
    axis('image');
    view([-135 35]);
    rotate3d on
end
if hcd >= 85 && hcd <= 100
    set(handles.texizqtip,'string','Cavo extremo')

```

```

    axes(handles.graizqpla);
    plaizq = stlread('C:\Users\Fernando\Documents\Podoscopio\plaizqc.stl');
    handles.plaizq=plaizq;
    guidata(hObject,handles)
    patch(plaizq,'FaceColor',[0.8 0.8 1.0], ...
          'EdgeColor','none', ...
          'FaceLighting','gouraud', ...
          'AmbientStrength', 0.15);
    camlight('headlight');
    material('dull');
    axis('image');
    view([-135 35]);
    rotate3d on
end

% --- Executes on slider movement.
function barra_Callback(hObject, eventdata, handles)
barra = get(handles.barra,'Value')
barra1 = barra*40;
set(handles.texambbar,'string', num2str(barra1))

% hObject    handle to barra (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function barra_CreateFcn(hObject, eventdata, handles)
% hObject    handle to barra (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function texambbar_Callback(hObject, eventdata, handles)
% hObject    handle to texambbar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of texambbar as text
%         str2double(get(hObject,'String')) returns contents of texambbar as a
double

% --- Executes during object creation, after setting all properties.

```

```

function texambbar_CreateFcn(hObject, eventdata, handles)
% hObject    handle to texambbar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in botizqpla.
function botizqpla_Callback(hObject, eventdata, handles)
plaizq=handles.plaizq;
stlwrite(uiinputfile('PieIzquierdo.stl'), plaizq);
% hObject    handle to botizqpla (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in botderpla.
function botderpla_Callback(hObject, eventdata, handles)
plader=handles.plader;
stlwrite(uiinputfile('PieDerecho.stl'), plader);
% hObject    handle to botderpla (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```