



Universidad de la República

CINARTRO: Herramienta de estudio de la cinemática de la rodilla por videofluoroscopia

Memoria de un Proyecto de Informática Médica presentada a la
Facultad de Ingeniería

por

*Marcio Rodríguez
Williams Olivera*

Con la tutoría del Prof. Ing. Franco Simini y en colaboración con el Prof. Adj. M.Sc Darío Santos, en cumplimiento parcial de los elementos requeridos para la obtención del título de Ingeniero en Computación

Núcleo de Ingeniería Biomédica de las Facultades de Medicina e Ingeniería
Hospital de Clínicas, piso 15
Montevideo – URUGUAY

Mayo de 2016



nib
núcleo de ingeniería biomédica
facultades de ingeniería y medicina
universidad de la república

Resumen

El proyecto CINARTRO consiste en el diseño e implementación de una aplicación de escritorio para el estudio asistido de la cinemática de la rodilla. Imágenes de rayos X tomadas de una videofluoroscopia son cargadas en el programa, en las cuales el usuario ubica un conjunto de puntos que permiten determinar el Punto de Contacto Tibio-Femoral y el Brazo de Momento con el cual trabaja el cuádriceps. El programa genera informes clínicos a partir de los resultados en formato PDF, Excel y CDA, donde este último es subido a un repositorio XDS como parte de la historia electrónica del paciente. Para el problema de distorsión en las imágenes causadas por el efecto *pincushion*, fueron implementados un algoritmo de reconocimiento de grises y un algoritmo de corrección de distorsión basado en un modelo teórico.

CINARTRO fue probado en 3 pacientes en su rodilla sana y en su rodilla lesionada con el Ligamento Cruzado Anterior roto. Se determinó un tiempo promedio de 60 minutos en completar una evaluación para ambos estados de la rodilla y se detectaron errores de precisión introducidos durante la ubicación de los puntos. Los valores obtenidos para el Punto de Contacto Tibio-Femoral muestran un promedio del 59% de desplazamiento sobre el platillo tibial en una rodilla sana y 49% en una lesionada, mientras que para el Brazo de Momento se obtuvo un promedio de 50 mm y 45 mm en rodilla sana y lesionada respectivamente.

Palabras clave: Informática médica, procesamiento de imágenes, informe clínico, cinemática de la rodilla.

Índice General

Índice General	5
Índice de Figuras.....	7
Índice de Tablas.....	9
Capítulo 1	11
Introducción	11
Capítulo 2	15
Estado del arte	15
2.1. Tecnologías para la salud en nuestro país.....	15
2.2. Conceptos previos	16
2.3. Marco teórico.....	20
2.3.1. Método de corrección de la distorsión de Baltzopoulos.....	20
2.3.2. Otros métodos de procesamiento de videofluoroscopia.....	23
2.4. Herramientas existentes	27
Capítulo 3	29
Especificación del software	29
Capítulo 4	33
Interfaz y flujo del programa.....	33
4.1. Procedimientos previos	33
4.2. Descripción del flujo de trabajo	35
Capítulo 5	43
Arquitectura y Diseño.....	43
5.1. Arquitectura del sistema.....	43
5.2. Diseño de la aplicación	44
Capítulo 6	47
Implementación.....	47
6.1. Herramientas utilizadas	47
6.1.1. Principales características.....	47
6.1.2. Otras herramientas	50
6.2. Aspectos de programación	51
6.2.1. Estructuras básicas.....	52
6.2.2. Cálculo de parámetros de interés.....	52
6.2.3. Digitalización del fantoma	54
6.2.4. Corrección de puntos	55
6.2.5. Implementación de tests.....	56
Capítulo 7	57
Pruebas	57
7.1. Procedimiento médico con pacientes.....	57
7.2. Tiempo de realización de estudios	58
7.3. Error global del método de corrección de la distorsión	59
7.4. Pruebas con usuarios finales.....	61
7.5. Performance	62
7.6. Compatibilidad	64
Capítulo 8	67
Resultados.....	67
8.1. Resultados obtenidos tras la implementación del software.....	67
8.2. Resultados de los cálculos de interés	68

8.3. Resultados en los métodos de corrección.....	74
Capítulo 9	77
Conclusiones.....	77
Capítulo 10	79
Trabajos futuros.....	79
Bibliografía.....	81
Glosario.....	85
Apéndice A.....	89
Utilización del índice EMPI	89
Apéndice B.....	93
Creación de un reporte CDA y utilización de repositorio XDS	93
Apéndice C	97
Contenido para Ingeniería de Muestra	97
Apéndice D	101
Paper de CINARTRO	101
Apéndice E.....	107
Curso Biomecánica de la Rodilla	107
Apéndice F	109
Gestión.....	109
Apéndice G	113
Representación de clases	113

Índice de Figuras

Figura 1: Estructura de la rodilla.	16
Figura 2: Lesión del Ligamento Cruzado Anterior.	17
Figura 3: Ubicación del PCTF y el Brazo de momento.	18
Figura 4: Fantoma.....	19
Figura 5: Esquema de un equipo de videofluoroscopia.	21
Figura 6: Construcción de Reuleaux.	25
Figura 7: Arco en C y equipo de videofluoroscopia.	33
Figura 8: Radiografía que cumple con características de buena calidad.....	34
Figura 9: Radiografía considerada de mala calidad.	35
Figura 10: Vista inicial al ejecutar el programa CINARTRO.....	36
Figura 11: Pantalla para registro de pacientes.....	37
Figura 12: Vista previa de imágenes cargadas al programa.	38
Figura 13: Pantalla para selección de puntos anatómicos de referencia.....	38
Figura 14: Pantalla con los puntos anatómicos ubicados por el usuario en la imagen.	39
Figura 15: Gráficas presentadas por CINARTRO.	40
Figura 16: Imagen del fantoma.....	41
Figura 17: Ingreso de la información institucional y del usuario.....	42
Figura 18: Preferencias de idioma.	42
Figura 19: Componentes de la arquitectura de CINARTRO.	43
Figura 20: Diagrama de clases simplificado de CINARTRO.....	45
Figura 21: Representación del cálculo del Punto de Contacto Tibio Femoral (PCTF).	53
Figura 22: Reconocimiento de zonas grises en cada punto del fantoma.....	55
Figura 23: Uso de memoria previo a optimización.	62
Figura 24: Uso de memoria posterior a optimización.	63
Figura 25: Estudio del PCTF en rodilla sana por Kellis.....	70
Figura 26: PCTF del paciente 1 estudiado por CINARTRO.....	70
Figura 27: PCTF del paciente 2 estudiado por CINARTRO.....	71
Figura 28: PCTF del paciente 3 estudiado por CINARTRO.....	71
Figura 29: Promedio del PCTF en tres pacientes para rodilla sana y lesionada.	72
Figura 30: Brazo de Momento (BM) del paciente 1 estudiado por CINARTRO.....	73
Figura 31: BM del paciente 2 estudiado por CINARTRO.....	73
Figura 32: BM del paciente 3 estudiado por CINARTRO.....	74
Figura 33: Consola de administración de una cuenta EMPI.....	90
Figura 34: Ejemplo de mensaje PIX/PDQ en HL7.....	91
Figura 35: Métodos de búsqueda y creación de pacientes en EMPI.....	92
Figura 36: XML no parseable por JAXB.	94
Figura 37: Representación interna del CDA.	94
Figura 38: Resultado en XML del CDA generado.	95
Figura 39: Póster de CINARTRO presentado en Ingeniería de Muestra.	98
Figura 40: Folleto presentado en Ingeniería de Muestra.	99
Figura 41: Folleto presentado en Ingeniería de Muestra.	100
Figura 42: Paper CINARTRO. Página 1.....	102
Figura 43: Paper CINARTRO. Página 2.....	103
Figura 44: Paper CINARTRO. Página 3.....	104
Figura 45: Paper CINARTRO. Página 4.....	105

Figura 46: Paper CINARTRO. Página 5.....	106
Figura 47: Cálculo del Centro de Rotación Instantáneo (CRI) utilizando velocidades.	108
Figura 48: Gráfica del total de horas en promedio de los integrantes por área de trabajo. ...	109
Figura 49: Distribución por área de trabajo.	110
Figura 50: Distribución por categoría del esfuerzo en el área de implementación.	110
Figura 51: Diagrama de Gantt con la distribución de categorías de trabajo.....	112
Figura 52: Clase CinartroCDA.....	113
Figura 53: Clase CinartroImageController.	114
Figura 54: Clase CinartroReportController.	114
Figura 55: Clase CinartroPersistenceController.	115
Figura 56: Clase CinartroXMLController.	115
Figura 57: Clase CinartroMedicalController.	115
Figura 58: Clase CinartroPatientController.	116
Figura 59: Clase CinartroPDFController.	116
Figura 60: Clase CinartroExcelController.....	116
Figura 61: Clase AnglePointsDT.	116
Figura 62: Clase SquareDT.....	117
Figura 63: Clase CorrectionPointsDT.....	117
Figura 64: Clase CinartroRect.....	117
Figura 65: Clase CinartroCore.	118
Figura 66: Clase CinartroPhantomCore.....	118
Figura 67: Clase CinartroImage.	119
Figura 68: Clase CinartroPatient.	120
Figura 69: Clase CinartroMedical.	120
Figura 70: Clase CinartroReportStudy.....	121
Figura 71: Clase CinartroPhantomValues.....	121
Figura 72: Clase CinartroPhantom.	122
Figura 73: Clase ScalingImage.	122
Figura 74: Clase OpenEmpiConnector.	122
Figura 75: Clase XDSRepositoryConnector.	123
Figura 76: Clase VideoThumbnails.....	123

Índice de Tablas

TABLA 1: ALTERNATIVAS A LAS PRINCIPALES CARACTERÍSTICAS DE CINARTRO.....	47
TABLA 2: HERRAMIENTAS Y CONSIDERACIONES PARA EL DESARROLLO DE CINARTRO.....	50
TABLA 3: OPTIMIZACIONES DEL PROGRAMA.....	59
TABLA 4: ERROR GLOBAL DE LA CORRECCIÓN SEGÚN NÚMERO DE PUNTOS DE CONTROL.	61
TABLA 5: REQUERIMIENTOS DE HARDWARE MÍNIMOS.....	64
TABLA 6: COMPATIBILIDAD CON SISTEMAS OPERATIVOS.....	65
TABLA 7: PUNTO DE CONTACTO TIBIO-FEMORAL EN TRES PACIENTES LESIONADOS.....	69
TABLA 8: BRAZO DE MOMENTO EN TRES PACIENTES LESIONADOS.....	69

Capítulo 1

Introducción

Motivación

En el transcurso de la vida muchas personas practican, entre tantas otras cosas, diversas disciplinas deportivas con el objetivo de mantener su salud y bienestar, tanto mental como físico, ya sea profesionalmente o como actividad de tiempo libre. Sin embargo, de forma contraproducente, en muchas oportunidades se enfrentan a diversas lesiones que impiden desarrollar dichas actividades con normalidad.

En la extensa rama de la medicina, una de las lesiones más comunes que experimentan los deportistas es la rotura de los ligamentos cruzados (Griffin et al. [1], Gianotti et al. [2]). Esta aflicción se produce en la zona media de la rodilla, más comúnmente en el Ligamento Cruzado Anterior, afectando el movimiento natural de la articulación y su contención al desplazarse. Cuando una persona experimenta este daño, recibe la recomendación de someterse a una operación que logre reparar el área afectada y le permita reincorporarse a su rutina. Luego de que el paciente experimenta la cirugía, comienza para él un arduo proceso de recuperación (Myklebust y Bahr [3], Kruse et al. [4]).

Desde el punto de vista clínico, si nos focalizamos en este proceso postoperatorio, los médicos realizan el seguimiento de los pacientes mediante análisis periódicos que permiten evaluar la adaptación del individuo a su rodilla reparada. Además de las evaluaciones terapéuticas que se realizan sobre el paciente, el especialista desarrolla otras actividades de análisis interno que incluyen comúnmente resonancias magnéticas y fluoroscopías, entre otros. Ciertamente esto ofrece una forma relativamente práctica para la determinación del estado de la rodilla; sin embargo, se tiene solamente una percepción cualitativa de los elementos y no así cuantitativa que permita identificar cómo las partes de interés se trasladan, modifican e interactúan entre sí (Simini y Santos [5], Santos [6]).

Una evaluación desestimada puede traer problemas a futuro para la rodilla del individuo e incluso sobre la rodilla contralateral, que es utilizada para compensar el mal movimiento de la rodilla lesionada. La valoración clínica en estos casos, resulta muchas veces subjetiva si se quiere afirmar con certeza que una rodilla está en vías de una recuperación satisfactoria; el error puede resultar grande si no se determinan y justifican adecuadamente los resultados analizados.

En los últimos años se han realizado diversos estudios referentes a la evaluación cuantitativa del movimiento de la articulación lesionada en comparación con la sana, de los que se desprenden algunos modelos matemáticos que analizan la cinemática de la rodilla para estos casos, intentando minimizar el error de medición (Baltzopoulos [7], Kellis y Baltzopoulos [8], Bryant et al. [9], Giphart [10]). Todos estos estudios están limitados al ámbito de la investigación y no tiene aplicación en la clínica por su alto costo y dificultad de implementación. Resulta de gran importancia para los especialistas médicos y la rama de la fisioterapia en general, contar con herramientas que ayuden a mejorar el procedimiento de análisis tras una operación del Ligamento Cruzado Anterior.

Propósito general y resultados esperados

El propósito de este trabajo es la construcción de una aplicación que analice la cinemática de la rodilla y que sea apta para el uso rutinario clínico, incorporando el resultado de la investigación en el estado del arte actual, poniéndolo a disposición de los pacientes.

Se plantea procesar una serie de imágenes de rayos X tomadas de una videofluoroscopia y calcular dos parámetros de interés resumiéndolos en un estudio de uso clínico. Estos parámetros son el Punto de Contacto Tibio-Femoral (PCTF) y el Brazo de Momento (BM), siendo ambos de gran importancia para el análisis de la cinemática de la rodilla. En particular, se desea utilizarlos para determinar el estado de una rodilla tras una operación de Ligamento Cruzado Anterior. En el Capítulo 2 se presentan más detalles sobre estos elementos y conceptos.

Las imágenes de entrada representan en su conjunto un movimiento de extensión de un paciente. Los parámetros deben ser calculados a partir de 11 puntos marcados por el usuario sobre cada una de estas imágenes, por lo que el software debe permitir realizar esta interacción. El producto que nos planteamos desarrollar será empleado por personal del área de la salud (Médicos y especialistas en Fisioterapia y Rehabilitación, entre otros), por lo que utilizaremos la terminología adecuada para los conceptos manejados así como una interfaz simple e intuitiva. Además, es importante presentar la información al usuario mediante gráficas para que sea visualizada fácilmente y den un panorama de los resultados obtenidos.

La Informática Médica cuenta con diversos estándares que marcan los lineamientos para el intercambio de información entre sistemas de salud, por lo que es de interés la integración y adaptación con servicios externos para generar datos que sean comunicados siguiendo los protocolos y mecanismos establecidos en la actualidad. Respecto a esto, es importante la integración en dos sentidos: la identificación de pacientes y el aporte a su historia clínica. En el primero de los casos se requiere que los usuarios sean registrados unívocamente a través de un sistema EMPI (*Enterprise Master Patient Index*), de forma que los estudios realizados estén asociados a un único individuo. Se pueden obtener más detalles sobre EMPI en el Capítulo 6 y el Apéndice A. En segundo lugar, se espera que la aplicación sea capaz de generar

documentos en formato CDA (*Clinical Document Architecture*) [11], conteniendo la información de los estudios realizados y que éstos sean agregados junto a la historia clínica del paciente en un repositorio XDS [12]. En el Capítulo 6 y Apéndice B se profundiza sobre estos conceptos.

En cuanto a los resultados desde el punto de vista de la investigación, el objetivo es utilizar el programa CINARTRO sobre imágenes radiográficas de pacientes y evaluar los datos obtenidos para los parámetros de interés en diferentes estados de la rodilla y cómo es el comportamiento a lo largo del movimiento. Si bien no se espera de CINARTRO que realice la interpretación automática de los datos desde un punto de vista médico, es su función facilitar la comparación de los resultados con los valores obtenidos en los estudios previos.

Todos estos aspectos representan los principales propósitos a los cuáles está destinado CINARTRO. La intencionalidad del programa tiene como premisa el concepto de herramienta y su utilidad en los procedimientos rutinarios médicos en Traumatología y en Medicina de la Rehabilitación.

Organización del documento

El documento a partir de aquí, está organizado de la siguiente manera.

El capítulo 2 define el marco teórico que fundamenta el propósito de la investigación. Las primeras secciones introducen los conceptos claves, tanto para la comprensión de las restantes partes del capítulo, así como para el proyecto en general. Seguidamente, se pone foco en el estudio de las principales publicaciones utilizadas en nuestro trabajo. La tercera sección está dedicada a realizar comparaciones con otros productos existentes.

En el capítulo 3 se define la especificación de la aplicación de software, detallando las características esperadas del producto.

El capítulo 4 está enfocado en describir los principales escenarios de CINARTRO en una perspectiva de alto nivel. La primera parte describe los procedimientos clínicos para la obtención de imágenes de rayos X. Posteriormente se profundiza en la descripción de las distintas pantallas de la aplicación.

Luego, en el capítulo 5 se presentan la arquitectura de CINARTRO y el diseño del software, describiendo primeramente los componentes involucrados en el sistema y luego cómo fue diseñada la solución previamente a la implementación.

En el capítulo 6, desde una perspectiva más técnica y a bajo nivel, se profundiza en los aspectos de implementación. Primero mencionando las herramientas utilizadas y luego detallando las clases y principales aspectos de programación.

Posteriormente, el capítulo 7 puntualiza las pruebas realizadas en pacientes y con elementos de calibración, las cuales se dividen en varias secciones que cubren los distintos aspectos del desarrollo.

En el capítulo 8 se presentan los resultados obtenidos del proyecto tras analizar las pruebas realizadas.

El capítulo 9 y 10 detallan la conclusión del proyecto y el trabajo a futuro respectivamente.

Posteriormente sigue un glosario, con las definiciones y descripciones de los conceptos manejados a lo largo del manuscrito.

Finalmente se da pie a la sección de apéndices conteniendo la descripción de otros aspectos interesantes del proyecto.

Capítulo 2

Estado del arte

El presente capítulo define el marco teórico que fundamenta el propósito de la investigación. A través de las siguientes secciones, se describen y citan trabajos académicos realizados por diversos autores donde se trata de proveer las herramientas necesarias para la comprensión de la problemática así como demostrar que la propuesta no está aislada en el área de investigación sino que tiene sus bases formadas en estudios previamente realizados.

2.1. Tecnologías para la salud en nuestro país

En nuestro país los procesos y políticas públicas referentes a Tecnologías de la Información y Comunicaciones, entre ellos el área de salud, son abordados e impulsados por la Agencia de Gobierno Electrónico y Sociedad de la Información y del Conocimiento de Uruguay (AGESIC) [13]. En particular esta entidad lleva a cabo un programa denominado Salud.uy [14] que se encarga de propulsar la modernización en términos de Tecnología de la Información y Comunicación (TIC) en el área de salud y viabilizar mejoras en la calidad de las prestaciones sanitarias recibidas por los ciudadanos. El programa Salud.uy se encarga de proveer prácticas informativas que ayudan a la planificación del diseño arquitectónico necesario para la creación de información valiosa, que pueda ser reutilizable de forma universal en el área de software clínico.

En el documento Portilla [15], se destacan los lineamientos generales acerca del funcionamiento que aborda el programa de Salud.uy. A partir de allí se deriva un entendimiento de cómo trabajan las distintas partes que hacen posible el funcionamiento del intercambio de datos médicos a través de los conceptos de Historias Clínicas Electrónicas, Técnicas de Interoperabilidad y los diversos protocolos existentes para la formación de archivos que en cierto punto el médico podrá utilizar a través de CINARTRO.

2.2. Conceptos previos

A continuación, presentamos los conceptos más relevantes, manejados en las siguientes partes del capítulo. El lector puede obtener más detalles refiriéndose a los enlaces bibliográficos.

Estructura general de la rodilla:

La rodilla es una articulación del cuerpo humano ubicada en la zona media de la pierna. Está constituida principalmente por tres huesos que son el fémur, la tibia y la rótula. Está rodeada por varios ligamentos y músculos que permiten la estabilidad de toda la pierna. Es la articulación más compleja de todo el cuerpo, permitiendo a la pierna realizar movimientos de extensión, flexión y de rotación (Easley et al. [16]). En la Figura 1 se muestran los principales componentes de la rodilla.

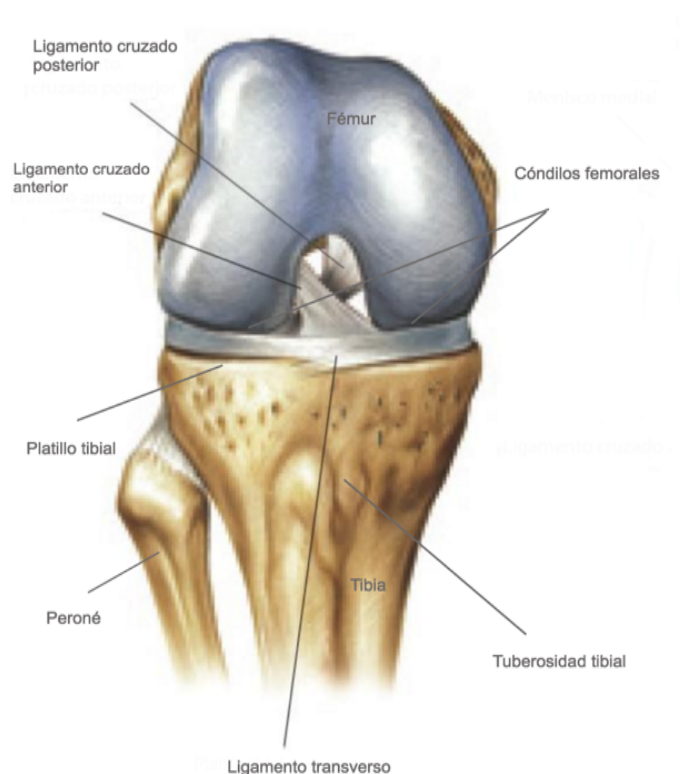


Figura 1: Estructura de la rodilla.
Tomado de J. Rodés [17].

Ligamento Cruzado Anterior:

El Ligamento Cruzado Anterior (LCA) es uno de los principales ligamentos de la rodilla; conecta la parte posterior del fémur con la parte anterior de la tibia. El objetivo de este ligamento, es evitar que la tibia se desplace hacia adelante respecto del fémur,

proporcionando equilibrio y rotación controlada al movimiento mutuo de ambos huesos. En la Figura 1 se puede ver este ligamento.

Rotura del LCA:

Partiendo de la definición anterior, la rotura del LCA es precisamente el quiebre del ligamento. Esto produce en el individuo dolor e inflamación y le impide a la rodilla moverse con normalidad (De Valk et al. [18]). En particular la tibia excede la línea del fémur causando irregularidades en el movimiento.

La Figura 2 presenta una visión interna de la rodilla que ilustra el LCA y su estado luego de una rotura.

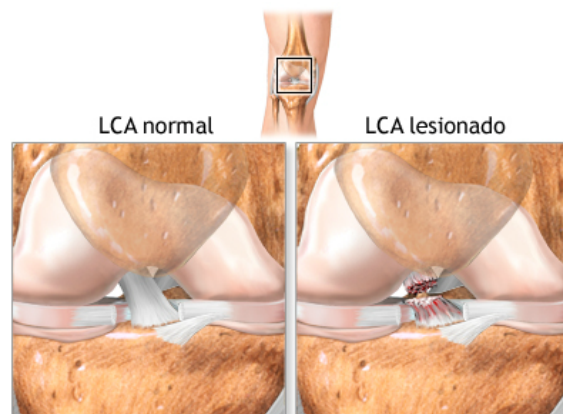


Figura 2: Lesión del Ligamento Cruzado Anterior.

A la izquierda se presenta el ligamento en su estado normal. A la derecha se muestra el ligamento luego de la rotura producida. Tomado de A.D.A.M [19].

Cóndilos femorales:

Los cóndilos femorales son un par de protuberancias circulares que emanan de cada lado de la parte inferior del fémur. El cóndilo medial es el ubicado en la parte más interna de la rodilla mientras que el externo es nombrado cóndilo lateral. Los ligamentos cruzados anterior y posterior están unidos a los cóndilos. La Figura 1 señala los cóndilos femorales dentro de las partes de la rodilla.

Platillo tibial:

El platillo tibial es la parte superior de la tibia y es junto a los cóndilos el otro extremo que une los ligamentos cruzados anterior y posterior. Ubicada como parte de la tibia también podemos identificar la tuberosidad tibial, otro elemento importante que consideramos en el estudio. En la Figura 1 podemos identificar ambos elementos.

Punto de Contacto Tibio-Femoral:

El Punto de Contacto Tibio-Femoral (PCTF) también denominado *Tibio-Femoral Contact Point* (TFCP) por sus siglas en inglés es, como su nombre lo indica, un punto que se ubica en el punto medio del segmento de menor distancia posible entre los cóndilos femorales y el platillo tibial de una rodilla. Este punto es definido con el objetivo de obtener un parámetro cinemático (o de movimiento) de la rodilla, que sea de utilidad para evaluar el estado del Ligamento Cruzado Anterior. El valor del PCTF se expresa como el porcentaje representado por el desplazamiento de dicho punto respecto al largo del platillo (Baltzopoulos [7]). La Figura 3 ubica la posición del punto y los conceptos involucrados.

Brazo de Momento:

Brazo de Momento (BM) o *Moment Arm* (MA) por sus siglas en inglés, se define en general para la rama de la mecánica como la distancia perpendicular desde la línea de acción de una fuerza al eje de rotación. En el caso de la articulación de la rodilla, la línea de acción de la fuerza se define como aquella que se origina a causa de la contracción del cuádriceps, el cual provoca un efecto polea de la rótula. Luego, la rótula termina generando una fuerza sobre la tibia debido al tendón rotuliano. Generalmente, se toma como línea de acción de fuerza, aquella que representa el tendón rotuliano, el cual lo podemos definir desde la tuberosidad tibial hasta el vértice inferior de la rótula. Por lo tanto, se obtiene el Brazo de Momento calculando la distancia perpendicular desde la línea de acción de fuerza (rótula, tuberosidad tibial) hasta el PCTF (Kellis y Baltzopoulos [8], Smidt [20]). La Figura 3 facilita la comprensión de este concepto.

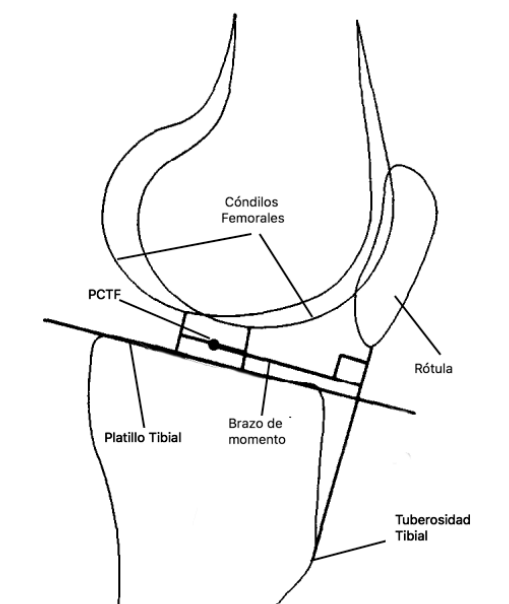


Figura 3: Ubicación del PCTF y el Brazo de momento.
El PCTF es calculado como el punto medio entre la mínima distancia de los cóndilos femorales y el platillo tibial mientras que el Brazo de Momento como la distancia de la línea de acción de la fuerza hasta el PCTF.

Figura tomada de Baltzopoulos [7].

Videofluoroscopia:

Se refiere a una técnica de imagen realizada en un laboratorio clínico para obtener, en forma de video, una toma de rayos X. En general, el sistema se compone de una cámara de video, un amplificador de imagen y la fuente de rayos X. Las imágenes son grabadas y transmitidas a un monitor, lo que luego resulta en un archivo DICOM [21] conteniendo dicho video y la metadata correspondiente. La videofluoroscopia es utilizada en situaciones que es necesario visualizar en tiempo real la movilidad de determinada estructura interna de un paciente. Los casos más utilizados son en las articulaciones (Kellis y Baltzopoulos [8], Baltzopoulos [7]) y en estudios del proceso de deglución (Linden [22]).

Fantoma:

Es un objeto de calibración utilizado para determinar la deformación de una imagen debido al efecto *pincushion* provocado por la curvatura del intensificador de imágenes. Está formado por una placa acrílica que contiene una grilla de puntos de plomo separados de forma equidistante, por lo general entre 1 y 5 centímetros. La Figura 4 muestra el fantoma utilizado en el Núcleo de Ingeniería Biomédica (NIB) empleado para la investigación, con separación entre puntos de 10 mm y de dimensiones 27 cm por 33 cm en dos placas distanciadas por 15 cm. A diferencia del mostrado en Kellis y Baltzopoulos [8], esta versión más actualizada del fantoma no utiliza cables de acero para marcar la grilla sino pequeñas esferas de plomo como material contrastante.



Figura 4: Fantoma.

Perteneciente al Núcleo de Ingeniería Biomédica (NIB). Formado por dos placas de acrílico donde una de ellas está parcialmente cubierta por una grilla de puntos de plomo.

2.3. Marco teórico

Diversos autores han realizado trabajos de investigación enfocados al análisis y comprensión de la anatomía de la rodilla, tratando de encontrar comportamientos que describan la mecánica que la conforma, los parámetros que definen el movimiento, los patrones que se suceden, entre muchas otras interrogantes (Baltzopoulos [7], Kellis y Baltzopoulos [8], Nissel [23], Komistek et al. [24], Laz et al. [25]). En particular para la rama de la biomecánica, es importante la determinación de estas cuestiones en situaciones de anomalía, como es el caso de las rodillas dañadas por rotura de ligamentos cruzados (Micheo et al. [26], Podraza y White [27]).

Entre las publicaciones más relevantes sobre este tema, se destaca el trabajo realizado por Baltzopoulos en 1995 [7] donde se presenta un estudio sobre la cinemática de la rodilla utilizando un método por videofluoroscopia. Este *paper* es el proveedor de las ideas iniciales del presente trabajo, que junto a otros trabajos de la misma índole impulsan lo que es CINARTRO.

2.3.1. Método de corrección de la distorsión de Baltzopoulos

“Un método de videofluoroscopia para la corrección de la distorsión y medición de la cinemática de la rodilla” se titula el *paper* de Baltzopoulos [7] que da inicio a la fundamentación teórica de buena parte de la presente investigación y prolonga los conceptos hasta proveerlos a disposición de la rutina clínica. En este artículo se destaca el hecho de utilizar la videofluoroscopia como el método de análisis de la mecánica articular. Primeramente, se justifica el uso de dicho procedimiento con respecto a métodos como el Análisis Estéreo Fotogramétrico de Roentgen (RSA) que produce dosis de radiación muy invasivas para un uso diario. Explícitamente se dice que *“...El Análisis Estéreo Fotogramétrico de Roentgen (RSA) está basado principalmente en la grabación estática convencional de rayos X, con exposición a relativamente largas dosis de radiación ... Por el contrario, la videofluoroscopia, mediante sistemas de intensificador de imagen, permite adquirir imágenes de rayos X con movimientos de video, reduciendo la exposición a la radiación significativamente.”*

Por otro lado, sobre el uso práctico de la videofluoroscopia menciona que *“... La principal limitación en sistemas de videofluoroscopia es la distorsión óptica de la imagen de rayos X, por lo que métodos de corrección apropiados, son esenciales para el análisis cuantitativo del movimiento grabado.”* Siendo esta limitación uno de los objetivos a cumplir durante el desarrollo y sobre la que se hará mención a lo largo de los capítulos.

A partir de allí, se presenta un método de corrección de la distorsión de la imagen. Tomemos como referencia la Figura 5, donde podemos ver los elementos involucrados y destacar que la distorsión es producida por la superficie curva del intensificador de imagen.

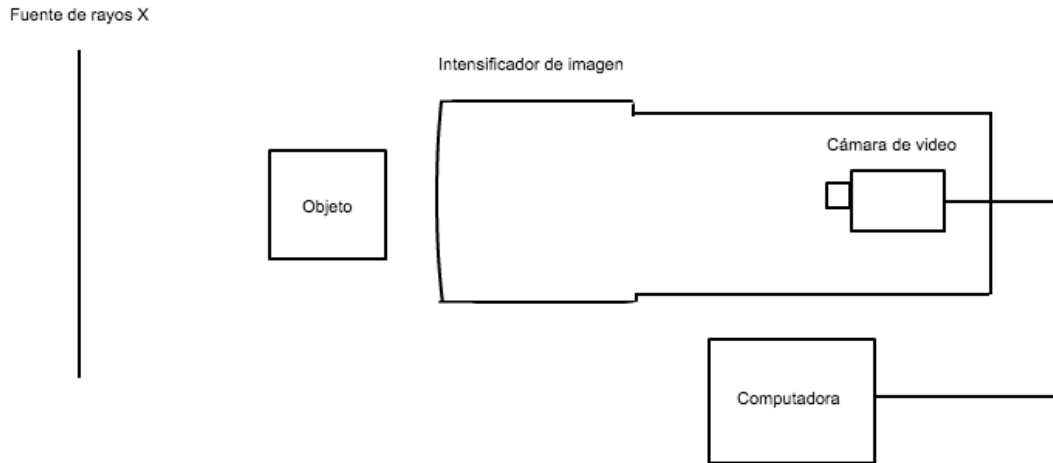


Figura 5: Esquema de un equipo de videofluoroscopia.
La cámara de video capta a través de un intensificador de imagen la aplicación de rayos X sobre un objeto determinado. Luego la información es procesada desde una computadora.

Dando comienzo a la explicación del método, se definen las coordenadas globales X_P e Y_P de cualquier punto del plano del objeto, las cuales son computadas desde el correspondiente sistema de coordenadas de video x_P e y_P usando las funciones polinomiales:

$$X_P = a_1 + a_2x_P + a_3y_P + a_4x_P^2 + a_5x_Py_P + a_6y_P^2 \quad (1)$$

$$Y_P = b_1 + b_2x_P + b_3y_P + b_4x_P^2 + b_5x_Py_P + b_6y_P^2 \quad (2)$$

El objetivo es hallar los coeficientes $a_1 \dots a_6$ que son incógnitas en el sistema de ecuaciones. Para lograrlo es necesario utilizar n puntos de calibración, con $n \geq 6$, sobre el plano del objeto, considerados puntos conocidos en un sistema de coordenadas global. Llamemos a estos últimos $(X'_1, Y'_1, \dots, X'_n, Y'_n)$. Asimismo, se tienen las correspondientes coordenadas de los puntos de calibración en el sistema de coordenadas de video $(x_1, y_1, \dots, x_n, y_n)$.

Asumiendo que X_i es una aproximación de la coordenada X'_i , la suma de los cuadrados de las desviaciones de n puntos de calibración nos da la función de error de la coordenada X :

$$E(X) = \sum_{i=1}^n (X_i - X'_i)^2 \quad (3)$$

o

$$E(X) = \sum_{i=1}^n (a_1 + a_2 x_i + a_3 y_i + a_4 x_i^2 + a_5 x_i y_i + a_6 y_i^2 - X'_i)^2$$

Estas coordenadas pueden expresarse en forma matricial como $|X \cdot a - X'|^2$ donde:

$$X = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n^2 & x_n y_n & y_n^2 \end{bmatrix}$$

Las 6 incógnitas son determinadas al minimizar la función de error $E(X)$. Hay varias formas de resolver este sistema, una de ellas es igualando a cero la derivada de la matriz anterior, esto es:

$$\frac{\partial E(X)}{\partial a_j} = 0$$

Las condiciones obtenidas proveen 6 ecuaciones que son lineales en los coeficientes desconocidos y son expresadas matricialmente a través de la ecuación normal como:

$$(X^T \cdot X) \cdot a = X^T \cdot X'$$

La solución a este sistema, puede ser realizada utilizando métodos de álgebra lineal como por ejemplo eliminación por Gauss Jordan, sin embargo puede suceder que las matrices sean singulares, o cercanas a ello, por lo que es posible que estas ecuaciones tengan soluciones compatibles indeterminadas en términos algebraicos. En estos casos la técnica apropiada es la de Descomposición a Valores Singulares (SVD), que en lugar de usar las ecuaciones normales descompone la matriz X de tamaño $N \times M$ (con $N = n$ y $M = 6$) como el producto de una matriz ortogonal U $M \times N$, una matriz diagonal A $M \times M$ con números cero o positivos, y la traspuesta de una matriz ortogonal V $N \times N$. Esto es:

$$X = U \cdot \sigma \cdot V^T$$

donde:

$$U^T \cdot U = V^T \cdot V = 1 \quad \text{y} \quad \sigma = \text{diag} \{ \sigma_1 \dots \sigma_6 \}$$

Por lo tanto la solución del sistema original es el coeficiente a que minimiza $|X \cdot a - X'|^2$, el cual está dado por:

$$a = V \cdot \sigma^{-1} \cdot U^T \cdot X'$$

Ciertamente el sistema polinómico (1) es válido en una situación en que la distorsión es lineal en todo el campo de visión, sin embargo, la realidad es diferente y la distorsión funciona de manera tal que es creciente desde el centro hacia los extremos. Por lo tanto es necesario considerar esto en el método, para lo cual se introducen funciones de peso no negativas $w_i(x_p, y_p)$ para cada punto de calibración X_i, Y_i que dependen de la distancia relativa entre las coordenadas del video (x_i, y_i) de los puntos de calibración y las coordenadas de video x_p, y_p de todo punto en el plano del objeto. Al introducir una distancia entre puntos d , las funciones de peso quedan determinadas como:

$$w_i(x, y) = \begin{cases} 1, & \text{si } \{(x_p - x_i)^2 + (y_p - y_i)^2\}^{\frac{1}{2}} \leq d \\ 0, & \text{si } \{(x_p - x_i)^2 + (y_p - y_i)^2\}^{\frac{1}{2}} > d \end{cases} \quad (4)$$

Convirtiendo la función de error en:

$$E(X) = \sum_{i=1}^n w_i(X_i - X'_i)^2 \quad (6)$$

La distancia d depende del grado de distorsión de cada sistema de intensificación, idealmente debería obtenerse el valor que minimice las ecuaciones (4) y (5). En el *paper* se menciona que dicho valor mínimo se alcanza cuando d es tal que la cantidad de puntos de calibración tomados para corregir la distorsión cubre un 40% del campo de visión de la imagen.

De manera similar se sigue el mismo procedimiento para obtener las incógnitas b en el eje Y .

Luego de la presentación del método se tratan los aspectos relacionados a la mecánica de la rodilla. Posteriormente, tras extraer una serie de frames de la videofluoroscopia, se describe el procedimiento realizado para obtener los ya mencionados Punto de Contacto Tibio-Femoral (PCTF) y Brazo de Momento (BM), digitalizando para el primero los puntos que describen los cóndilos y el platillo tibial, y para el segundo los puntos ubicados en la rótula y la tuberosidad tibial.

2.3.2. Otros métodos de procesamiento de videofluoroscopia

La formulación del concepto de CINARTRO ya fue elaborada previamente en Simini y Santos [5] y posteriormente enriquecida en Santos et al. [28], donde se realiza un análisis

detallado de los aspectos clínicos y la especificación en términos generales de las características que deberá tener el software, hasta ese momento no desarrollado. Una de las cosas más interesantes a mencionar en dichos textos, es la conclusión de que hasta el momento no existe una herramienta que permita determinar cuantitativamente el valor de estos puntos durante el movimiento de la articulación y que pueda ser empleada en el uso diario. De aquí otro aporte a la motivación de crear un producto que sí lo haga.

Nos limitamos al estudio en detalle de aquellos textos que realizan un análisis de la cinemática en dos dimensiones, ya que el uso de la videofluoroscopia nos permite observar solamente un plano de visión. De todas formas es importante mencionar que existen diversos trabajos (Giphart et al. [10], Van de Velde y Bingham [29], Mahfouz et al. [30]), sobre cómo abordar los cálculos en un plano tridimensional. La principal ventaja está justificada en los términos de precisión y se debe a razones puramente anatómicas. Esto quiere decir que la forma de la rodilla no es simétrica en el eje sagital para ninguno de los huesos que la componen y tampoco así la traslación de estos en un movimiento de extensión. De aquí se deduce que el Punto de Contacto Tibio-Femoral (PCTF) se ubica con cierta profundidad, cercano al centro de la rodilla, y no en la parte más externa de los cóndilos.

La razón por la cual no se utilizan este tipo de análisis está principalmente relacionada a los costos. En general los equipos que permiten extraer imágenes en tres dimensiones, como resonancias magnéticas, no son utilizados para investigaciones. Otras limitantes se deben a que no están adaptados para la toma de videos ni para el movimiento de una articulación como la rodilla.

La videofluoroscopia utilizada por Baltzopoulos hace más de 20 años no deja de ser en nuestra región uno de los instrumentos más usados por especialistas médicos, por tener una relación de costo beneficio más que aceptable y que además permite realizar la toma de videos durante un movimiento complejo.

Lo cierto respecto a la disyuntiva de estudiar un escenario en dos o tres dimensiones es que el caso ideal está claramente en este último donde, por más que los cálculos se vuelvan mucho más complejos, se toman en consideración más parámetros de la realidad. Una alternativa aceptando el uso de la videofluoroscopia, es la toma del movimiento desde diferentes ángulos que permitan reconstruir la estructura tridimensional de la rodilla. Se podría pensar en realizar dos tomas con el fluoroscopia desde distintas perspectivas, pero esto genera el problema de que aumenta la dosis de radiación sobre el paciente, algo que claramente trata de ser evitado. Principalmente por estos motivos, se realiza la simplificación de utilizar un corte sagital de 2 dimensiones.

El concepto de Punto de Contacto Tibio-Femoral (PCTF) es introducido por varios autores como Baltzopoulos [7], Kellis [8], Wretenberg et al. [31], Scarvell et al. [32] y De Jong et al. [33], quienes valoran la importancia del cálculo de este punto, el cual resulta de gran interés en el estudio de la cinemática articular.

Cuando hablamos de calcular este punto el valor relevante es la cuantificación, en términos de porcentaje, respecto al eje marcado por la línea del platillo tibial, siendo 0 el extremo de la parte anterior del platillo. En un estudio bidimensional es considerado el cóndilo externo como referencia para los cálculos.

El PCTF es un valor fiel del estado de la rodilla, su ubicación a lo largo de un movimiento de extensión no es fijo ni se mueve en una línea recta sino que sigue una compleja trayectoria, reflejando el movimiento de la articulación de una rodilla, todo causado por el desplazamiento de la tibia con respecto al fémur.

En varios estudios se ha hecho referencia a un punto denominado Centro de Rotación Instantáneo (CRI) (Simini y Santos [5], Ehrig et al. [34], Moorehead et al. [35]) que es presentado con gran relevancia por sus características desde el punto de vista de la mecánica. En algunos casos se considera al PCTF como una forma de calcular el CRI (Soudan et al. [36], Simini y Santos [5]). En nuestro caso los consideramos como dos conceptos diferentes siendo el PCTF el utilizado por Baltzopoulos [7] y definido como hasta el momento, mientras que el cálculo del CRI también puede ser utilizado para el análisis de la cinemática y se obtiene realizando el método de Reuleaux para imágenes consecutivas de una videofluoroscopia (Soudan et al. [36]).

En el método de Reuleaux para el cálculo del CRI se toman como referencia 2 puntos del objeto en movimiento para 2 instantes de tiempo diferentes. Para el mismo punto en las 2 imágenes se traza la mediatriz del segmento que los une. Luego el CRI es la intersección de dichas mediatrices. En la Figura 6 se muestra lo mencionado.

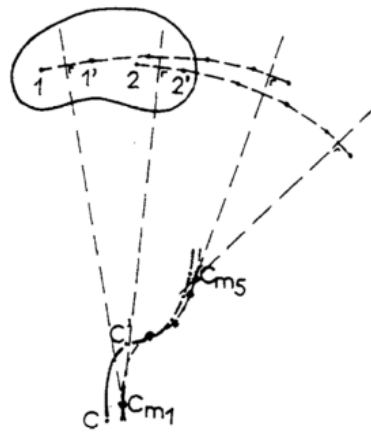


Figura 6: Construcción de Reuleaux.

C y C' indican los centros instantáneos en A y A'(posiciones).

Reuleaux ubica la intersección C_m para el movimiento AA'.

Los sucesivos C_{mi} permiten estimar el movimiento del CRI. Tomado de Soudan [36].

El Brazo de Momento (BM) funciona como otro de los parámetros más importantes, principalmente para determinar las fuerzas empleadas en el movimiento articular dentro de un modelo biomecánico (Kellis y Baltzopoulos [8]). Es otro de los parámetros de confianza, en el caso bidimensional, para el estudio de la cinemática y puede ser aplicado para computar la fuerza muscular durante pruebas de resistencia junto al uso de un dinamómetro.

En nuestro caso de estudio se toma ventaja de haber calculado el PCFT, utilizando éste como el punto de referencia de la recta perpendicular a la línea de acción de la fuerza, para así obtener el valor del Brazo de Momento.

En Kellis y Baltzopoulos [8] se trabaja sobre la corrección de la distorsión, retomando el cálculo de puntos de control dentro de un eje de coordenadas conocido (presentado por Baltzopoulos [7]), exponiendo mejores detalles sobre cómo llevar esto a la práctica. La propuesta allí es utilizar un fantoma para la determinación de la deformación de la imagen sobre diferentes zonas de la pantalla. El fantoma debe ser ubicado en la misma posición que estará el plano sagital de la rodilla posteriormente, generando tomas por videofluoroscopia del mismo. De esta forma, conocidas las distancias entre los puntos de la grilla, y ahora el resultado de esa grilla luego de experimentar la distorsión, es posible aplicar un cambio de coordenadas e identificar el nivel de distorsión que se sucede una vez que se obtienen las imágenes de la rodilla. Esta mejora dentro del método de corrección, es considerada también dentro de los parámetros de CINARTRO siendo una exigencia para mejores resultados contar con una imagen del fantoma desde la misma posición en la que se hace el movimiento de extensión.

Teniendo claro estos procedimientos, cómo se obtienen los parámetros de interés y la relevancia de los mismos, si nos adelantamos a la interpretación de resultados, se podría esperar que existan determinados estándares que permitan fácilmente, dado un conjunto de valores obtenidos en la serie de imágenes, identificar el estado de una rodilla. Sin embargo, esto no es tan trivial en la práctica. Cabe indicar que tras una investigación realizada no se encontró bibliografía donde se determine cuales son valores “normales” y valores fuera del rango de normalidad. De todos modos es posible obtener valores aproximados dentro de un análisis de resultados para un mismo paciente, al realizar varios estudios en el tiempo y así encontrar puntos de comparación. En la práctica, el especialista exige la videofluoroscopia de la rodilla afectada y la contralateral sana. De esta forma sus valores “normales” de referencia se encuentran en esta última, facilitando así la interpretación de resultados. Más detalles sobre el proceso de recuperación luego de una operación de LCA pueden encontrarse en Micheo et al. [26].

Se han visto las descripciones teóricas que muestran los aspectos más relevantes que determinan las bases de CINARTRO. Pese a que el pilar inicial está puesto en el estudio de Baltzopoulos [7], los demás textos contribuyen al entendimiento y mejora del planteo de la propuesta desde el punto de vista clínico, mecánico, de hardware y software. El resultado de comprender estos aspectos permite elaborar la especificación presentada en el siguiente capítulo.

Destacamos también el intercambio interdisciplinario junto a especialistas del área médica que permitieron avanzar en el conocimiento e interpretación de las publicaciones mencionadas y en forma conjunta comprender en detalle el significado de los aspectos más desconocidos y cómo las ideas podrían ser llevadas a la práctica.

2.4. Herramientas existentes

En cuanto al estado del arte desde el punto de vista del software, existen aplicaciones que de alguna forma u otra se relacionan a determinadas características de nuestra aplicación. Debajo se hace referencia a las más relevantes.

Nombre: CINARTRO: aplicación para la determinación del centro de rotación de la articulación de la rodilla y el seguimiento de los pacientes operados del ligamento cruzado [37].

Descripción: Actualmente, bajo el concepto de CINARTRO, existe un prototipo de aplicación desarrollado en el lenguaje Matlab [38] que plasma algunas de las características matemáticas del diseño teórico presentado en la parte anterior. Se propone la racionalización del conjunto de rutinas y el diseño y realización de un prototipo operativo para producir un producto de características compatibles con la práctica médica.

El gran inconveniente que presenta este software, es la gran cantidad de trabajo necesario para realizar el procedimiento de evaluación. El usuario debe marcar una enorme cantidad de puntos que termina resultando engorroso si se tienen muchas imágenes a procesar. Esta versión no está consolidada en un programa, sino que consta de varios módulos por separado que se deben ejecutar en una secuencia específica.

Nombre: Corrección de distorsión y detección de movimiento en Fluoroscopías de rodilla [39].

Descripción: Este proyecto, mediante su aplicación de software, logra describir el movimiento de la rotación de la tibia respecto al fémur a partir de una fluoroscopia. El usuario se encarga de marcar una serie de puntos de guía y el programa identifica patrones de sombra para las siguientes imágenes. Al igual que CINARTRO también realiza correcciones de distorsión en las imágenes, a través del uso de puntos de calibración. De este trabajo se tomaron algunas ideas que luego fueron implementadas en la aplicación de CINARTRO.

Nombre: ACL Rehab [40].

Descripción: Aplicación para celulares y *tablets*, disponible en AppStore, para la rehabilitación luego de una operación de ligamentos cruzados. Se enfoca en proveer ejercicios adecuados al usuario según las distintas etapas del proceso de rehabilitación. Ofrece además información sobre la anatomía de la rodilla.

Nombre: Knee XRayTo3D [41].

Descripción: Este sitio web brinda la funcionalidad de generar modelos tridimensionales de los huesos de una rodilla a partir de imágenes radiográficas. El usuario sube una imagen de rayos X, luego se selecciona qué parte de la rodilla quiere modelar y posteriormente marca un contorno sobre el borde a modo de guía. Posteriormente el programa crea la figura basándose en su forma plana y asumiendo su forma en el plano 3D.

Nombre: My Knee Guide [42].

Descripción: Aplicación móvil disponible en la AppStore que organiza el proceso pre y post operatorio de un paciente, enfocado específicamente a las lesiones de rodilla. Proporciona información acerca de la operación, ejercicios y recordatorios.

Capítulo 3

Especificación del software

Las características que definen la aplicación abordan aspectos que cubren diversas necesidades entre los que se destacan cuestiones técnicas, de usabilidad, de integración con componentes externos, de datos procesados y generados. Debajo se presentan las funcionalidades esperadas del software extendiendo la descripción del propósito general presentado en el Capítulo 1.

Plataforma y uso

Se debe construir una aplicación de tipo escritorio utilizando como base el lenguaje Java, la cual funcione en los principales sistemas operativos: Windows, Mac y Linux. Su uso debe ser posible sin una conexión a internet para la mayoría de sus utilidades.

Idiomas

El contenido del programa en cuanto a textos refiere, es importante para describir al usuario el significado de lo que puede observar en su pantalla. Al ser diseñada como una herramienta clínica, que eventualmente puede ser usada independientemente de la lengua del usuario, se apunta a contar con la opción de soportar múltiples idiomas seleccionables por el usuario. La manera de realizar la internacionalización debe permitir el ingreso de nuevos idiomas fácilmente.

Tipo de usuario objetivo

El público al cual apunta CINARTRO, como herramienta y como producto, son usuarios con perfil o conocimiento médico. Los términos utilizados dentro del programa que referencien a la anatomía deben ser palabras técnicas. De todos modos se deben proveer descripciones en aquellos casos que no sea claro el objetivo de una pantalla. Además la forma de navegación debe ser simple permitiendo seguir un flujo lo más intuitivamente posible.

Cálculos de interés

La aplicación de software debe ser capaz de resolver programáticamente los cálculos numéricos relacionados a la propuesta teórica original. Es decir que debe obtener los valores

de los parámetros de interés PCTF, BM y el ángulo en el que se encuentra la pierna del paciente en una imagen dada. Para lograrlo, se propone utilizar una interfaz donde el médico que opera la aplicación seleccione la cantidad de puntos suficientes que permitan inferir los valores para los parámetros de interés. A modo de enriquecer la experiencia del usuario, esta interfaz debe ser intuitiva y mostrar qué se está marcando y cómo se relacionan los puntos entre sí.

Agentes externos

Es de gran importancia la posibilidad de generar información que pueda ser compartida e intercambiada entre otros documentos médicos. Por lo tanto, la aplicación no puede estar aislada respecto a otras entidades que proporcionan puntos de entrada para la integración destinada al traspaso de información.

Cada evaluación realizada a partir del uso de la aplicación está asociada a una persona, cuya identidad es única cualquiera sea el lugar donde realice el estudio. Es aquí donde entra en juego la existencia del concepto de *Enterprise Master Patient Index* (EMPI) [43]. En términos generales, esta herramienta está ideada como estándar para el manejo de pacientes médicos, funcionando en cierto sentido como una base de datos, contando con su propio protocolo de intercambio de metadata. Por lo tanto es importante para CINARTRO utilizar la información de los pacientes de forma tal que éstos estén registrados bajo el sistema EMPI. Para ello, se debe hacer uso de los servicios ofrecidos por AGESIC [13] quien hace pública una interfaz para el manejo de individuos para ser accedidos desde cualquier origen, unificando la información de las personas.

El otro ítem propuesto dentro de los agentes externos, es la utilización de un repositorio XDS [12]. Este sitio de información es, en pocas palabras, el lugar donde un estudio clínico es compartido entre todas las entidades médicas. El estudio clínico es en realidad un reporte en formato CDA [11]. En base a esto, el software debe generar reportes en formato CDA siguiendo las reglas especificadas para este tipo de documentos y enviar los datos del estudio a un repositorio XDS.

Generación de información

El software debe manejar distintos tipos de información con el objetivo de enriquecer la historia clínica de un paciente, lo que incluye entre otras cosas: datos del paciente y el médico que realizó el estudio, imágenes, gráficas y metadata tales como fecha y lugar.

Este tipo de información debe ser obtenida de tres maneras:

- Utilizando un formato como por ejemplo PDF, que sea fácilmente legible y pueda ser impreso como registro personal de cada especialista.

- A través de la exportación de planillas Excel con la información de cálculos realizados, conteniendo valores de magnitudes y distancias, que podrían ser utilizados luego como comparación entre valores obtenidos.
- Mediante la creación de reportes en formato XML, lo cual se relaciona con el punto anterior, debido a que los reportes CDA están escritos en este formato.

Tiempo necesario para la realización de estudios

Exigir al usuario la ubicación de puntos sobre muchas imágenes demanda un tiempo considerable, por lo que es importante minimizar el tiempo requerido para realizar un estudio completo a un paciente. Se deben buscar alternativas que faciliten la usabilidad del software, disminuyendo el tiempo necesario para completar el procedimiento.

Precisión y corrección de errores

Llevar el error al mínimo posible es de gran relevancia para la aplicación. Se debe proveer al operario de CINARTRO con herramientas que permitan reducir el margen de error durante su interacción con las imágenes.

Por otro lado, considerando lo mencionado en el Capítulo 2 acerca de los problemas introducidos por la distorsión de las imágenes, internamente el software necesitará hacer los cálculos necesarios que permitan aplicar los algoritmos de corrección de la distorsión y presentar al usuario los resultados una vez aplicadas estas correcciones.

Capítulo 4

Interfaz y flujo del programa

A continuación se profundiza cómo procesar la información dentro de la aplicación, con el objetivo de permitir comprender el conjunto de pasos y etapas para la generación de resultados. Para complementar la descripción de las acciones, se utilizan algunas de las pantallas del software para facilitar el entendimiento de los diferentes pasos. Este capítulo puede ser utilizado como manual de uso de CINARTRO.

4.1. Procedimientos previos

Destaquemos primeramente que existen ciertas exigencias para el usuario al momento de utilizar el software. Una de ellas es contar con las imágenes que permiten visualizar la cinemática de la rodilla. Por lo tanto, la primer sección esclarece los procedimientos que se realizan para obtener las mismas.

El procedimiento de evaluación de un paciente mediante CINARTRO, comienza cuando éste es sometido a un estudio por videofluoroscopia. Para realizarlo, en el laboratorio clínico generalmente se utiliza un instrumento médico llamado Arco en C como se muestra en Figura 7. La estructura de este dispositivo permite libertad en el movimiento de la rodilla del paciente a diferencia de otros artefactos de la misma índole. El resultado de esta rutina es un video formado por la secuencia de imágenes obtenidas de los movimientos realizados.



Figura 7: Arco en C y equipo de videofluoroscopia.
Tomado de Siemens [44].

Respecto al procedimiento práctico, el operario debe asegurarse del cumplimiento de una serie de parámetros previo a realizar la captura del movimiento. Por ejemplo, la rodilla del paciente debe colocarse de tal forma que su plano sagital quede paralelo al intensificador, de lo contrario el video resultante no se verá con la calidad adecuada. Para lograr una buena calidad en las imágenes, no sólo es importante la calidad visual sino también la visibilidad de algunas estructuras para el fácil reconocimiento de las mismas. Como regla general, de acuerdo a lo descrito en Gigirey [45], se propone que las imágenes cumplan con las siguientes propiedades:

1. Los cóndilos externo e interno deben estar alineados en su parte anterior.
2. Se debe visualizar el espacio entre la rótula y los cóndilos.
3. Se debe visualizar el espacio entre el platillo tibial y los cóndilos.

Las Figuras 8 y 9 muestran la diferencia entre una toma que cumple las características y otra que no, respectivamente.

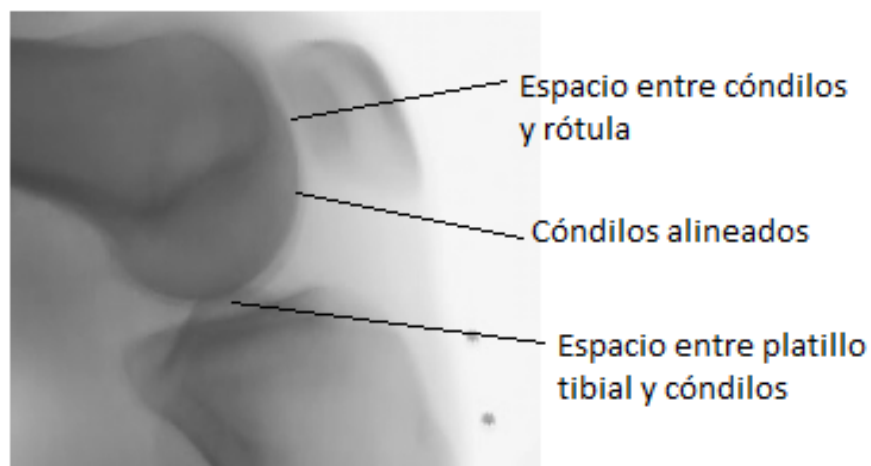


Figura 8: Radiografía que cumple con características de buena calidad.

La imagen cumple con las tres propiedades exigidas para ser considerada de buena calidad: cóndilos alineados, espacio entre cóndilos y rótula y espacio entre platillo tibial y cóndilos.

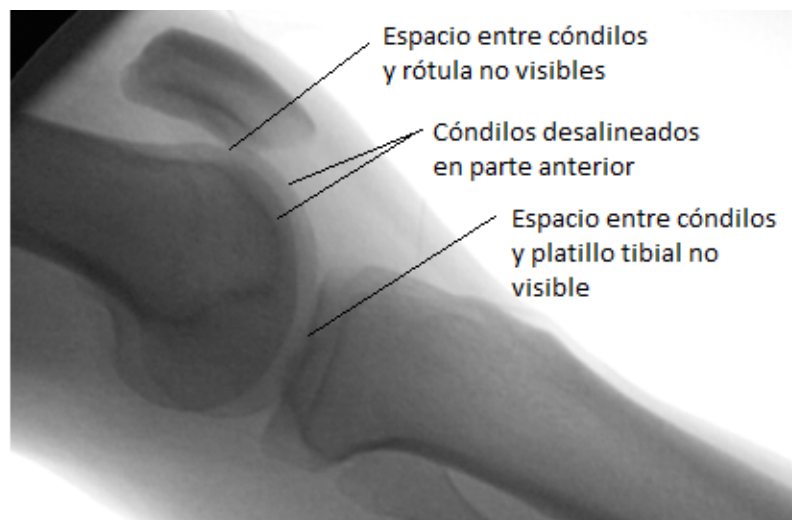


Figura 9: Radiografía considerada de mala calidad.

Si bien los colores son nítidos, no cumple con las características exigidas para ser considerada de buena calidad para un estudio. Algunas partes se superponen dificultando la identificación de la anatomía.

La otra parte del procedimiento incluye, de forma opcional, la captura del fantoma para la posterior corrección de distorsión. La exigencia aquí es que la toma se realice bajo las mismas circunstancias que el movimiento previo de rodilla, con el objetivo de mantener las distancias respecto al intensificador de imagen.

Una vez obtenidos los recursos necesarios para utilizar CINARTRO el usuario tiene la posibilidad de emplear directamente el video obtenido o las imágenes individuales (*frames*) como fuente de datos de entrada.

4.2. Descripción del flujo de trabajo

La idea aquí es presentar principalmente la interfaz de la aplicación realizada profundizando el detalle de las principales pantallas.

En primera instancia, el usuario deberá iniciar el programa CINARTRO. Una vez que se encuentra con el programa en ejecución, se le presentan varias opciones que podrá seleccionar cuyos detalles están en la Figura 10.



Figura 10: Vista inicial al ejecutar el programa CINARTRO.

A la izquierda se tiene el panel con el resumen de información del paciente y usuario actuales. En el centro el acceso directo a las principales funcionalidades.

Supongamos que el usuario se encuentra por primera vez ante CINARTRO y desea realizar un estudio para cierto paciente donde pueda observar las características de la anatomía de una rodilla. El flujo de uso comienza por el registro del paciente con el que desea trabajar. Para ello, el médico o usuario ingresa los datos del mismo completando un formulario. Luego de confirmar, la aplicación internamente inicia el proceso de comunicación con los servicios EMPI para corroborar los datos del usuario. Aquí pueden suceder dos cosas: o bien el usuario no existe en el sistema, con lo cual es ingresado a la plataforma EMPI como un nuevo individuo, o de lo contrario el usuario sí existe debido a que su nombre, apellido y número de identidad coinciden con otro perfil que fue previamente ingresado, en cuyo caso se le muestra al usuario esta coincidencia para que éste confirme que se trata de la misma persona. En el último caso, claro está, no se realiza el alta de un nuevo individuo. En la Figura 11 se refleja lo antes mencionado.

 Registro de paciente

*NOMBRE:

Ismael

*APELLIDO:

Alcoba

*CI:

48585859

F.NAC:

DIRECCION:

NUM TEL:

N° REGISTRO:

IMC:

Guardar

Estos pacientes coinciden con su búsqueda

Nombre	Apellido	CI	Direccion
Ismael	Alcoba	48585859	Bvar Artigas 1234

Seleccionar

Ninguno

(*) OBLIGATORIO

Figura 11: Pantalla para registro de pacientes.
Formulario para ingresar los datos del paciente. A la derecha se muestran los resultados obtenidos tras una búsqueda en el repositorio EMPI.

Ahora que en CINARTRO se tiene registrado cuál es el paciente con el que se está trabajando, el usuario procede a realizar la carga de recursos necesarios para la evaluación. El médico utiliza las imágenes obtenidas a partir del estudio práctico realizado con el Arco en C como fue mencionado. En caso que se seleccione un video, se le pregunta al usuario cuántas imágenes quiere obtener a partir de éste. Luego internamente se realiza un procesamiento del mismo y se extrae la cantidad seleccionada de imágenes de forma equidistante según la duración del video, lo que permite obtener una relación proporcional de grados para la posición de la rodilla entre los distintos *frames*.

Una vez finalizada esta configuración el usuario es redirigido a una nueva pantalla, como la que se muestra en Figura 12, donde se presenta una vista previa de las imágenes seleccionadas para trabajar. En esta instancia se le brinda al usuario la posibilidad de subir nuevas imágenes, eliminar existentes o de ordenarlas de forma diferente a la que se muestran. Con respecto a esto último, vale aclarar que el orden de las imágenes es importante en CINARTRO. Esto es debido a que los cálculos son realizados en el movimiento de extensión, donde el ángulo que podemos calcular, considerando los planos femoral y tibial, es creciente en todo momento, lo cual le da sentido a los gráficos tanto de ese mismo estudio como en la comparación con otros previamente realizados.

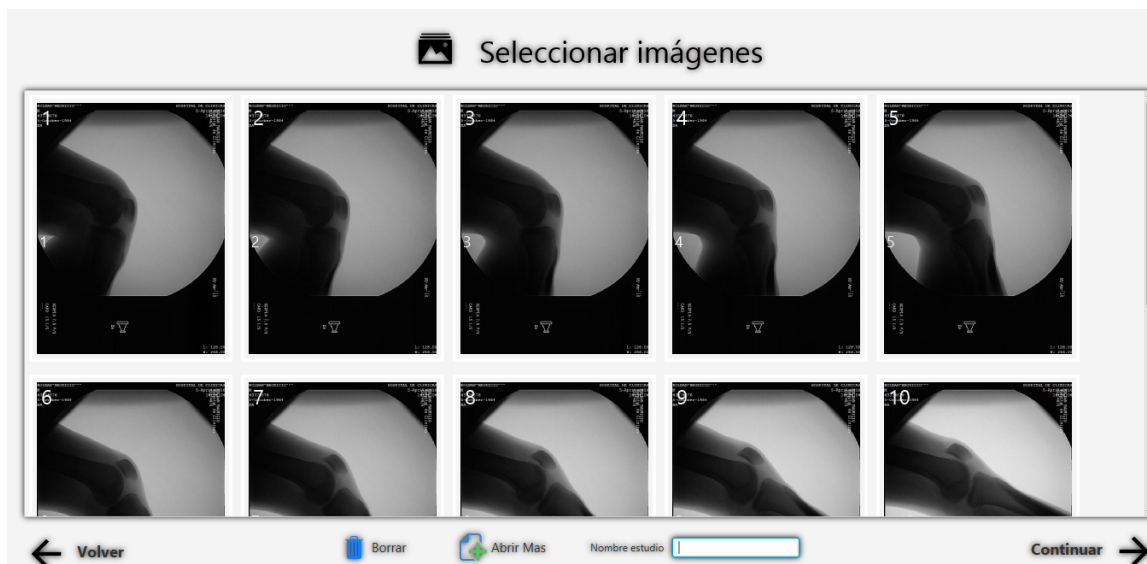


Figura 12: Vista previa de imágenes cargadas al programa.
El usuario puede modificar las imágenes de un estudio agregando nuevas o eliminando las que fueron cargadas previamente.

Una vez que el usuario confirma el conjunto de imágenes a considerar para el estudio, se prosigue con la siguiente etapa que se presenta como en Figura 13.

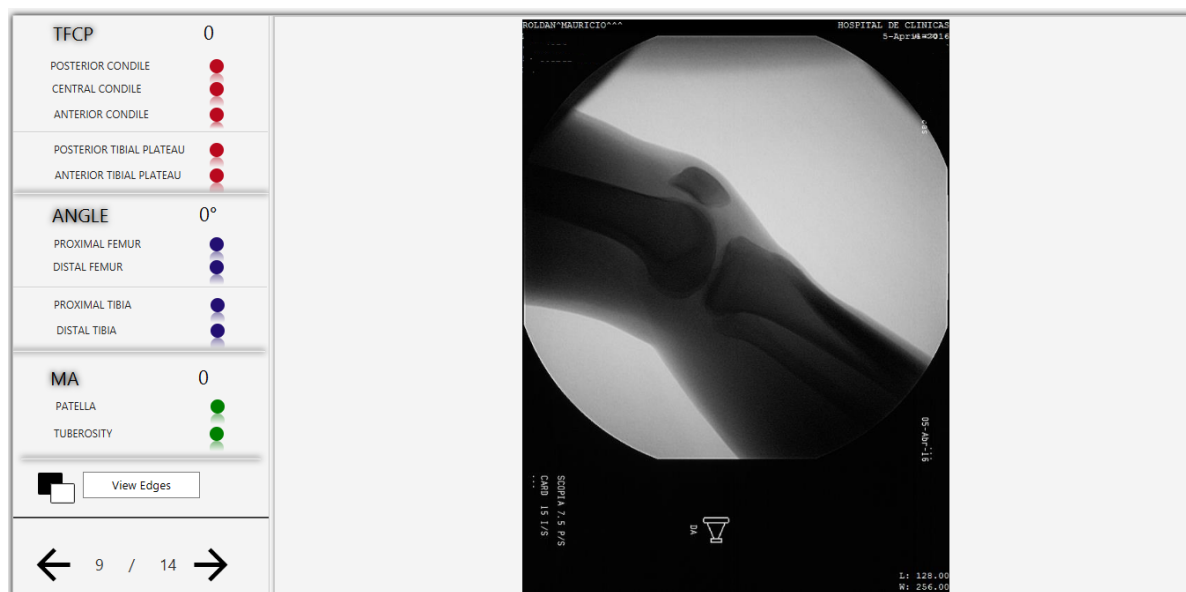


Figura 13: Pantalla para selección de puntos anatómicos de referencia.
El total de imágenes cargadas es presentadas individualmente para la ubicación de puntos.

Es aquí donde comienza lo que podríamos llamar el “trabajo de precisión” para el usuario. Cada imagen de las previamente seleccionadas, es mostrada en escena para que el médico

indique sobre la misma la posición de 11 puntos que luego serán utilizados en los cálculos. El objetivo en cada imagen procesada es obtener valores para:

- El Punto de Contacto Tibio-Femoral (PCTF)
- El Brazo de Momento (BM)
- El ángulo en ese instante, formado por la intersección de los planos femoral y tibial.

Cada uno de estos 11 puntos está referenciado en el panel izquierdo, distribuidos entre los subgrupos (PCTF, BM y el ángulo). El usuario deberá seleccionar cada uno de ellos y marcar su posición en la imagen. Cuando todos los puntos de un subgrupo son configurados, instantáneamente se muestra el valor calculado. En la Figura 14 se muestra un ejemplo de una imagen con todos los puntos ubicados sobre la imagen.

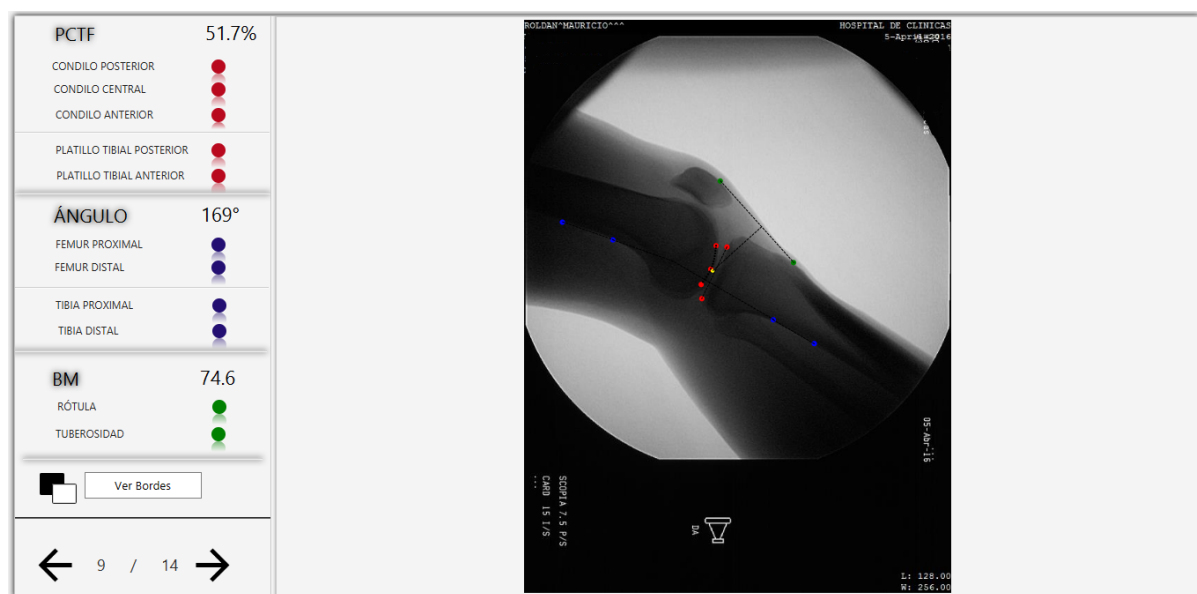


Figura 14: Pantalla con los puntos anatómicos ubicados por el usuario en la imagen.
Esta determinación debe hacerse en todas las imágenes del movimiento.

Finalizado esto, el médico ha completado una imagen y ya puede pasar a la siguiente. Al ser varias imágenes para un rango de movimiento tan corto, es esperable que de una imagen a otra el ángulo de la rodilla haya variado muy poco. Tomando esta consideración, el programa automáticamente muestra cuáles fueron las posiciones de los once puntos en la imagen anterior para que solo sea necesario ajustar levemente la nueva ubicación de los mismos.

Este procedimiento puede llevar varios minutos (en el Capítulo 7 veremos con mayor detalle los tiempos promedios calculados en las distintas pruebas), por ello es importante guardar el progreso de lo realizado hasta el momento, de forma que si se desea continuar con la marca de puntos en otro momento sea posible retomar desde una instancia previa. El usuario puede guardar el estado de avance desde el menú de control y de forma análoga cargar una instancia previa y continuar el trabajo.

Una vez finalizada la edición de todas las imágenes, es posible visualizar los resultados para el estudio. Se presentan gráficos como en Figura 15 para el PCTF y el BM, ambos mostrando

su valor respecto a los grados en los que se evaluaron, además de un gráfico para los ángulos respecto a la cantidad de imágenes.

Estas gráficas permiten visualizar rápidamente la variación de estos puntos a lo largo del movimiento de la pierna, ya sea la traslación del PCTF respecto al platillo tibial, o la magnitud alcanzada por el BM.

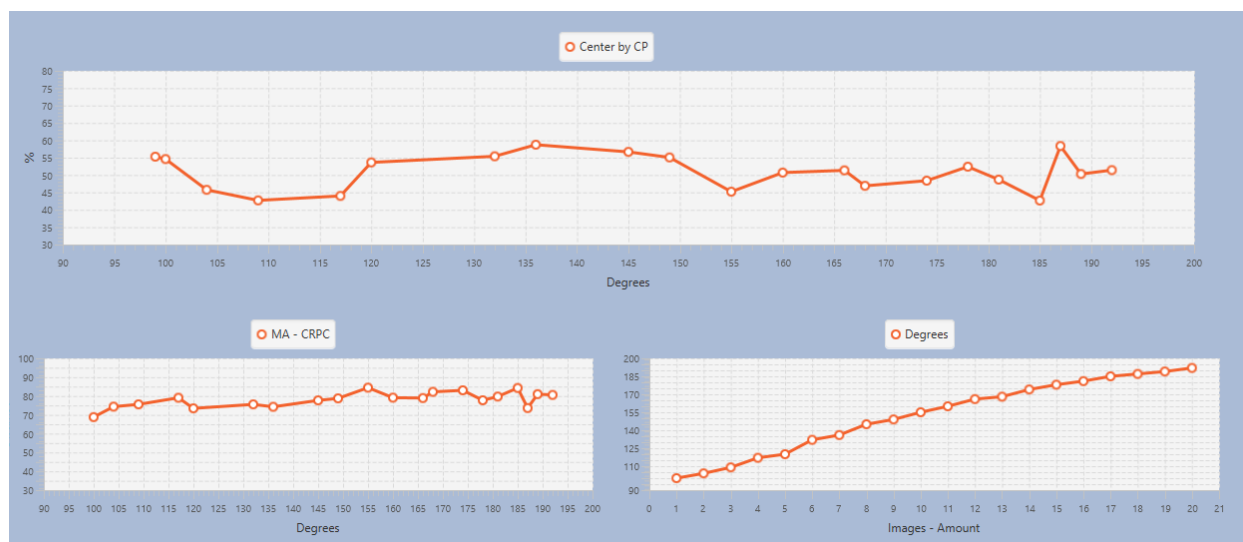


Figura 15: Gráficas presentadas por CINARTRO.

Arriba la ubicación del Punto de Contacto Tibio-Femoral (PCTF) respecto a la extremidad posterior del Platillo Tibial, expresada en % del platillo y, abajo, largo del Brazo de Momento (BM) del cuádriceps y correspondencia (a modo de verificación del regularidad del movimiento) entre ordinal de imagen y ángulo de la rotación de la rodilla.

Para resolver el problema de la distorsión *pincushion* provocado por el Arco en C (Baltzopoulos [7], Garling [46]), CINARTRO cuenta con la posibilidad de cargar una imagen de fantoma. Una vez cargada la imagen se deben seleccionar con un rectángulo los puntos del fantoma que van a ser utilizados para la corrección. El motivo de dicha selección es para ayudar al algoritmo a distinguir los puntos de plomo del fantoma del resto de las secciones de la imagen y para descartar aquellas filas que no estén completas. La Figura 16 muestra un ejemplo de imagen del fantoma desarrollado en Uruguay y que hemos utilizado.

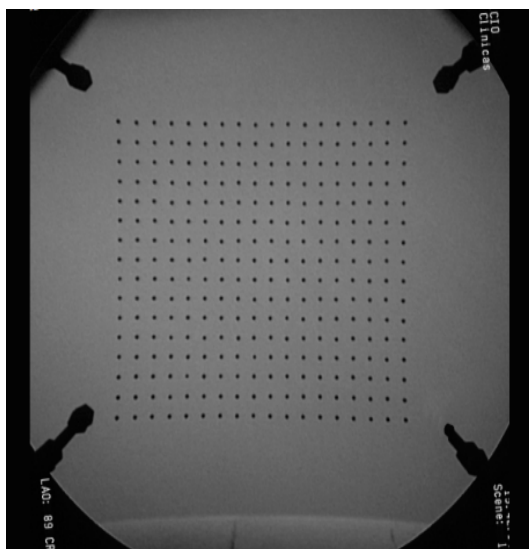


Figura 16: Imagen del fantoma.

Esta imagen es cargada para aplicar la corrección de la distorsión tipo “*pincushion*”. Radiografía tomada con un videofluoroscopio. Hospital de Clínicas, Montevideo, Uruguay. 2016.

Así finaliza un caso de uso que llamaríamos típico dentro de la aplicación CINARTRO, aunque es posible agregar más imágenes en caso de ser necesario que simplemente se acoplan junto con las anteriores para obtener los resultados antes descritos.

Sin embargo, las evaluaciones toman mayor sentido cuando son comparadas. El enriquecimiento de los gráficos y valores calculados se da en el momento de enfrentar resultados de dos instancias en el tratamiento y así plasmar las diferencias entre las mismas.

Supongamos entonces que, para cierto paciente, se han realizado varios estudios a través de CINARTRO. Sabiendo esto iniciamos la aplicación con otro objetivo que es el de comparar resultados y extraerlos fuera de la aplicación. Las opciones disponibles para exportar la información son PDF, Excel y CDA.

Por otro lado, para realizar la comparación de los resultados, se le pide al usuario que cargue hasta tres estudios previamente guardados (se consideró el máximo de 3 que es un número razonable de situaciones a comparar para un paciente). Paso seguido, el usuario selecciona la dirección de destino para los archivos PDF y CDA. El PDF contiene los datos más relevantes del caso, como son la información del médico, del paciente, tres imágenes que son extraídas de cada estudio (la primera, la última y una central) y la gráfica comparativa. Adicionalmente se agregan metadatos de la institución así como un espacio para la firma del médico.

Los reportes generados contienen información no sólo del paciente, sino también del médico que está realizando la tarea. Es probable que el programa, una vez que esté instalado en una máquina personal, sea usado por la misma institución y posiblemente por el mismo médico. CINARTRO permite configurar esta información como parte de las preferencias del usuario, la cual es recordada en una futura ejecución del programa como se muestra en la Figura 17. De esta forma los reportes generados contienen la metadata precargada sin que sea necesario ingresarla cada vez. Bajo esta línea de razonamiento, el idioma con el que se desee utilizar la aplicación, ver Figura 18, es otro factor poco variable por lo que también es otra de las preferencias en el perfil de la aplicación.

The screenshot shows a web application interface with two tabs at the top: 'Institute' (selected) and 'Language'. The main heading is 'Informacion Medica'. Below it, there are four input fields with labels on the left and values in the boxes: 'Nombre Institucion' with 'Hospital Clinicas', 'Nombre del Medico' with 'Jorge', 'Apellido del Medico' with 'Barovero', and 'Modelo Arco en C' with 'Philips Pulsera'. At the bottom left, there is a teal button labeled 'Guardar'.

Figura 17: Ingreso de la información institucional y del usuario.
Las preferencias de cada usuario son conservadas en un perfil personal para futuras sesiones de trabajo.

The screenshot shows the same web application interface but with the 'Language' tab selected. The main heading is 'Select Language'. Below it, there are two circular icons representing flags: the Spanish flag on the left and the British flag on the right. At the bottom center, there is a teal button labeled 'Save'.

Figura 18: Preferencias de idioma.
Al seleccionar un idioma, todos los textos de la pasan a estar en el idioma elegido.

Capítulo 5

Arquitectura y Diseño

El foco en este capítulo lo centramos en describir de qué forma se construye CINARTRO, basándonos en las características de arquitectura, bajo una visión global de las partes más importantes que forman el sistema.

5.1. Arquitectura del sistema

Anteriormente hemos mencionado los conceptos y elementos de la aplicación. La idea aquí no es centrarnos en la descripción de los mismos nuevamente, sino que es ligarlos entre sí y detallar cómo fluyen las comunicaciones. Introducimos en Figura 19 un diagrama con los principales componentes de la arquitectura y seguidamente el desarrollo de su descripción.

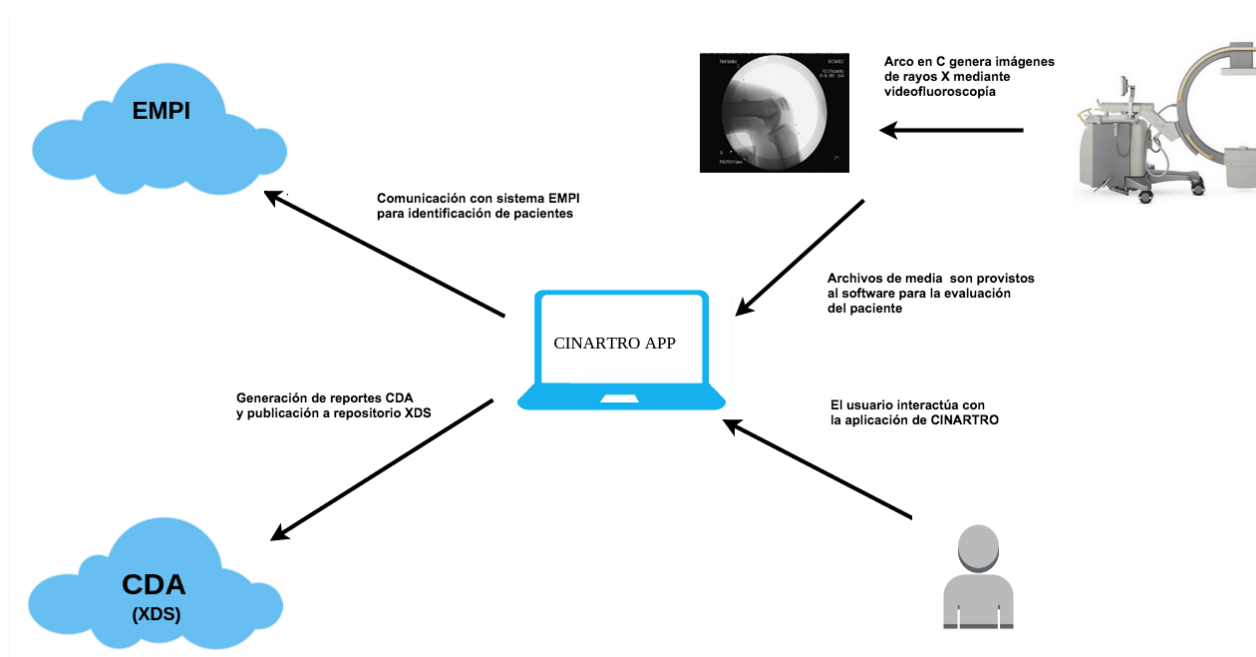


Figura 19: Componentes de la arquitectura de CINARTRO.

El ciclo dentro del sistema comienza con los procedimientos en el laboratorio médico realizados con el Arco en C, que dan como resultado archivos en formato video (secuencia de frames eventualmente transformados a extensión .AVI) que son los datos de entrada a la aplicación. El software tiene la característica de ser un programa de escritorio, el cual es distribuido como un instalador para ser utilizado en los computadores. Debido a que este tipo de programa es dependiente del sistema operativo instalado en la computadora, es necesario generar estos instaladores apropiadamente para cada caso.

El usuario o especialista médico es el siguiente factor en el sistema; es quien utiliza la aplicación, procesa las imágenes videofluoroscópicas y obtiene resultados que ayudan a la posterior evaluación del paciente.

El repositorio EMPI [15] es otro de los elementos que conforman CINARTRO, marcando la diferencia en la forma que se asocia la información clínica a la del paciente en cuestión, respecto a otros enfoques más simples que no hacen uso de los estándares de identificación de individuos, lo que constituye un aporte valioso. La conexión entre la información ingresada sobre el paciente y el registro en la plataforma EMPI, se produce dentro del software de forma transparente al usuario quien sólo necesitará ingresar los datos.

El programa establece la conexión con el repositorio siguiendo las reglas establecidas por el protocolo del servidor y lo hace con dos objetivos básicos que son el de buscar a un paciente y el de ingresar uno nuevo al índice. Cualquier otro sistema que consulte estos datos, hará referencia a los mismos individuos cooperando y enriqueciendo la historia clínica de una forma centralizada.

El último componente destacable de la arquitectura, son los reportes en formato CDA. Son el resultado de la interacción entre las partes mencionadas anteriormente, comprimiendo la información del paciente, del médico, e incluyendo los aspectos más relevantes de los estudios creados. Estos archivos con formato XML son generados bajo un estándar que especifica determinadas reglas y sintaxis que se deben respetar para que sea considerada como información clínica válida.

Tanto el repositorio EMPI como la generación de información en formato CDA son dos factores que hacen que CINARTRO no sea un software aislado, sino que lo unen a los canales de intercambio de información médica y permiten la integración con otras plataformas y sistemas; siendo ésta una de las características más resaltantes e interesantes. Con estas características CINARTRO queda habilitado para la interoperabilidad con todos los sistemas de informática médica del Uruguay, de acuerdo al plan de Salud.uy [14].

5.2. Diseño de la aplicación

El objetivo de esta sección es mostrar los aspectos relacionados al diseño del software. Para facilitar la explicación, la Figura 20 presenta un diagrama de clases simplificado que contiene las principales entidades del sistema. En el Apéndice G se presenta el conjunto de clases completo.

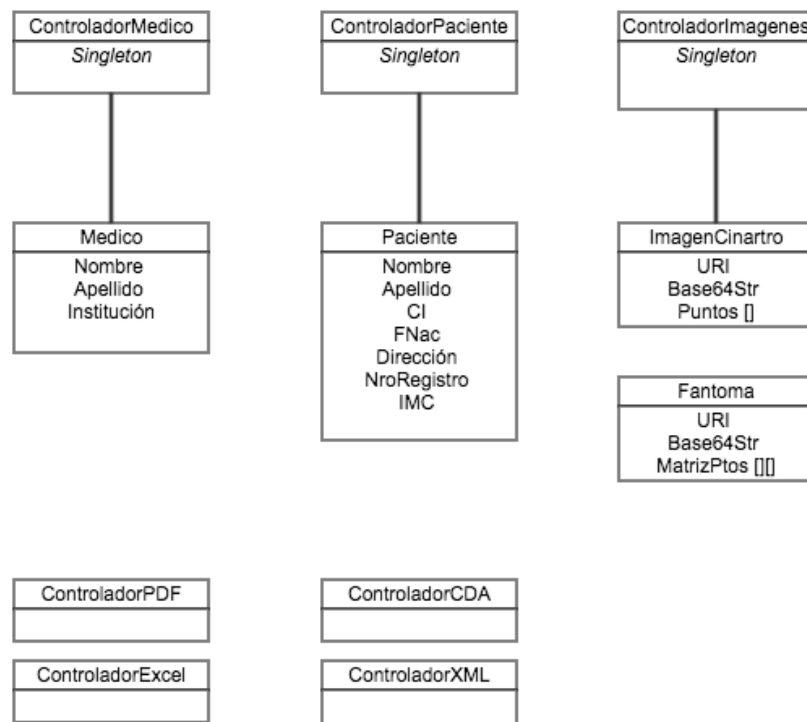


Figura 20: Diagrama de clases simplificado de CINARTRO.
Se detallan los principales conceptos del sistema y cómo se relacionan entre sí.

Podemos observar los conceptos de *Medico*, que representa una instancia del médico que hace uso de la aplicación, y de *Paciente*, que refiere a una instancia de paciente en un momento dado. El primero de ellos está conformado por el nombre y apellido del médico y por la información del instituto al cual pertenece. Esta información es utilizada principalmente para identificar al profesional en los estudios realizados cuando los reportes son descargados. En segundo lugar, la clase representativa del paciente guarda como elementos más relevantes: nombre y apellido, cédula de identidad, fecha de nacimiento, dirección, número de registro e índice de masa corporal, donde estos datos están sincronizados con los que existen en la plataforma EMPI. Ambos componentes están manejados por su respectivo controlador. Estos controladores, tienen la particularidad de que están implementados siguiendo el patrón de diseño *Singleton*, para que en todo momento, mientras la aplicación esté en ejecución, haya sólo una instancia de cada uno de estos elementos.

Otro de los componentes importantes del diseño, es el modelo de *ImagenCinartro*; que de la misma forma que los anteriores, es manejado por un controlador *Singleton*. La diferencia con los otros casos es que este controlador incluye una colección de instancias *ImagenCinartro* en lugar de solo un objeto.

Como es de esperarse, estas instancias guardan la información relacionada con las imágenes de un estudio en particular. Cada una de ellas contiene por ejemplo la ruta al archivo y los datos en formato byte64, pero más importante aún, cada una de las posiciones de los puntos marcados durante la edición. Así, una vez que se levanta un estudio existente, las imágenes y sus puntos son asignados a una instancia actual.

Cabe aclarar que la posición de cada uno de los once puntos en cada imagen es guardada mientras que los cálculos de ángulos, PCTF y BM son realizados dinámicamente en tiempo de ejecución, ya que tenemos las variables necesarias para hacerlo.

Por lo tanto habrá tantas instancias como imágenes sean subidas para un estudio, y de esto se deriva que sólo se trabaja con un estudio por vez. Cuando se desee comenzar con un nuevo conjunto de datos, la información de este primer estudio debe ser guardada.

Otra de las clases relevantes, es la de *Fantoma* que, además de los datos de imagen, maneja la matriz de valores necesaria para aplicar el método de corrección.

Finalmente, la aplicación cuenta con clases creadas para manejar los formatos de archivos y exportación de datos, como la creación de PDF, archivos Excel y archivos XML, éstos representados en el diagrama por *ControladorPDF*, *ControladorExcel* y *ControladorXML* respectivamente.

Capítulo 6

Implementación

Cambiamos el enfoque nuevamente para adentrarnos en un análisis a más bajo nivel. Nos centramos ahora, en las propiedades técnicas implicadas en la construcción de la aplicación de software, justificando las decisiones tomadas durante el desarrollo, desde un punto de vista de programación del sistema.

6.1. Herramientas utilizadas

Se presentan a continuación las herramientas de programación utilizadas que hacen a la aplicación funcionar en base a los objetivos planteados. Primeramente se profundiza en los aspectos más importantes y luego en la descripción de las restantes bibliotecas y *frameworks* aplicados.

6.1.1. Principales características

La Tabla 1 realiza una comparación de las principales propiedades de CINARTRO respecto a otro conjunto de opciones que se podrían tomar en cuenta para hacer el software. Seguidamente, se discute cada uno de los casos.

TABLA 1: ALTERNATIVAS A LAS PRINCIPALES CARACTERÍSTICAS DE CINARTRO

Característica	CINARTRO	Alternativas descartada
Tipo de aplicación/instalación	Escritorio/Standalone	Web/navegador, dispositivo móvil
Lenguaje de programación	Java	C++, C#, .NET
Librería de interfaz gráfica	JavaFX	Swing
Formatos de salida	XML, PDF, Excel	Customizado, txt

Tipo de aplicación

El software fue desarrollado como una aplicación de escritorio a diferencia de otras opciones como aplicaciones web o para dispositivos móviles. De esta manera CINARTRO no depende de servidores ni base de datos externos para manejar la información y, más importante aún, de una conexión a internet para su funcionamiento. Por otro lado los aspectos negativos de una aplicación de escritorio, se encuentran en cuanto a las actualizaciones siendo necesario que el usuario tenga que descargar la última versión y, en muchos casos, reinstalar el software.

Una aplicación móvil no resulta adecuada. Aunque provee accesibilidad, es casi imposible realizar acciones como la edición de los puntos en las imágenes en un espacio tan reducido de trabajo.

Para tomar la decisión se han priorizado principalmente dos factores. Uno es el hecho de que los médicos en los laboratorios no siempre cuentan con una conexión a internet. Y en segundo lugar se debe a que el acceso esté limitado a quienes tengan la aplicación instalada y no en un dominio totalmente público.

En cuanto al rendimiento, el hecho de que sea un software de escritorio permite aprovechar el hardware de la propia máquina, necesario para el renderizado y procesamiento de imágenes, procesamiento de video y cálculos matemáticos. Una versión web o móvil podría tener problemas de rendimiento debido a mala conexión con el servidor y por disponer por ejemplo de bajos recursos de hardware; estos son los posibles cuellos de botella para un buen funcionamiento de aplicación web.

Lenguajes de programación

El núcleo del programa está construido con el lenguaje Java [47], basado en el paradigma de orientación a objetos. Acerca de los lenguajes de programación, consideramos como alternativas a Java, opciones como C, .NET y C++, que aunque no son las únicas, igualmente son variantes comúnmente utilizadas para nuestro tipo de aplicación.

La preferencia de Java sobre C, se debe a que este último penetra en un nivel de abstracción muy bajo dentro de la programación, innecesario para los objetivos planteados. Además, al no ser un lenguaje orientado a objetos no se tienen conceptos de clases, objetos ni instancias, algo que se adapta muy bien a nuestro planteo.

Respecto a .NET, la razón aquí es el soporte y el funcionamiento multiplataforma. .NET es un lenguaje adaptado al desarrollo de programas bajo el sistema operativo Windows. Sabiendo que en nuestro caso el soporte multiplataforma es uno de los requerimientos, ésta fue otra opción descartada.

C++ es un excelente candidato y de hecho no hay fundamentos tan claros como en los otros casos para descartarlo. La diferencia en este caso se debe principalmente a dos elementos. Encontramos una mayoría relativa de ejemplos de implementación hechos en JAVA: como

librerías, hilos de discusión y otras evidencias en internet. En segundo lugar, nuestra experiencia personal incluye el trabajo en Java más que en C++.

Interfaz gráfica

Al tratarse de un software de escritorio, se utiliza un framework (también basado en Java) llamado JavaFX [48], que provee un amplio conjunto de utilidades que ayudan y simplifican al desarrollo de este tipo de programas. Principalmente da soporte a lo relacionado con la interfaz gráfica, facilitando la integración entre los datos manejados internamente y cómo estos son presentados al usuario. Otra característica importante de JavaFX es que permite compilar las aplicaciones de escritorio de dos maneras: generando un archivo de extensión *.jar*, que se ejecuta en la máquina virtual de Java, y compilando la aplicación para generar instaladores nativos a cada sistema operativo.

Una de las principales desventajas de JavaFX es el bajo soporte que existe en la Web, ya sea como documentación de las funciones, ejemplos y resolución de problemas específicos.

La diferencia con otras opciones como por ejemplo JavaSwing [49], es que JavaFX es una tecnología más actual, contando con componentes gráficos bastante más recientes, pero más importante aún es lo fácil e intuitivo que puede resultar utilizar JavaFX. El hecho de que tenga un enfoque *Model-View-Controller*, permite llevar a cabo un desarrollo más ordenado. Estas características hicieron que JavaFX fuera la decisión elegida para CINARTRO.

Formatos de salida

Otro de los aspectos relevantes entre las decisiones tomadas, está relacionado a los formatos de salida del programa. Cada formato se debe a un objetivo en particular, donde uno resulta más adecuado que los otros, y es por eso principalmente que se trabajan con múltiples tipos de archivos.

El caso de los archivos Excel cumple con el objetivo de obtener los datos crudos de la aplicación, esto es los valores con sus respectivas magnitudes de los parámetros de interés en nuestro sistema. Si bien inicialmente se manejó la posibilidad de persistir en un archivo de texto (.txt) o PDF, claramente la separación en celdas aporta una legibilidad mucho mayor para este tipo de información, dándole al clínico la posibilidad de profundizar en procesamiento y datos e investigaciones específicas.

Los informes generados para uso clínico del médico son realizados en formato PDF. Es un formato sumamente utilizado, e ideal para impresión y visualización en cualquier dispositivo. La norma nacional sobre Historia Clínica Electrónica y su interoperabilidad indica que los informes deban estar disponible en formato CDA y para este caso, como también se ha mencionado en el Capítulo 5, se utilizan archivos con tipo de datos XML. Además tomando ventaja de las bibliotecas incorporadas para esto, también se utiliza este formato XML para el guardado de datos internos de CINARTRO.

6.1.2. Otras herramientas

A continuación, la Tabla 2 muestra un conjunto de bibliotecas y *frameworks* empleados en la construcción del programa que son complementarios a los aspectos mencionados en la sección anterior. Posteriormente se realiza una breve descripción de cada uno.

TABLA 2: HERRAMIENTAS Y CONSIDERACIONES PARA EL DESARROLLO DE CINARTRO.

Característica	CINARTRO
Diseñador de interfaz	SceneBuilder
Identificación de pacientes	EMPI
Formatos de salida	XML, PDF, Excel
Biblioteca para creación de XML	JAXB
Biblioteca para creación de PDF	iTextPDF
Biblioteca para creación de Excel	No, CSV
Biblioteca para cálculos matemáticos	Apache Commons Math
Biblioteca para manipulación de videos	Xuggler
Biblioteca para envío de mensajes HTTP	HTTPClient

Uno de los mayores desafíos de CINARTRO fue diseñar la interfaz gráfica de tal forma que se incluyeran las funcionalidades exigidas y se mantuviera una interfaz amigable para el usuario. Aquí es donde entra en juego otra de las herramientas fundamentales utilizadas llamada SceneBuilder [50], la cual es creada especialmente para el desarrollo en JavaFX. Este programa oculta la complejidad de editar las propiedades de los datos directamente y en su lugar se diseña cómo lucen cada una de las pantallas a través de una interfaz donde se definen los estilos, tamaños, distancias entre los elementos, y cómo éstos se relacionan con los objetos Java de bajo nivel.

En cuanto a bibliotecas que fueron integradas a la aplicación, se destaca Apache Commons Math [51], que se emplea para ejecutar los cálculos matemáticos y principalmente la programación en cuanto a la geometría del espacio. En particular, se utiliza esta librería en CINARTRO para la resolución de Sistemas a Valores determinados (SVD) en el algoritmo

de corrección de distorsión y para la interpolación de los puntos sobre el borde del cóndilo como fue explicado en el Capítulo 2.

Uno de los problemas interesantes a resolver, era el de transformar archivos de video provenientes del estudio de videofluoroscopia en un conjunto de imágenes. Se integró para esto, una biblioteca llamada Xuggler [52] que ofrece una serie de funciones para descomprimir el archivo original en diferentes frames encapsulando utilidades de otra biblioteca adicional llamada FFmpeg [53]. Programáticamente se implementó la lógica para que el usuario tenga la posibilidad de seleccionar el número de imágenes a extraer del video y así obtener los frames exactamente distribuidos según la duración del video, como se describió en el Capítulo 4. Esto se logra con poco esfuerzo adicional con un cálculo de *duración/número de imágenes*, para identificar cada cuánto tiempo se debe separar una imagen del video.

Otra de las acciones realizadas internamente, es la comunicación con el repositorio EMPI para la consulta y transferencia de datos de los pacientes con el servidor remoto. La comunicación se sostiene sobre envío de mensajes https a través de internet. Aquí se utilizó otra biblioteca llamada HTTPClient [54], que ofrece una interfaz simplificada para este tipo de comunicaciones donde se configura el contenido de las peticiones, la encriptación de mensajes y los encabezados identificativos de la sesión abierta con el repositorio.

Para el caso de exportación de archivos en formato Excel no se incluyó ningún software externo, sino que fue suficiente tratarlo internamente como un objeto *CSV* en forma de *String* que simplemente se guarda con la extensión *.xls*.

Por otro lado, sí fue necesario incluir dependencias para la creación de archivos PDF y XML que fueron iText [55] y JAXB [56] respectivamente. El primero de ellos define ciertas clases que dan formato al PDF y permite agregar textos e imágenes en tiempo de ejecución. El segundo caso funciona de forma similar al PDF, pero tiene la particularidad de que es posible mapear el contenido del XML con objetos propios del programa lo que simplifica la escritura y lectura de estos archivos. Esta es la razón principal por la que XML es el formato de preferencia para persistir el progreso realizado durante el uso de CINARTRO.

De esta forma tenemos un panorama de las elecciones clave que se realizaron; si bien las decisiones de implementación se produjeron continuamente, este conjunto de componentes es la base del desarrollo del software. Cada una de las tecnologías fue seleccionada cuidadosamente, realizando un análisis comparativo (Tabla 1) en cada caso respecto a otras opciones, considerando además la validez en cuanto a la distribución como software libre (Creative Commons [57]). En general todas las bibliotecas así como nuestro lenguaje base Java, continúan hoy en día bajo mantenimiento y siendo soportados en todos los sistemas operativos, lo que justifica aún más la solidez de los componentes sobre los que se construye CINARTRO.

6.2. Aspectos de programación

El objetivo de esta sección es describir en detalle los aspectos y consideraciones de programación más importantes del desarrollo. Esto implica principalmente los conceptos

geométricos utilizados y su representación, así como la implementación de los algoritmos de procesamiento y corrección de distorsión.

6.2.1. Estructuras básicas

En primer lugar surge la necesidad de representar el concepto de un punto en el plano. Para ello se utiliza la clase `Point2D.Double` del paquete `java.awt.geom`, que permite representar puntos de 2 coordenadas en decimales. Además, provee otras funcionalidades útiles como por ejemplo el cálculo de la distancia entre 2 puntos.

Por otro lado, es necesario construir rectas y segmentos ya que la representación del platillo tibial, Brazo de Momento y ángulo de la rodilla en el plano, manejan dichos conceptos. Aquí se decide implementar una clase propia denominada `CinartroRect`, que se ajusta a las necesidades del caso. Vale aclarar que el concepto de `CinartroRect` no es exactamente la representación de una recta, sino que posee además los conceptos de punto inicial y punto final. Por lo tanto, internamente se almacena un segmento y la recta en la que está contenido. De esta forma es posible calcular, por ejemplo, la intersección de la prolongación de dos segmentos. Esta clase contiene la representación general de la recta ($Ax + By + C = 0$) y su forma explícita ($y = mx + n$); esta última con el objetivo de facilitar los cálculos.

Además de lo mencionado, la clase `CinartroRect` cuenta con las siguientes funcionalidades:

1. Evaluación de un punto: Dado un punto x y la recta r , se busca su imagen a partir de la ecuación explícita. Esto es: $r(x)$
2. Obtención de la recta perpendicular a otra por un punto: Dada un instancia de `CinartroRect` y un punto `Point2D`, se devuelve otra instancia de `CinartroRect`, perpendicular a la primera y que pasa por el punto dado.
3. Intersección: Dadas 2 instancias de `CinartroRect`, devuelve un punto `Point2D`, que es intersección de las 2 instancias tomadas como rectas.
4. Punto medio: Devuelve el punto medio del segmento que representa una instancia de `CinartroRect`.
5. Ángulo entre rectas: Dadas 2 instancias de `CinartroRect`, devuelve el ángulo en grados entre ellas tomando las representaciones de recta.

6.2.2. Cálculo de parámetros de interés

A partir de la representación de estructuras básicas, se crea la clase `CinartroCore` que brinda una interfaz de funcionalidades utilizadas por la capa gráfica, para la visualización de información, y las clases a cargo de la exportación de resultados. Dichas funcionalidades incluyen el cálculo de los elementos de interés para CINARTRO como son el PCTF, BM y el ángulo de flexión.

Punto de Contacto Tibio-Femoral

Como se menciona en las secciones 2.1 y 2.2, el PCTF es un punto que se calcula como el medio del segmento de distancia mínima entre el cóndilo externo y el platillo tibial.

Para la representación del borde del cóndilo, el usuario marca 3 puntos de guía sobre la imagen para los cuales `CinartroCore` realiza una interpolación *spline*. A partir de esta función, se dibuja la curva que pasa por dichos puntos permitiendo al usuario obtener un *feedback* en su pantalla. Para la generación de la interpolación, se utiliza la clase `SplineInterpolator` del paquete `Apache Commons Math`.

Por otro lado, para la representación del platillo tibial se utiliza una instancia de `CinartroRect` generada por dos puntos marcados por el usuario sobre el largo del platillo tibial.

Para calcular en qué valor del eje x la distancia entre el *spline* y la instancia de `CinartroRect` se hace mínima, se van tomando valores de x de forma incremental dentro del dominio del *spline* y para cada punto evaluado en x , se calcula la distancia entre dicho punto y la instancia de `CinartroRect`. Se toma como distancia mínima, la registrada en una iteración anterior al momento que la distancia comienza a crecer.

Una vez que se tiene en qué valor del eje x la distancia es mínima, se calcula el segmento comprendido entre el valor del *spline* evaluado en x y la instancia de `CinartroRect` que representa el platillo tibial. El PCTF es el punto medio de dicho segmento. La Figura 21 permite visualizar los distintos elementos.

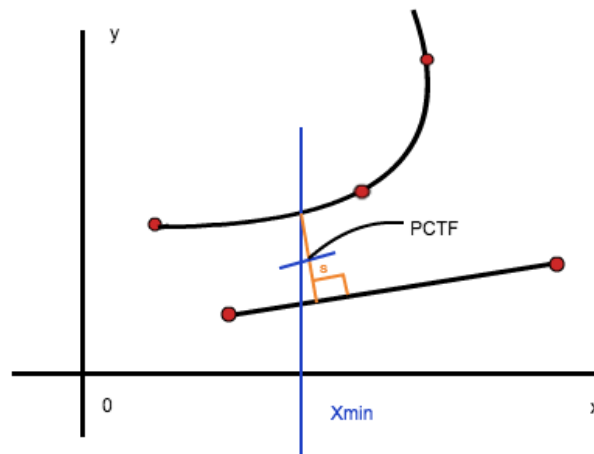


Figura 21: Representación del cálculo del Punto de Contacto Tibio Femoral (PCTF). Mediante 3 puntos en el cóndilo y las extremidades del platillo tibial definidos por el usuario, CINARTRO determina el PCTF como punto medio de la distancia entre cóndilo femoral y el platillo.

Brazo de Momento

Repasando el planteamiento teórico de las secciones 2.1 y 2.2, consideramos como Brazo de Momento a la distancia del PCTF al tendón rotuliano. Este último es representado por una instancia de la clase `CinartroRect` generada por 2 puntos marcados por el usuario, uno en el vértice inferior de la rótula y otro en la tuberosidad tibial. Para calcular el brazo de momento simplemente se calcula el largo del segmento que es perpendicular a la recta formada por el tendón rotuliano y la tuberosidad, y que pasa por el PCTF.

Ángulo de flexión

El ángulo de flexión es el ángulo en grados interno entre el fémur y la tibia. La pierna extendida completamente representará un ángulo cercano a 180 grados y a medida se flexione la rodilla, el ángulo irá disminuyendo. Para el cálculo del ángulo en un *frame* dado, el usuario marca 2 puntos a lo largo del borde del fémur y 2 puntos sobre el largo del borde de la tibia. A partir de dichos puntos se generan 2 instancias de `CinartroRect` y se calcula el ángulo entre ellas.

6.2.3. Digitalización del fantoma

Para la imagen del fantoma se necesitan obtener las coordenadas de los puntos de plomo en la imagen e identificar su correspondiente coordenada en la matriz de puntos real. En esta matriz real, se asocia la coordenada (0,0) al punto de plomo ubicado en el borde inferior izquierdo. De acuerdo a Baltzopoulos [7] para que el algoritmo de corrección funcione se necesitan a lo sumo 6 de estas coordenadas.

Para evitar que el usuario seleccione en la imagen del fantoma las coordenadas de los puntos, se implementa un procedimiento con el objetivo obtener dichos puntos automáticamente y de forma más precisa, reduciendo el esfuerzo del operador. Para realizarlo, el usuario simplemente debe marcar un rectángulo que encierre todos los puntos de la grilla con el fin de facilitar el procesamiento de la imagen.

El primer paso del procesamiento consiste en pasar todos los píxeles de la imagen a blanco y negro. Para cada píxel se transforma su color a HSB (Hue, Saturation, Brightness) y se evalúa la componente de brillo. Si dicha componente supera un umbral de brillo determinado, el píxel se pasa a blanco de lo contrario a negro. Tras realizar algunas pruebas, se utilizó un umbral de 0.2 para el cual se obtuvieron los mejores resultados. Finalmente, todos los píxeles exteriores al rectángulo definido por el usuario, son pasados a blanco. La Figura 22 muestra el resultado del pasaje a colores de la imagen original.

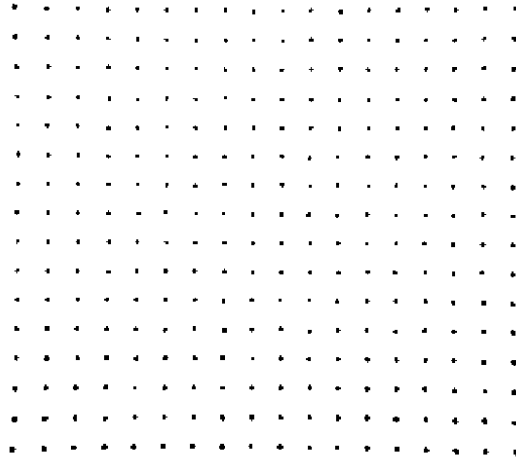


Figura 22: Reconocimiento de zonas grises en cada punto del fantoma.
Imagen obtenida por CINARTRO luego de aplicar el algoritmo de reconocimiento de puntos.

Como se dijo se desean obtener todas las coordenadas de los puntos del fantoma, pero dichos puntos se muestran en la imagen como una “nube de píxeles”. Por lo tanto en una segunda etapa se implementó un algoritmo que se encarga de recorrer los píxeles de la imagen y al encontrar uno negro, acumula todos los píxeles negros que sean contiguos para posteriormente calcular el promedio de los valores de las coordenadas. Entonces, para cada “nube de píxeles” se obtiene un punto que representa el centro de dicha nube. Si bien todos los píxeles se representan como enteros, los promedios hallados se almacenan como decimales para mayor precisión.

Hasta ahora tenemos las coordenadas de los puntos de calibración del fantoma en el sistema de coordenadas de la imagen. Para obtener los puntos en la grilla real en milímetros se toma el punto (0,0) y, conociendo la distancia entre los puntos de la grilla (en nuestro caso de 10 mm) se pueden mapear todos los puntos ya digitalizados.

Los mencionados algoritmos se implementan en una clase llamada `CinartroPhantomCore` donde la salida final es una instancia de otra clase `CinartroPhantomValues`, la cual contiene la matriz de coordenadas digitalizadas en el sistema de ejes de la imagen y la matriz de coordenadas de los puntos reales del fantoma en milímetros.

6.2.4. Corrección de puntos

En esta sección se detalla cómo se aplica el algoritmo de corrección de Baltzopoulos [7]. La entrada de dicho algoritmo consta de: el punto marcado en la imagen por el usuario y una instancia de la clase `CinartroPhantomValues` conteniendo los puntos del fantoma en el sistema de coordenadas de la imagen y en el sistema de la grilla en milímetros.

Como primer paso se toman los puntos del fantoma que se van a utilizar para la corrección, seleccionando aquellos que entren en un radio r con centro en el punto a corregir. Dicho radio es calculado de tal forma que la cantidad de puntos seleccionados sea el 40% del total. Este primer proceso representa la función de peso utilizada en [7], donde se toman puntos de

calibración cercanos al punto a corregir para que la distorsión en el fantoma y en el punto marcado sea de una magnitud comparable. En el Capítulo 8, se muestran los distintos valores de errores del algoritmo de corrección al variar la cantidad de puntos seleccionados.

A continuación se genera la matriz X , descrita en el Capítulo 2, a partir de los puntos del fantoma en coordenadas de la imagen. Con dicha matriz se construye una instancia de la clase `SingularValueDecomposition` de la biblioteca `Apache Commons Math`. A esta instancia se le pasa el vector X' con el cual se quiere resolver el sistema, donde X' se forma con las coordenadas reales de los puntos del fantoma. Como resultado del sistema, tenemos el vector a que minimiza el error de: $|X * a - X'|$. Con este vector resultante podemos construir una coordenada x de forma corregida y en milímetros a partir de un punto ingresado como entrada. De forma análoga, se construyen los valores de la coordenada y .

6.2.5. Implementación de tests

En lo que respecta a la implementación de tests, se le dio una mayor importancia a los cálculos geométricos ya que es uno de los pilares fundamentales de la aplicación. Para ello se utilizaron pruebas unitarias (JUnit) sobre la clase `CinartroCore` y `CinartroPhantomCore`, responsables de todos los cálculos geométricos.

Contar con pruebas unitarias facilitó la construcción de la aplicación ya que se podía verificar durante el desarrollo la correcta utilización de fórmulas al comparar resultados obtenidos con valores calculados fuera del software. Además las pruebas unitarias brindan la facilidad que ante algún cambio o mejora, podemos volver a ejecutar los test para verificar que no se introdujo ningún problema. Por otro lado permitió contemplar ciertos casos bordes que con la interfaz de usuario no es sencillo de reproducir.

Capítulo 7

Pruebas

Luego de finalizada la etapa de implementación, se da comienzo a un conjunto de pruebas sobre el sistema. El objetivo de las siguientes secciones, es explicar cómo se validaron los principales aspectos del software y comentar los procedimientos involucrados más importantes. Con mayor relevancia identificamos:

- El procedimiento para obtener las imágenes.
- El tiempo necesario para realizar los estudios.
- El error global del método de corrección de la distorsión.
- Pruebas realizadas con usuarios finales.
- Performance.
- Compatibilidad.

La idea aquí es exponer la forma en que las pruebas se realizan y cómo se resolvieron los problemas que surgieron a partir de las mismas.

7.1. Procedimiento médico con pacientes

Inicialmente se proporcionaron imágenes históricas de varios pacientes, extraídas de un estudio videofluoroscópico con un Arco en C, las cuales fueron utilizadas durante el desarrollo para probar la aplicación. Cada conjunto cuenta con aproximadamente 30 imágenes que describen el movimiento de extensión de una rodilla. A su vez, esta serie de imágenes fue tomada bajo 3 circunstancias para cada paciente: rodilla sana contralateral, rodilla con Ligamento Cruzado Anterior roto y rodilla reconstruida luego de la cirugía.

En segunda instancia, ya durante el transcurso del proyecto actual, el equipo participó del proceso de toma de imágenes de rayos X sobre pacientes, esta vez utilizando un angiógrafo. Vale aclarar que este dispositivo médico no es el equipo tradicional utilizado para una videofluoroscopia, sino que es incluso una tecnología más avanzada y con un costo de utilización mucho mayor. Sin embargo, para nuestros intereses proporciona el mismo resultado, en términos de imagenología, que la videofluoroscopia. En esta ocasión solo se tomaron imágenes sobre la rodilla lesionada y rodilla contralateral sana.

A la hora de realizar las tomas con el angiógrafo, surgieron algunos retos prácticos interesantes. En primer lugar el angiógrafo debió ser ajustado, con la ayuda de los técnicos

que suelen utilizar el dispositivo, para que la ubicación del paciente sea la adecuada. Los ajustes constan de: ajuste de rotación y altura de la camilla, ajuste de rotación y traslación del arco, y potencia del dispositivo para el estudio. Lograr que estos parámetros sean los correctos no fue una tarea sencilla ya que el angiógrafo es un dispositivo normalmente utilizado para procedimientos cardíacos, donde el paciente permanece acostado en la camilla. Una vez ajustado, ya durante las tomas de imágenes, surge un fenómeno durante el movimiento de extensión: el desarrollo muscular de cada paciente puede producir, en menor o mayor medida, rotaciones de la articulación durante el movimiento debido a la acción de dichos músculos. Esto hace que resulte difícil para el paciente realizar todo el rango de movimiento sin que se pierda el perfil de la rodilla requerido para el estudio, es decir que se mantenga siempre recta en un plano sagital.

7.2. Tiempo de realización de estudios

El objetivo de las pruebas, en este caso, es calificar el tiempo necesario para un usuario desde el momento que inicia la aplicación hasta que obtiene un reporte con los datos procesados para un paciente. Como era de esperarse, estas pruebas con datos reales permitieron descubrir varios errores de implementación e identificar mejoras en la usabilidad de la aplicación lo que derivó en una optimización considerable en el tiempo necesario para realizar un estudio. En la Tabla 3 se muestran los tiempos de las diferentes fases, donde entre cada una de ellas se probaron repetidas veces los casos de uso y se sucedieron arreglos y mejoras en el código del programa. Para presentar los resultados, consideramos la obtención de un reporte a partir de la comparación de 2 estudios. Se miden los tiempos de las tareas que más tiempos demandan del flujo, lo cual implica:

- Registrar al paciente
- Cargar las imágenes a la aplicación
- Ubicar puntos en un total de 60 imágenes
- Generar los documentos de salida

A modo de referencia, ubicar 11 puntos en una imagen demanda aproximadamente 45 segundos.

TABLA 3: OPTIMIZACIONES DEL PROGRAMA.

#Fase	Tiempo en minutos (luego de la mejora)	Problema detectado	Mejora realizada
1	75	Demoras al editar puntos marcados y al pasar de una imagen a la siguiente.	Se renderizan menor cantidad de puntos para el dibujado de las líneas en la interfaz.
2	70	Generar las gráficas de los puntos de interés consume excesiva cantidad de tiempo.	Optimización en cálculo del PCTF. Se aumenta el tamaño del paso iterativo utilizado para el cálculo de distancia mínima entre el cóndilo y el platillo tibial. (Inicialmente, se tenía un delta muy pequeño. Realizando pruebas se comprobó que aumentando el delta no cambiaban los resultados y se mejoraba la performance).
3	60	El usuario debe ingresar más de una vez información común a varios estudios.	Se persiste información como el perfil del médico, modelo del arco en C e institución para la cual trabaja. De esta forma, solo es necesario ingresarlo una vez.

7.3. Error global del método de corrección de la distorsión

La segunda categorización de nuestras pruebas son aquellas ligadas a la calibración de los procedimientos y fórmulas necesarias para realizar los distintos cálculos dentro del software. Entendemos por calibración a los elementos relacionados con el uso del fantoma en el proceso clínico. Como se ha discutido principalmente en los Capítulos 2 y 6, el procesamiento de las imágenes cuando son capturadas con el arco en C presenta distorsiones con respecto a una visión del mundo real, y utilizando el fantoma es que se logra la relación entre

coordenadas, lo cual permite corregir dicha distorsión utilizando unidades reales, como milímetros, en los resultados.

Antes de comenzar con la explicación se definen los siguientes conceptos:

Puntos de calibración: Puntos del fantoma digitalizados que son utilizados para aplicar el método de corrección de distorsión de Baltzopoulos [7].

Puntos de calibración cercanos: Utilizados para aplicar el método de Baltzopoulos [7] con el objetivo de corregir la distorsión de un punto dado. Estos puntos se toman dentro de un radio d y centro el punto a corregir.

Puntos de control: Puntos con coordenadas conocidas tanto en la imagen como en la grilla en milímetros. A estas coordenadas en la imagen se les aplica el algoritmo de corrección y se compara el resultado con las coordenadas en milímetros iniciales. Dicha comparación es el resultado del error global.

Esta prueba tiene como objetivo calcular el error global generado por el método utilizado para la corrección de distorsión de las imágenes [8] que se mostró en el Capítulo 2. Además, se busca un valor óptimo para la función de peso del algoritmo. Esta función busca utilizar sólo los puntos de calibración más cercanos al punto que se quiere corregir. Esto sirve para que la corrección de la distorsión se realice localmente ya que la misma no es uniforme en todo el campo de visión de la imagen. La función de peso se puede ajustar de forma que cambie la cantidad de puntos de calibración tomados.

Para el cálculo del error global, se utiliza la siguiente fórmula:

$$\bar{e} = m^{-1} \cdot \sum_{i=1}^m \{ (X_{Pi} - X'_{Pi})^2 + (Y_{Pi} - Y'_{Pi})^2 \}^{1/2}$$

Donde m es el número de puntos de control que se toman para el análisis del error. Para cada uno de los puntos, se toman sus coordenadas en el sistema de ejes de la imagen y se aplica el algoritmo de corrección. El resultado es comparado con las coordenadas conocidas en el sistema de ejes en milímetros. A diferencia del método de Baltzopoulos [7] se tiene la digitalización de todos los puntos del fantoma, por lo tanto se toman todos los puntos como puntos de control. Para las pruebas realizadas, se utilizaron un total de 234.

Por otro lado, X_{Pi} e Y_{Pi} representan las coordenadas del punto de control p en el sistema de coordenadas de la grilla en milímetros, donde el (0,0) se tomó como el punto ubicado en el borde inferior izquierdo del fantoma. Debido a que las distancias entre los puntos del fantoma son conocidas, se deducen las coordenadas para cada punto de control.

Finalmente X'_{Pi} e Y'_{Pi} son las coordenadas del punto de control p estimadas a partir del algoritmo de corrección de la distorsión. Dichos valores se obtienen aplicando el algoritmo de corrección a los 234 puntos digitalizados de la imagen del fantoma.

Mediante la Tabla 4, vemos los resultados obtenidos al variar el conjunto de puntos de control. Para el último valor se emplean 6 puntos, el equivale a un 2.5%, que es el mínimo número de puntos necesarios para aplicar el algoritmo.

TABLA 4: ERROR GLOBAL DE LA CORRECCIÓN SEGÚN NÚMERO DE PUNTOS DE CONTROL.

Puntos de control (%)	Error de la corrección
100	0.727
90	0.540
80	0.440
70	0.363
60	0.307
50	0.264
40	0.235
30	0.217
10	0.186
5	0.182
2.5	0.074

Observamos que al disminuir la cantidad de puntos de calibración utilizados, el error se hace más pequeño. Sin embargo si utilizamos un bajo porcentaje de puntos cercanos al punto a corregir, aquellos que queden fuera del rectángulo definido por los puntos de calibración presentan un error muy alto. Luego de varias pruebas heurísticas se determinó que con un 40% se minimiza el error global para la corrección de puntos lejanos a los puntos de calibración; además este porcentaje coincide con el utilizado por Baltzopoulos [7]. Por lo tanto, se decidió utilizar un 40% de los puntos de calibración cercanos al punto a corregir para realizar los cálculos de corrección de la distorsión.

7.4. Pruebas con usuarios finales

Otra de las formas en que la aplicación fue probada se dio con el uso de la misma por los usuarios finales. A lo largo del desarrollo hubo dos oportunidades en que distintas personas utilizaron el programa. Un total aproximado de 20 usuarios, la mayoría de ellos relacionados al área de la medicina, interactuaron con la aplicación dando una devolución acerca de su impresión y experiencia. Estas pruebas permitieron detectar algunos errores en el flujo del

programa y tomar nota de ciertas mejoras en la interfaz de usuario. Además se hicieron propuestas de nuevas funcionalidades que mejorarían la calidad del producto.

7.5. Performance

Con el crecimiento constante del programa, en términos del número de pantallas, carga de datos y lógica utilizada para resolver diversas funcionalidades, el uso de los recursos de la computadora que aloja a la aplicación empezó a ser un factor considerable. Se detectó que tras largos periodos de utilización del programa, la performance descendía funcionando cada vez con mayor lentitud y consumiendo una excesiva cantidad de memoria del sistema.

Para atacar estos problemas se utilizó un programa llamado Virtual VM [58], el cual permite monitorear la actividad de ciertos recursos de hardware. Puede ser integrado a Eclipse para acoplarse a la ejecución de un programa y observar las fluctuaciones en el uso de: CPU, memoria, clases generadas y *threads* creados, en el momento que se interactúa con la aplicación.

Durante la etapa de análisis se detectan problemas relacionados al alto consumo de memoria del sistema y de CPU, que deterioran la usabilidad del programa la cual se ve afectada tras el continuo aumento de estas métricas. En la Figura 23 podemos visualizar una gráfica extraída de Virtual VM sobre el control de memoria respecto del tiempo. La gráfica naranja representa el tamaño de memoria provisto por el Sistema Operativo mientras que la gráfica celeste es la memoria utilizada por el programa. Los picos que se observan se producen al recalcular los parámetros de interés, por ejemplo al pasar de una imagen a otra o editar un punto ya marcado.

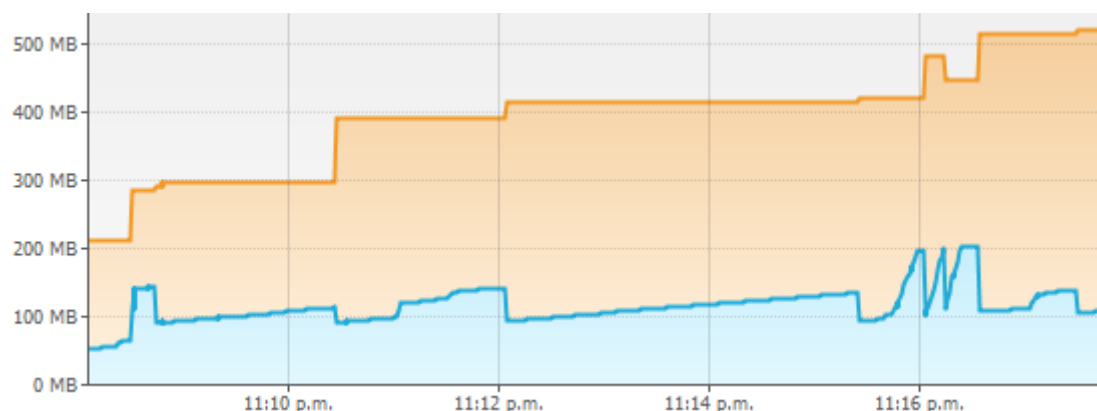


Figura 23: Uso de memoria previo a optimización.

La imagen muestra en naranja la memoria proporcionada por el Sistema Operativo. La gráfica celeste muestra la memoria utilizada por CINARTRO.

Bajo la misma situación se analizaron los tiempos necesarios para generar las gráficas, una de las acciones que sucedía con mayor retraso para el usuario, y se contabilizaron aproximadamente un total de 30 segundos en procesar los gráficos de un estudio de 30 imágenes.

Luego de comprender y acotar las causas de los problemas de performance, se introdujeron una serie de mejoras en el programa. Primero, con el objetivo de reducir los excesivos aumentos de utilización de memoria, se buscó reducir al mínimo las estructuras mantenidas en memoria eliminando aquellas referencias a estructuras que no se requerían para que el *garbage collector* las eliminara. Para este proceso, Virtual VM fue de gran ayuda.

Por otro lado, se buscaron optimizaciones dentro el propio código del programa en funciones que pudiesen exigir en demasía a los recursos del sistema.

Uno de los problemas detectados es a nivel de interfaz, cuando es dibujado el *spline* que une los puntos marcados sobre el cóndilo de la rodilla en la radiografía. Allí se redujo el número de puntos necesarios para el trazado y con esto el número de iteraciones necesarias para proveer las coordenadas a la interfaz.

El tercer elemento modificado, también relacionado a aspectos de implementación, fue la mejora en los cálculos del PCTF. Estos cálculos son realizados a través de una serie de funciones utilizadas constantemente en donde se aplicaron optimizaciones que simplificaron el algoritmo y redujeron el uso de CPU.

Tras aplicar estos cambios se lograron mejoras respecto a la situación inicial obteniendo resultados como se muestra en Figura 24, donde podemos ver que los picos ya no se producen de forma tan clara en los puntos críticos de uso de memoria.

Adicionalmente el cálculo del PCTF resultó ser un cuello de botella al generar las gráficas, por lo que luego de aplicar las optimizaciones mencionadas se redujeron los tiempos hasta aproximadamente 5 segundos para un estudio de 30 imágenes.

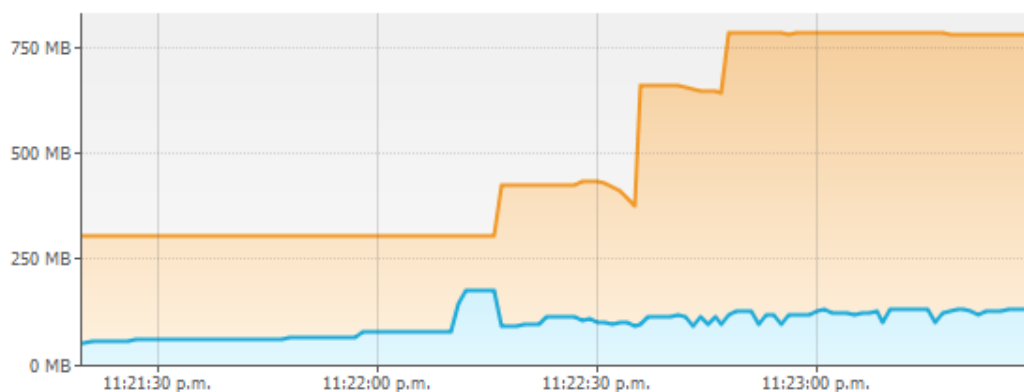


Figura 24: Uso de memoria posterior a optimización.

Los picos de la gráfica celeste (memoria utilizada por CINARTRO) ya no se producen con tanta notoriedad.

Los problemas de performance en la experiencia de usuario al utilizar la aplicación, llevaron a realizar pruebas en diferentes computadoras con distintas características de hardware. Esto permitió definir un conjunto de exigencias mínimas para tener una experiencia aceptable con la aplicación, de lo contrario el programa funcionará más lento de lo debido. La Tabla 5 propone los requerimientos mínimos de hardware para utilizar el programa.

TABLA 5: REQUERIMIENTOS DE HARDWARE MÍNIMOS

Hardware	Mínimo exigido
Memoria RAM	2 GB
Procesador	1.8 GHz
Disco Duro	1 GB libre

7.6. Compatibilidad

Finalmente la última categoría de pruebas incluye el uso de la aplicación en diferentes computadoras que realicen la ejecución y así testear la compatibilidad con los sistemas operativos. Se generan instaladores específicos para los 3 principales sistemas de uso común en el mercado: Windows, Mac y Linux.

El equipo efectuó todo el desarrollo del programa utilizando máquinas Windows, en particular trabajando con las versiones 7, 8 y 10. Además fue probado con las ediciones Vista y XP. La aplicación funcionó correctamente en todos los casos.

En segundo lugar, se crearon instaladores para dispositivos con sistema operativo Mac. Se probó la aplicación en dos computadoras, una de ellas una Mac Mini con versión OSX Yosemite, y el otro caso fue una Macbook Pro con la versión OSX El Capitán.

Tras realizar ciertas pruebas, se descubrió que en ambos casos, el sistema operativo inicia el instalador utilizando Java, por lo que es necesario, que exista previamente instalado en el sistema una versión Java que como mínimo debe ser 8.

Por último, CINARTRO fue probado en Linux. Si bien existen muchas variantes de este sistema se tomaron 2 opciones de las distribuciones más utilizadas: Ubuntu y Mint. De la misma forma que sucedió en las pruebas sobre computadores Mac, para accionar el ejecutable el sistema operativo exige tener instalado Java o de lo contrario surgirán errores al iniciar el programa.

Luego de un tiempo de investigación se hallaron fundamentos de que el problema sucede en todos los sistemas Unix por lo que es una exigencia contar con la versión mencionada de Java o de lo contrario el programa no funcionará. Esto se debe a la compatibilidad directamente relacionada a la librería de JavaFX. Una investigación al respecto permitió concluir que desde 2006 Linux viene por defecto con OpenJDK[59] , versión libre de Java, por lo que para los equipos más recientes la aplicación puede ejecutarse sin instalar otro software previamente.

La Tabla 6 resume lo mencionado respecto a las pruebas en los distintos sistemas operativos y sus versiones.

TABLA 6: COMPATIBILIDAD CON SISTEMAS OPERATIVOS.

Sistema Operativo	Versión	Resultado
Windows	7	Correcto
Windows	8	Correcto
Windows	10	Correcto
Windows	Vista	Correcto.
Windows	XP	Correcto.
Mac	OSX Yosemite	Correcto, pero es necesario tener como mínimo Java 8 instalado.
Mac	OSX El Capitán	Correcto, pero es necesario tener como mínimo Java 8 instalado.
Linux	Ubuntu	Correcto, pero es necesario tener como mínimo Java 8 instalado.
Linux	Mint	Correcto, pero es necesario tener como mínimo Java 8 instalado.

Capítulo 8

Resultados

Presentamos a continuación los resultados obtenidos una vez culminada la implementación y pruebas del software. Primeramente se analizan los resultados desde el punto de vista del programa y las funcionalidades que se lograron implementar. Luego se muestran los resultados obtenidos para el cálculo del Punto de Contacto Tibio-Femoral (PCTF) y el Brazo de Momento (BM) en base a las pruebas realizadas con pacientes. Finalmente se exponen los resultados de los métodos de corrección.

8.1. Resultados obtenidos tras la implementación del software

Para abordar los resultados obtenidos en el programa, analicemos las metas trazadas en la especificación. Los primeros ítems destacan la construcción de una aplicación de escritorio multiplataforma en el lenguaje Java. Esto pudo ser logrado principalmente a través de la biblioteca JavaFX que proporcionó el motor para la construcción de este tipo de software. Además, instaladores compatibles para los principales sistemas operativos permiten considerarlo como un programa multiplataforma.

Acerca de la internacionalización del software, se implementó una solución que permite al usuario la selección del idioma por defecto dentro de la aplicación. Las opciones existentes son Inglés y Español y si bien deberían ser agregados más idiomas en el futuro para mejorar la calidad del software, la forma en que fue resuelto permite agregar de forma simple nuevos idiomas a la colección.

La siguiente característica especificada apunta al usuario final. Dentro del flujo del programa se encuentran textos con terminologías y conceptos técnicos. La correctitud de los mismos fue validada con especialistas del área. Acerca de la amigabilidad y facilidad en el uso, se crearon barra de menús con las opciones accesibles en todo momento. Para aquellas pantallas con interacciones más complejas, se agregaron descripciones que detallan las acciones esperadas.

Respecto a los cálculos del PCTF, BM y ángulo, fue resuelto mediante una interfaz donde el médico realiza la selección de puntos que luego son procesados internamente para completar la lógica necesaria para calcular los valores de cada parámetro. En el Capítulo 4 se mostraron los pasos necesarios para realizarlo, mientras que en el Capítulo 6 cómo fue resuelto desde la lógica.

El manejo de usuarios y reportes son otras de las metas planteadas. Mediante el sistema EMPI se administra la información de pacientes analizados con CINARTRO. Se logró la comunicación a través del envío de mensajes mediante el protocolo REST y utilizando la API provista. A efectos de cumplir con una primera versión de estudio, se utilizaron sistemas EMPI de prueba que no fueron llevados a un caso real ya que se necesitaría un mayor tiempo de desarrollo y testing.

Cerca del final de la etapa de implementación los estudios clínicos fueron subidos al repositorio XDS, que al igual que los pacientes en EMPI, se emplearon ambientes de prueba.

En cuanto a la generación de información, CINARTRO permite obtener desde el software reportes en formato Excel y también reportes en formato PDF con información y comparación de estudios diferentes. Para la creación de reportes CDA, se crean archivos XML conteniendo la metadata especificada para este tipo de documentos.

Luego de analizar cada caso, es real decir que ha sido posible satisfacer cada una de las propuestas establecidas dentro del alcance. Si bien algunas cosas están sujetas a mejoras o crecimiento, consideramos que se logran los objetivos planteados inicialmente respecto a la aplicación y sus funcionalidades.

8.2. Resultados de los cálculos de interés

El segundo tipo de resultados se enmarca en el punto de vista de la investigación, basándose en un análisis respecto a las pautas teóricas planteadas en el Capítulo 2. Este enfoque propone la determinación del denominado Punto de Contacto Tibio-Femoral (PCTF) y del Brazo de Momento (BM) a través de imágenes radiográficas extraídas de una videofluoroscopia.

Recordemos que el PCTF es calculado como el porcentaje de desplazamiento respecto al platillo tibial. El BM por su parte se cuantifica como una distancia, la cual se calcula como la intersección entre el PCTF y la perpendicular a la línea de acción de fuerza determinada entre la parte baja de la rótula y la tuberosidad tibial.

Para establecer los resultados consideramos las pruebas realizadas sobre la etapa final en base al estudio de 3 pacientes con imágenes de su rodilla sana y lesionada. Se utilizó el fantoma correspondiente para realizar la corrección de distorsión. Las Tablas 7 y 8 muestran los valores obtenidos para el PCTF y BM respectivamente. Allí entendemos por media de la migración, tanto para el PCTF como para el BM, al promedio de los valores hallados en el total de imágenes. Cabe destacar que el sentido de hallar los resultados de esta forma es obtener una manera simple de comparar el estado de una rodilla sana y una lesionada. Además es la forma en que se realiza en los *papers* de referencia Baltzopoulos [7], Kellis y Baltzopoulos [8], Simini y Santos [5]. La interpretación de estos resultados desde un punto de vista de la mecánica y la medicina entran en un análisis más avanzado, el cual va más allá de este trabajo.

TABLA 7: PUNTO DE CONTACTO TIBIO-FEMORAL EN TRES PACIENTES LESIONADOS.

#Paciente	100° - 110° (Flexión) Datos en %		135° - 145° Datos en %		170° - 180° (Extensión) Datos en %		Media de la migración Datos en %	
	SANA	LES	SANA	LES	SANA	LES	SANA	LES
1	57	48	63	46	60	58	58	51
2	52	55	54	45	67	51	55	50
3	54	46	68	46	65	51	63	46

TABLA 8: BRAZO DE MOMENTO EN TRES PACIENTES LESIONADOS.

#Paciente	100° - 110° (Flexión) Datos en <i>mm</i>		135° - 145° Datos en <i>mm</i>		170° - 180° (Extensión) Datos en <i>mm</i>		Media de la migración Datos en <i>mm</i>	
	SANA	LES	SANA	LES	SANA	LES	SANA	LES
1	43	43	47	51	44	46	45	48
2	43	46	51	57	41	54	47	52
3	44	47	45	54	43	49	44	51

En base a los valores presentados en las tablas, se pueden identificar las siguientes apreciaciones:

1. Durante el movimiento de extensión, el valor del Brazo de Momento es creciente alcanzando su máximo valor cerca de la mitad del recorrido y volviendo a disminuir en la extensión total de la rodilla.
2. Para todos los pacientes, la media del Brazo de Momento fue mayor en la rodilla lesionada.
3. Para todos los pacientes, la media del desplazamiento del Punto de Contacto Tibiofemoral fue superior en la rodilla sana.

Observemos en primer lugar los valores obtenidos para el PCTF. En Simini y Santos [5] se observa la misma tendencia que en el punto 3. En dicho estudio la media del PCTF en una rodilla sana fue de 62% en promedio para 6 pacientes y de 61% para el caso de la rodilla lesionada. En nuestro caso se obtuvo 59% y 49% respectivamente para 3 pacientes.

En Kellis y Baltzopoulos [8] se obtuvieron valores para el PCTF en términos generales, no en una visión comparativa de estados de la rodilla. La Figura 25 muestra los resultados de dicho

experimento. Por otro lado las gráficas de las Figuras 26, 27 y 28 muestran el PCTF, a partir de los datos de CINARTRO, para cada uno de los 3 pacientes.

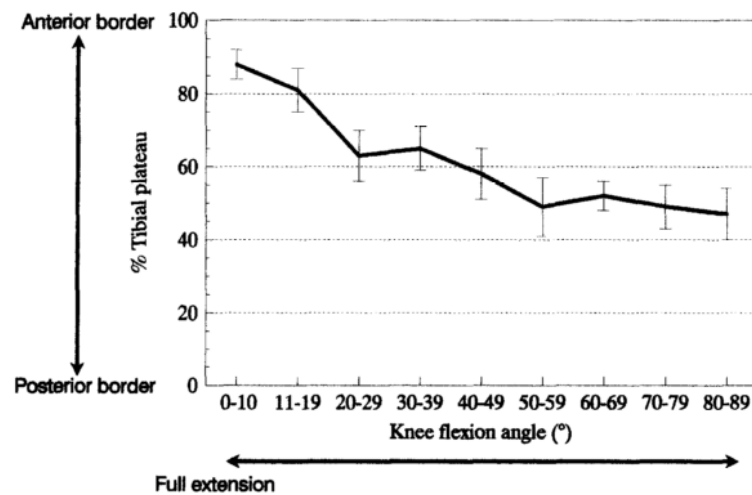


Figura 25: Estudio del PCTF en rodilla sana por Kellis.
De una movilidad de 48% en flexión 90 grados, llega a 88% en extensión completa.
Tomado de Kellis [8].

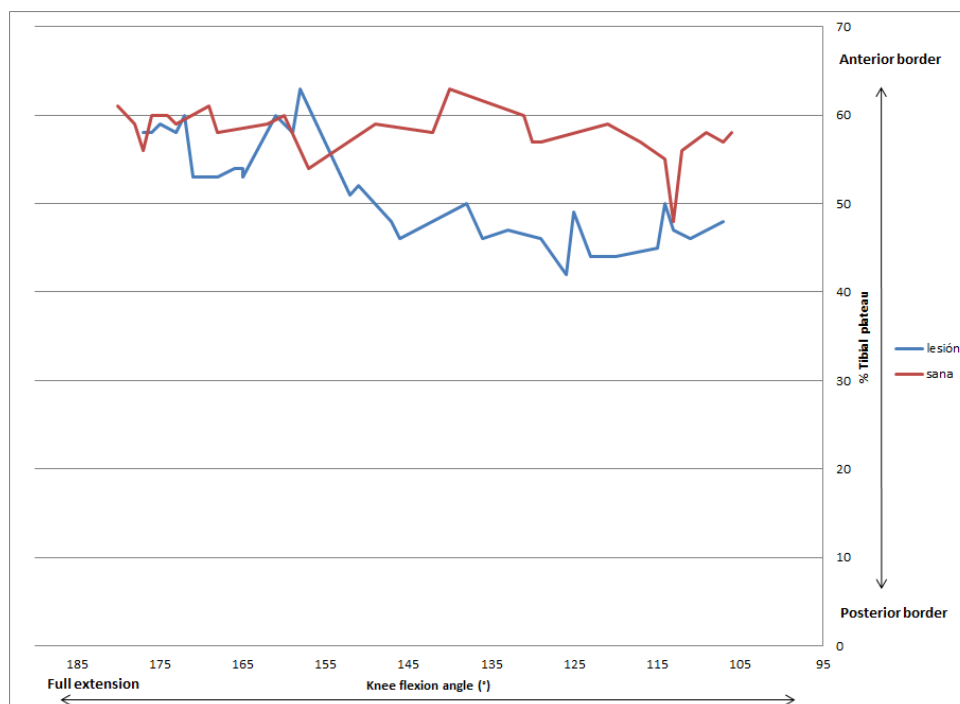


Figura 26: PCTF del paciente 1 estudiado por CINARTRO.
En rodilla sana la migración es de 58% en flexión y llega a 60% en extensión. En la rodilla lesionada, se observa una mayor restricción con migración limitada a 48% en flexión y cerca de 60% en extensión.

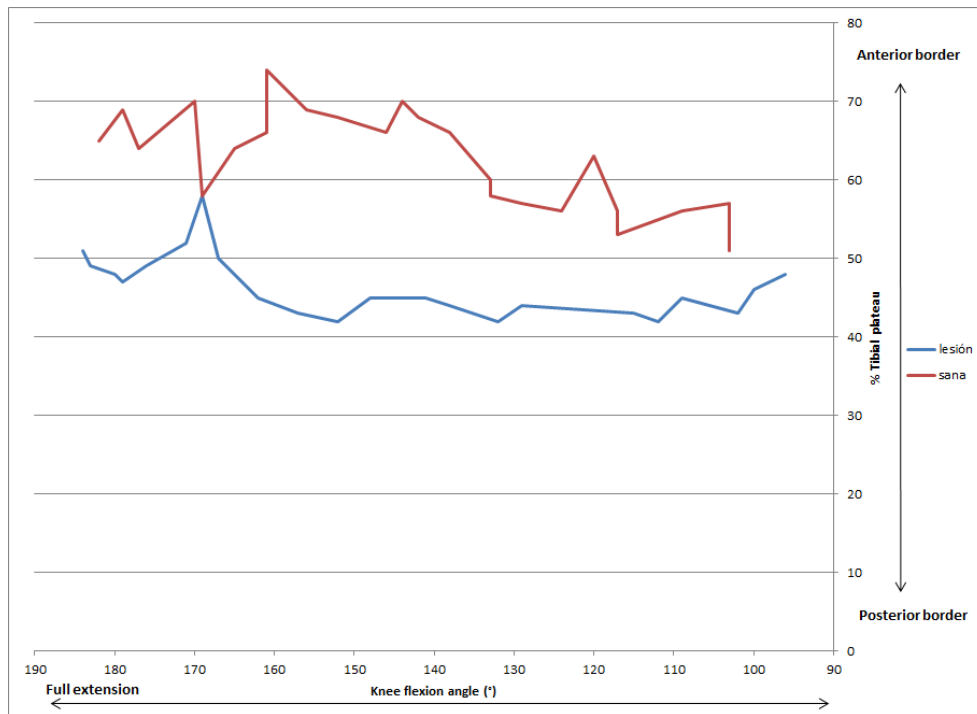


Figura 27: PCTF del paciente 2 estudiado por CINARTRO.

En rodilla sana la migración es de 52% en flexión y llega a 67% en extensión. En la rodilla lesionada, se observa un descenso en el valor con migración de 55% en flexión y cercano a 50% en extensión.

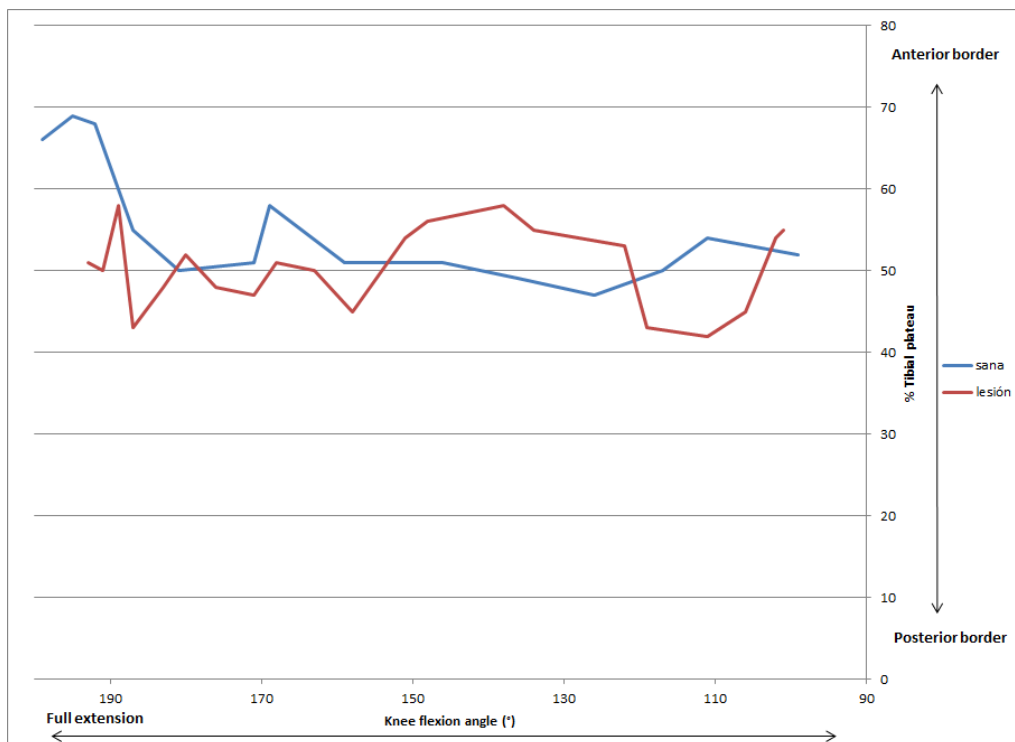


Figura 28: PCTF del paciente 3 estudiado por CINARTRO.

En rodilla sana la migración es de 54% en flexión y llega a 65% en extensión. En la rodilla lesionada, se observa una migración limitada a 46% en flexión y valores cerca de 50% en extensión.

En la Figura 25 puede notarse una tendencia decreciente de los valores a medida que la pierna se flexiona. Esto también puede apreciarse en las gráficas de los pacientes 1 y 3, aunque la velocidad al decrecer es mucho menor. Por otro lado, no se puede decir lo mismo del paciente 2 que alterna sus valores en todo el recorrido.

En la Figura 29 se muestra el promedio del PCTF de los tres pacientes por rangos de flexión, al igual que la Figura 25.

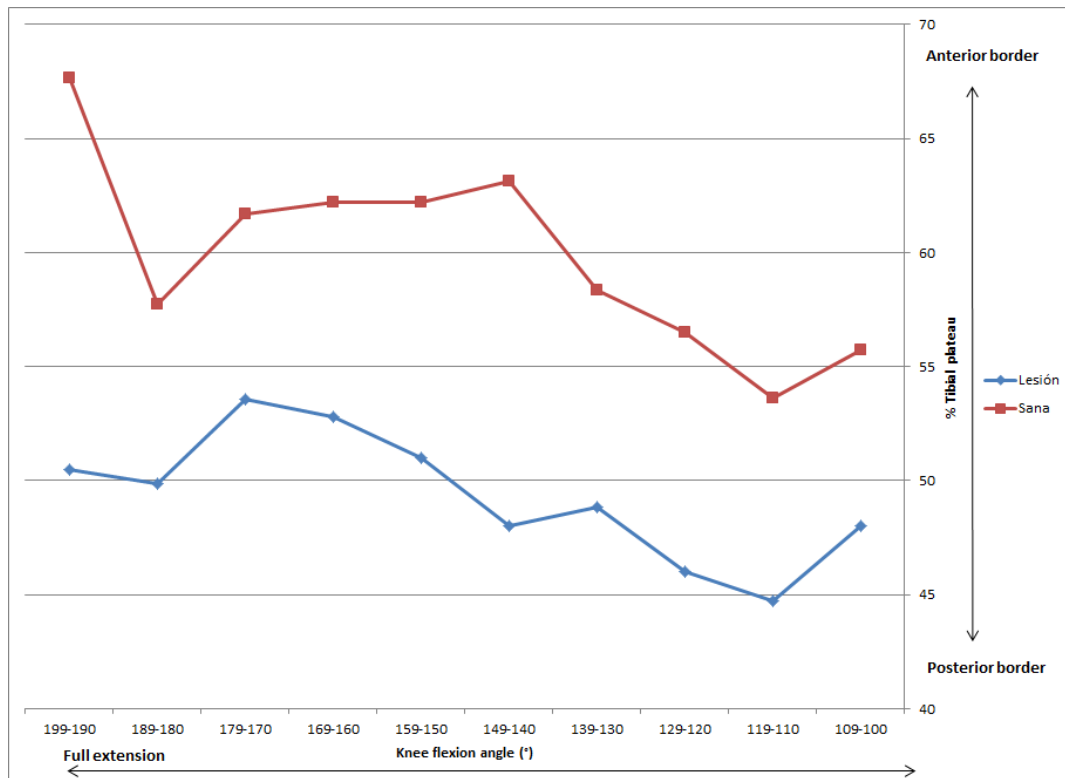


Figura 29: Promedio del PCTF en tres pacientes para rodilla sana y lesionada.

Se toman rangos de flexión para suavizar en un mismo par de ejes los valores obtenidos para el PCTF en los tres pacientes estudiados por CINARTRO.

En cuanto al Brazo de Momento, el punto 1 concuerda con los resultados mostrados en Baltzopoulos [7] y en Kellis y Baltzopoulos [8], donde se describe la misma tendencia para los datos obtenidos. Sin embargo no se ve una relación tan directa con los valores en sí, ya que en esos casos se mencionan mínimos cercanos a 35mm y máximos cercanos a 43mm cuando en nuestro caso se obtuvieron entre 44mm y 51mm respectivamente.

Las gráficas de Figura 30, 31 y 32 muestran el valor del Brazo de Momento en función del ángulo de extensión para los 3 pacientes. Los valores están graficados como puntos de dispersión y se agrega una línea de tendencia polinómica de segundo grado. De esta forma podemos apreciar el comportamiento a lo largo del movimiento.

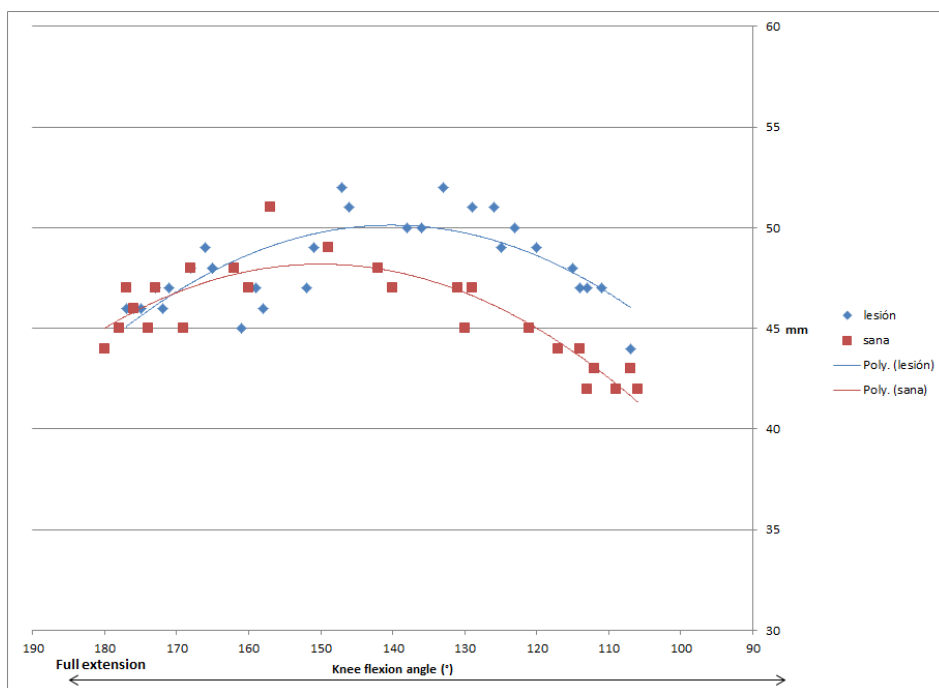


Figura 30: Brazo de Momento (BM) del paciente 1 estudiado por CINARTRO. En la rodilla sana el valor del BM varía levemente entre 43mm y 47mm, alcanzando su valor máximo en el medio del recorrido. La rodilla lesionada presenta un comportamiento similar obteniéndose 48mm en promedio.

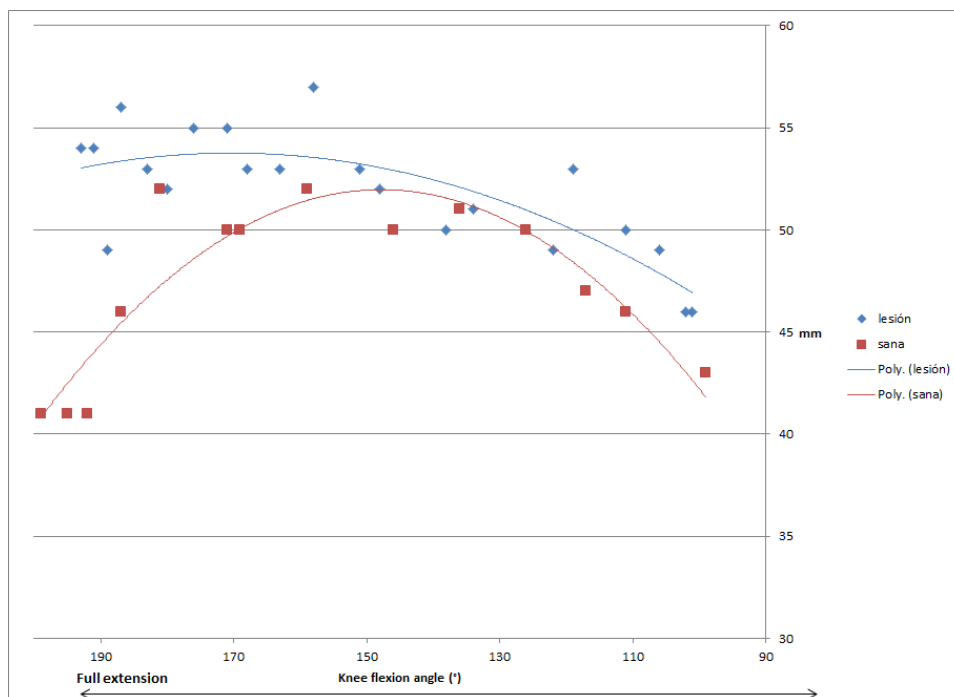


Figura 31: BM del paciente 2 estudiado por CINARTRO. En la rodilla sana el valor del BM es de 41mm en flexión y 43mm en extensión y alcanza su valor máximo de 51mm en el medio del recorrido. La rodilla lesionada presenta un comportamiento similar pero con mayores longitudes que varían entre los 46mm y 57mm.

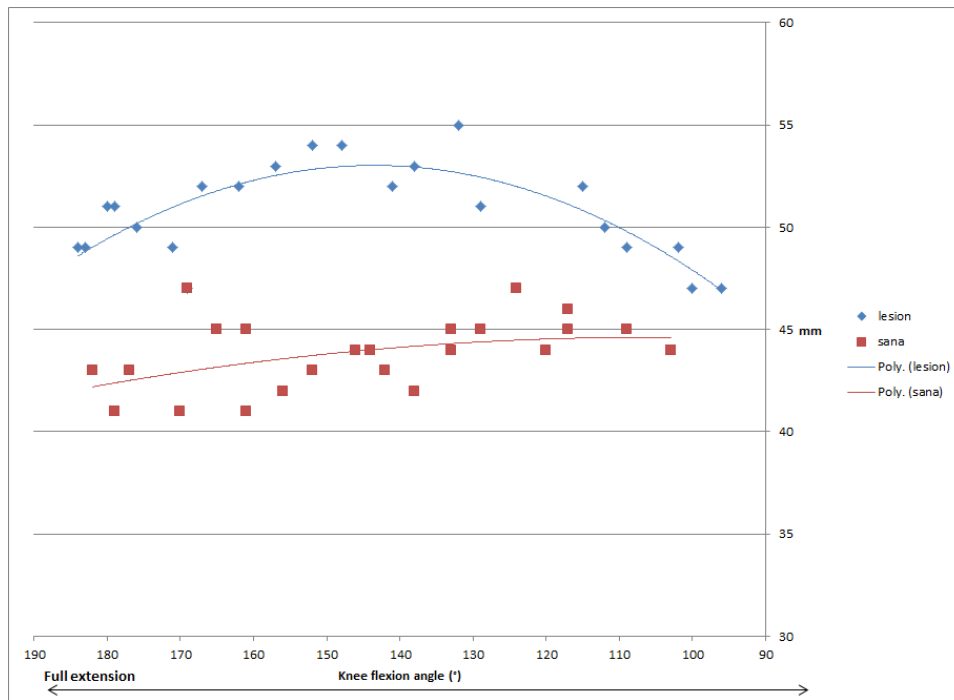


Figura 32: BM del paciente 3 estudiado por CINARTRO.

En la rodilla sana el valor del BM es de 41mm en flexión y 45mm en extensión aproximadamente, alcanzando su valor máximo de 47mm nuevamente en el medio del recorrido. Los valores del BM para la rodilla lesionada son mayores como se aprecia en la gráfica, con longitudes entre 47mm y 54mm.

Las diferencias en los resultados, tanto del PCTF como del BM, respecto a otros estudios se atribuyen principalmente a dos razones. La primera son los errores de precisión cometidos durante el proceso de digitalización de los puntos. Ubicar un punto en la posición incorrecta modifica levemente el resultado pero en el total de imágenes la diferencia es notable.

En segundo lugar es importante tener en cuenta que la rodilla de cada paciente es diferente, por lo tanto los valores asociados para los parámetros de interés varían de uno a otro. Cabe destacar que en los resultados de Kellis y Baltzopoulos [8], Baltzopoulos [7] y Simini y Santos [5] también existen este tipo de diferencias entre los estudios.

8.3. Resultados en los métodos de corrección

Otra de las partes significativas en el procedimiento de análisis es la corrección de la distorsión. Se implementaron dos algoritmos que permitieron la resolución de esta problemática: un algoritmo de reconocimiento de grises y otro de corrección de la distorsión de *pincushion*. El primero de ellos utiliza la imagen del fantoma y a través de un cuadro de guía marcado por el usuario, identifica la posición exacta de los puntos de plomo. El segundo algoritmo, a partir de las coordenadas obtenidas por el reconocimiento de imagen, construye una matriz de valores que posteriormente permite aplicar las fórmulas de corrección de la distorsión en tiempo de ejecución del programa.

En relación al segundo algoritmo mencionado, en el Capítulo 7 se realizaron pruebas variando el porcentaje de puntos considerados en torno a un punto seleccionado para la corrección de sus coordenadas. El valor utilizado es de 40% para lo cual se obtiene un error de $0.2348\text{mm} \sim 0.235\text{mm}$. Baltzopoulos [7], por otra parte, quien también considera un 40% de los puntos en torno al punto elegido, obtiene un error de 0.246 mm . Finalmente en Kellis y Baltzopoulos [8], utilizando un número mucho menor de puntos, se obtiene un error de 0.270 mm . Con estos resultados podemos decir que el valor hallado se asemeja al error global cometido respecto a los valores de referencia de otros estudios, validando en cierta forma los cálculos realizados.

Capítulo 9

Conclusiones

Se construyó una aplicación de software que reúne el conjunto de funcionalidades propuesto inicialmente. Dicha aplicación es capaz de determinar el valor del Punto de Contacto Tibio-Femoral (PCTF) y el Brazo de Momento (BM), ambos elementos fundamentales para el análisis de la mecánica de la articulación de la rodilla. En base al estudio de tres pacientes para su rodilla sana y lesionada utilizando 30 imágenes en cada caso, para el PCTF se obtuvo un promedio de 42% de desplazamiento respecto al platillo tibial en un movimiento de extensión y para el BM un promedio de 51 mm.

El procedimiento para cuantificar estos conceptos implicó la corrección de la distorsión en las imágenes fluoroscópicas a través del uso de un fantoma. Desde el punto de vista de la programación fue un gran desafío implementar los algoritmos de reconocimiento de puntos y de corrección de la distorsión. El primero de ellos sufrió diversos cambios y mejoras a lo largo del tiempo tomando en cuenta cada vez más consideraciones sobre cómo identificar la ubicación de los puntos. Para el segundo algoritmo las funciones de corrección fueron implementadas en base a una formulación puramente teórica, teniendo que diseñar la solución mediante la representación en código de la geometría del espacio. Ciertamente, estos procedimientos aportaron calidad a los estudios al reducir el error producido por el efecto *pincushion* y proveer la habilidad de procesar los datos en milímetros.

Destacamos lo aprendido acerca de las entidades, protocolos y datos involucrados para el manejo de pacientes y reportes clínicos. Si bien para algunos casos resultó trabajoso comprender todas las implicancias para lograr los objetivos, tanto el sistema EMPI como la creación de reportes CDA y el uso de repositorios XDS pudieron ser integrados al sistema potenciando la calidad de CINARTRO.

El trabajo fue probado por usuarios calificados, donde se relevaron opiniones tanto de personas que desconocen sobre el área como de aquellas que trabajan con ello. Claramente las evaluaciones más importantes vinieron de estos últimos, nutriendo de información desde su experiencia y también generando cuestiones muy interesantes. Su devolución permitió también mejoras en aspectos de la interfaz y resolver problemas encontrados en algunas funcionalidades. En cuanto a usabilidad, se redujeron los tiempos requeridos para completar un estudio y también se detectaron y mejoraron ciertas problemáticas de performance con la ayuda de programas externos.

Una de las cosas más enriquecedoras para el equipo fue el aprendizaje de elementos relacionados a la medicina. Todas las terminologías y conceptos de la anatomía y biomecánica de la rodilla, que inicialmente se presentaban como algo bastante alejado a lo que conocemos, terminó aportando muy interesantes conocimientos.

El trabajo realizado deja las puertas abiertas para su continuidad en el futuro, ya sea mejorando las funcionalidades o haciendo crecer el sistema con nuevas herramientas y

componentes o también incursionando en la metodología de trabajo pero aplicado a otras áreas del cuerpo humano.

Capítulo 10

Trabajos futuros

Los objetivos determinados inicialmente fueron alcanzados en su mayoría. Sin embargo, quedaron algunas puertas abiertas para la evolución del trabajo realizado. A continuación se listan los puntos más relevantes que tienen lugar a mejoras e investigación adicionales que enriquecerían el estado actual de CINARTRO.

Conexión con arco en C

El espacio de interacción de la aplicación podría ser extendido para abarcar la fase en que es creada la videofluoroscopia para un paciente, de forma de evitar trasladar manualmente la información de salida del Arco en C hacia la aplicación y así mejorar el flujo de trabajo. Para lograrlo, se debería investigar cómo comunicar CINARTRO con la interfaz de salida del Arco en C.

Sugerencia de puntos

Durante las pruebas realizadas notamos que al marcar los puntos sobre cada una de las imágenes, la variación de los ángulos formados para la posición de la rodilla entre todas las imágenes abarca en general siempre un mismo rango de grados debido al movimiento de extensión. Actualmente el usuario debe corregir el ángulo en cada nueva imagen, pero es viable poder sugerir ángulos que faciliten la tarea. La idea es utilizar la selección del usuario de imágenes anteriores, y considerando que el ángulo siempre va en aumento, sugerir la nueva posición una vez que cambie de imagen. En el mejor de los casos sería el ángulo correcto o de lo contrario podría corregirlo. Esto también podría ser extendido para todos los puntos, no solo los que permiten calcular el ángulo.

Reconocimiento de imágenes

Si bien la tarea del usuario de marcar puntos fue diseñada de forma que se deban ubicar la menor cantidad posible para realizar los cálculos, sería un avance mucho mayor aplicar funcionalidades de reconocimiento de imagen que ayuden a identificar zonas o patrones para una radiografía. El trabajo sería ir un paso más allá de lo que se menciona en el punto anterior. El objetivo ideal sería que automáticamente se identifiquen los puntos que hoy en día debe marcar o al menos una aproximación de los mismos.

Ayuda dentro del software

Aunque el usuario final al cual se apunta debe tener un perfil médico, es posible que algunas funcionalidades no resulten del todo intuitivas, perdiéndose de vista dentro de la aplicación. Por este motivo, sería de utilidad para algunos usuarios contar con textos de ayuda dentro del software que le permitan identificar las acciones en caso de ser necesario.

Trabajo de diseño para la interfaz

Debido a que no era uno de las metas prioritarias el diseño de la aplicación en cuanto a imágenes, tipografía, íconos y distribución de los elementos, entre otros, la misma contiene varias imperfecciones que podrían ser mejoradas principalmente para que al usuario agrade aún más visualmente el contenido del programa.

Idiomas adicionales

Otra de las características a mejorar es aumentar la cantidad de idiomas seleccionables al usuario para progresar en el área de internacionalización del software. Actualmente el programa obtiene todos sus textos de diccionarios internos ubicados en archivos distintos según su idioma. Para agregar nuevos idiomas simplemente se tendría que realizar la traducción de los textos mostrados y agregar ese nuevo diccionario a la colección.

Mejora de contraste de imágenes

Esta mejora fue solicitada exclusivamente por los usuarios que probaron la aplicación. Se propone tener una funcionalidad para modificar el contraste y brillo de las imágenes dentro del programa para facilitar la tarea de marcación de los puntos.

Corrección de la posición del fantoma

Uno de los inconvenientes experimentados al realizar las pruebas con un conjunto de imágenes reales, fue contar con imágenes de fantoma que estaban levemente rotadas respecto a un par de ejes convencionales. Estos casos hubo que descartarlos ya que no fue posible extraer la información necesaria para aplicar la corrección, debido a que el proceso encargado del reconocimiento sobre la imagen no podía identificar los puntos como en una grilla perfecta. Una solución que se propone a futuro para esto es proveer la habilidad de rotar manualmente la imagen dentro de la aplicación hasta que esta se encuentre correctamente alineada, y así evitar errores en los procedimientos.

Bibliografía

- [1] L. Y. Griffin, J. Agel et al. «Noncontact anterior cruciate ligament injuries: risk factors and prevention strategies», *J Am.Acad.Orthop.Surg.*, vol. 8, n.º 3, pp. 141-150, 2000.
- [2] S. M. Gianotti, S. W. Marshall et al. «Incidence of anterior cruciate ligament injury and other knee ligament injuries: a national population-based study.», *J. Sci. Med. Sport*, vol. 12, n.º 6, pp. 622-7, 2009.
- [3] G. Myklebust y R. Bahr, «Return to play guidelines after anterior cruciate ligament surgery.», *Br. J. Sports Med.*, vol. 39, n.º 3, pp. 127-31, 2005.
- [4] L. M. Kruse, B. Gray, y R. W. Wright, «Rehabilitation after anterior cruciate ligament reconstruction: a systematic review.», *J. Bone Joint Surg. Am.*, vol. 94, n.º 19, pp. 1737-48, 2012.
- [5] F. Simini y D. Santos, «Anterior Cruciate Ligament reconstruction follow-up instrumentation based on Centre of Rotation videofluoroscopy determination: Development of an original equipment, CINARTRO, and first clinical use», *Conf. Rec. - IEEE Instrum. Meas. Technol. Conf.*, pp. 923-926, 2014.
- [6] D. Santos, «Estudio de los Centros de Rotación Instantáneos de la Rodilla en Pacientes con Plastia del Ligamento Cruzado Anterior», Tesis de maestría en Ciencias Médicas; PROINBIO, Universidad de la República. Montevideo, 2014.
- [7] V. Baltzopoulos, «A videofluoroscopy method for optical distortion correction and measurement of knee-joint kinematics», *Clin. Biomech.*, vol. 10, n.º 2, pp. 85-92, 1995.
- [8] E. Kellis y V. Baltzopoulos, «In vivo determination of the patella tendon and hamstrings moment arms in adult males using videofluoroscopy during submaximal knee extension and flexion», *Clin. Biomech.*, vol. 14, n.º 2, pp. 118-124, 1999.
- [9] J. T. Bryant, H. W. Wevers, y P. J. Lowe, «One parameter model for error in instantaneous centre of rotation measurements.», *J. Biomech.*, vol. 17, n.º 5, pp. 317-323, 1984.
- [10] J. E. Giphart, C. a. Zirker et al. «Accuracy of a contour-based biplane fluoroscopy technique for tracking knee joint kinematics of different speeds», *J. Biomech.*, vol. 45, n.º 16, pp. 2935-2938, 2012.
- [11] Health Level Seven International (HL7), «Clinical Document Architecture (CDA)», *Release 2.0, HL7 2005 Normative Edition*. 2005.
- [12] Integrating the Healthcare Enterprise (IHE), «Cross-Enterprise Document Sharing (XDS)», *Visto en Mayo 2016*. [En línea]. Disponible en: http://wiki.ihe.net/index.php/Cross-Enterprise_Document_Sharing.
- [13] Presidencia de la República, «Agencia de Gobierno Electrónico y Sociedad de la Información de Uruguay (AGESIC)», *Visto en Mayo 2016*. [En línea]. Disponible en: <http://www.agesic.gub.uy/>.
- [14] Agencia de Gobierno Electrónico y Sociedad de la Información de Uruguay (AGESIC), «SALUD.UY», *Visto en Mayo 2016*. [En línea]. Disponible en: <http://salud.gub.uy/>.
- [15] F. Portilla, «Clínica Electrónica e Interoperabilidad en el Sector de Salud», Programa Salud.uy, AGESIC. Montevideo, Uruguay, 2015.
- [16] M. E. Easley, F. D. Cushner, y W. N. Scott, «Insall & Scott Surgery of the Knee», *Surgery of the Knee*. pp. 473-520, 2001.
- [17] J. Rodés y J. Piqué, *Libro de la Salud del Hospital de Clínicas de Barcelona*. Barcelona, España: Editorial Nerea S.A, Fundación BBVA, 2007.
- [18] E. J. De Valk, M. H. Moen et al. «Preoperative patient and injury factors of successful

- rehabilitation after anterior cruciate ligament reconstruction with single-bundle techniques», *Arthroscopy - Journal of Arthroscopic and Related Surgery*, vol. 29, n.º 11. pp. 1879-1895, 2013.
- [19] A.D.A.M American Accreditation HealthCare Commission, «Lesión del LCA», *Visto en Mayo 2016*. [En línea]. Disponible en: <https://www.nlm.nih.gov/medlineplus/spanish/ency/article/001074.htm>.
- [20] G. L. Smidt, «Biomechanical analysis of knee flexion and extension», *J. Biomech.*, vol. 6, n.º 1, 1973.
- [21] National Electrical Manufacturers Association (NEMA), «Digital Imaging and Communications in Medicine (DICOM)», *Visto en Mayo 2016*. [En línea]. Disponible en: <http://dicom.nema.org/>.
- [22] P. Linden, «Videofluoroscopy in the rehabilitation of swallowing dysfunction», *Dysphagia*, vol. 3, n.º 4, pp. 189-191, 1989.
- [23] R. Nisell, «Mechanics of the Knee», *Acta Orthop.*, vol. 56, n.º 770885181, pp. 1-42, 1985.
- [24] R. D. Komistek, T. R. Kane et al. «Knee mechanics: A review of past and present techniques to determine in vivo loads», *Journal of Biomechanics*, vol. 38, n.º 2. pp. 215-228, 2005.
- [25] P. J. Laz, S. Pal et al. «Probabilistic finite element prediction of knee wear simulator mechanics», *J. Biomech.*, vol. 39, n.º 12, pp. 2303-2310, 2006.
- [26] W. Micheo, L. Hernández, y C. Seda, «Evaluation, management, rehabilitation, and prevention of anterior cruciate ligament injury: current concepts.», *PM R*, vol. 2, n.º 10, pp. 935-944, 2010.
- [27] J. T. Podraza y S. C. White, «Effect of knee flexion angle on ground reaction forces, knee moments and muscle co-contraction during an impact-like deceleration landing: Implications for the non-contact mechanism of ACL injury», *Knee*, vol. 17, n.º 4, pp. 291-295, 2010.
- [28] D. Santos, F. Massa, y F. Simini, «Evaluation of anterior cruciate ligament reconstructed patients should include both self-evaluation and anteroposterior joint movement estimation?», *Phys. Ther. Rehabil.*, vol. 2, n.º 1, p. 3, 2015.
- [29] G. Li, S. K. Van de Velde, y J. T. Bingham, «Validation of a non-invasive fluoroscopic imaging technique for the measurement of dynamic knee joint motion», *J. Biomech.*, vol. 41, n.º 7, pp. 1616-1622, 2008.
- [30] M. R. Mahfouz, W. a. Hoff et al. «Effect of segmentation errors on 3D-to-2D registration of implant models in X-ray images», *J. Biomech.*, vol. 38, n.º 2, pp. 229-239, 2005.
- [31] P. Wretenberg, D. K. Ramsey, y G. Németh, «Tibiofemoral contact points relative to flexion angle measured with MRI», *Clin. Biomech.*, vol. 17, n.º 6, pp. 477-485, 2002.
- [32] J. M. Scarvell, P. N. Smith et al. «Evaluation of a method to map tibiofemoral contact points in the normal knee using MRI», *J. Orthop. Res.*, vol. 22, n.º 4, pp. 788-793, 2004.
- [33] R. J. de Jong, P. J. C. Heesterbeek, y A. B. Wymenga, «A new measurement technique for the tibiofemoral contact point in normal knees and knees with TKR», *Knee Surgery, Sport. Traumatol. Arthrosc.*, vol. 18, n.º 3, pp. 388-393, 2010.
- [34] R. M. Ehrig, W. R. Taylor et al. «A survey of formal methods for determining the centre of rotation of ball joints», *J. Biomech.*, vol. 39, n.º 15, pp. 2798-2809, 2006.
- [35] J. D. Moorehead, S. C. Montgomery, y D. M. Harvey, «Instant center of rotation estimation using the Reuleaux technique and a Lateral Extrapolation technique», *J. Biomech.*, vol. 36, n.º 9, pp. 1301-1307, 2003.
- [36] K. Soudan, R. Van Audekercke, y M. Martens, «Methods, difficulties and inaccuracies in the study of human joint kinematics and pathokinematics by the instant axis

- concept. Example: The knee joint», *J. Biomech.*, vol. 12, n.º 1, pp. 27-33, 1979.
- [37] B. Eliot y M. Bouza, «CINARTRO: aplicación para la determinación del centro de rotación de la articulación de la rodilla y el seguimiento de los pacientes operados del ligamento cruzado», Módulo de Taller, Facultad de Ingeniería - Instituto de Ingeniería Eléctrica (IIE), Montevideo, Uruguay, 2013.
 - [38] MathWorks Inc, «Matlab», *Visto en Mayo 2016*. [En línea]. Disponible en: <http://www.mathworks.com/products/matlab>.
 - [39] F. Jorge y R. Carbonell, «Corrección de distorsión y detección de movimiento en Fluoroscopias de rodilla», Proyecto de Investigación, Facultad de Ingeniería - Instituto de Ingeniería Eléctrica (IIE), Montevideo, Uruguay, 2008.
 - [40] G. Jones, «ACL Rehab», *Visto en Mayo 2016. v 1.4.0*. iTunes Apple, Globe Physiotherapy, United States, 2015.
 - [41] V. Karade, «Knee XRayTo3D», *Visto en Mayo 2016. Online*. AlgoSurgPvt, United States, 2012.
 - [42] EmpireDigital LLC, «My Knee Guide», *Visto en Mayo 2016. v 1.7*. iTunes Apple, EmpireDigital LLC, 2013.
 - [43] Sysnet International, «Open Enterprise Master Patient Index (OpenEMPI)», *Visto en Mayo 2016*. [En línea]. Disponible en: <http://www.openempi.org/>.
 - [44] Siemens AG, «Siemens C Arc Arcadis Avantis», *Visto en Mayo 2016*. [En línea]. Disponible en: <http://www.healthcare.siemens.com/surgical-c-arms-and-navigation/mobile-c-arms/arcadis-avantic>.
 - [45] V. Gigirey, «Dynamic X ray Image Acquisition. Xray Knee Anatomy.», Curso PROINBIO. Hospital de Clínicas. Montevideo - Uruguay, 2016.
 - [46] E. H. Garling, B. L. Kaptein et al. «Marker Configuration Model-Based Roentgen Fluoroscopic Analysis», *J. Biomech.*, vol. 38, n.º 4, pp. 893-901, 2005.
 - [47] Oracle Corporation, «Java Software», *Visto en Mayo 2016. v 8*. [En línea]. Disponible en: <https://www.java.com/es/about/>.
 - [48] Oracle Corporation, «JavaFX», *Visto en Mayo 2016. v 8*. [En línea]. Disponible en: <http://docs.oracle.com/javafx/>.
 - [49] Oracle Corporation, «Java Swing», *Visto en Mayo 2016. v 7*. [En línea]. Disponible en: <https://docs.oracle.com/javase/tutorial/uiswing/>.
 - [50] Oracle Corporation, «Java Scene Builder», *Visto en Mayo 2016. v 2.0*. [En línea]. Disponible en: <http://www.oracle.com/technetwork/java/javase/downloads/sb2download-2177776.html>.
 - [51] Apache Software, «Apache Commons Math», *Visto en Mayo 2016. v 3.6.1*. [En línea]. Disponible en: <https://commons.apache.org/proper/commons-math/>.
 - [52] LLC ConnectSolutions, «Xuggle/Xuggler», *Visto en Mayo 2016. v 5.2*. [En línea]. Disponible en: <http://www.xuggle.com/xuggler/>.
 - [53] FFmpeg team, «FFmpeg», *Visto en Mayo 2016. v 2.5*. [En línea]. Disponible en: <https://ffmpeg.org/>.
 - [54] Apache Software, «HTTPClient», *Visto en Mayo 2016. v 5.0*. [En línea]. Disponible en: <http://hc.apache.org/httpcomponents-client-ga/>.
 - [55] NV iText Group, «iText PDF», *Visto en Mayo 2016. v 5.5.7*. [En línea]. Disponible en: <http://www.itextpdf.com/>.
 - [56] Oracle Corporation, «Java Architecture for XML Binding (JAXB)», *Visto en Mayo 2016. v 2.2*. [En línea]. Disponible en: <https://jaxb.java.net/>.
 - [57] Expansion of flexible copyright and license., «Creative Commons (CC)», *Visto en Mayo 2016*. [En línea]. Disponible en: www.creativecommons.org.
 - [58] Oracle Corporation, «VisualVM», *Visto en Mayo 2016. v 1.3.8*. [En línea]. Disponible

- en: <https://visualvm.java.net/>.
- [59] Oracle Corporation, «Open Java Development Kit (OpenJDK)», *Visto en Mayo 2016. v 8*. [En línea]. Disponible en: <http://openjdk.java.net/>.
- [60] «Health Level Seven International (HL7)», *Visto en Mayo 2016*. [En línea]. Disponible en: <http://www.hl7.org/>.
- [61] «Integrating the Healthcare Enterprise (IHE)», *Visto en Mayo 2016*. [En línea]. Disponible en: <http://www.ihe.net/>.
- [62] «Asociación Colombiana de Bioingeniería y Electrónica Médica (ABIOIN)», *Visto en Mayo 2016*. [En línea]. Disponible en: <http://www.abioin.com/>.
- [63] Fog Creek Software, «Trello», *2011*. [En línea]. Disponible en: <https://trello.com/>.

Glosario

Software médico: Son programas de software destinados exclusivamente a fines médicos. La variedad de este tipo de programas es muy amplia, pudiendo encontrarse software específicamente creado para la comunicación con el hardware de aparatos médicos o también para el uso directamente sobre pacientes por parte de los especialistas.

Fluoroscopia: La fluoroscopia o radioscopia es una técnica de imagen usada en medicina para obtener imágenes en tiempo real de las estructuras internas de los pacientes mediante el uso de un fluoroscopio. Un fluoroscopio básico consiste de una fuente de rayos X y una pantalla fluorescente entre las que se sitúa al paciente.

Framework: Es un software formado por un conjunto de funcionalidades que provee soporte específico a otros programas de mayor escala. Por lo general resuelven problemas específicos y pueden ser reutilizados en diversos ambientes y situaciones.

Pincushion: Es un tipo de distorsión óptica dentro de un sistema de proyección. Se caracteriza por el efecto visible que se produce en aquellas líneas de proyección que no atraviesan el centro de la imagen, las cuales se inclinan hacia el interior causando curvaturas en la imagen resultante.

Angiógrafo: Es un equipo utilizado para procedimientos de radiología. Consta de un tubo de rayos X y un sistema de video que permite obtener imágenes en tiempo real. Se utiliza comúnmente para el análisis del sistema vascular.

CSV: Son las siglas de *Comma Separated Values* (o valores separados por comas), y refieren a un tipo de archivo, con extensión .csv, que admite texto con formato estructurado. Tiene la particularidad de que puede ser interpretado por programas que soportan la lectura de hojas de cálculo. Se diferencian de estos en que sólo pueden tener una hoja por archivo.

Frame: Si bien esta palabra puede tener muchas connotaciones, en nuestro caso se refiere a un frame de imagen. Lo definimos como la representación visual de cierto elemento a través de una imagen. En particular para este estudio, se refiere a una imagen individual dentro de un video el cual precisamente está formado por una gran secuencia de frames.

HSB (Hue, Saturation, Brightness): Es un tipo de representación de los colores, los cuales se construye a través de un cilindro que posee tres propiedades que son: matiz, saturación y brillo. Estas tres componentes permiten determinar un color específico.

Gantt: También conocido como Diagrama de Gantt, es un tipo de gráfica de barras que es utilizada para representar el cronograma de un proyecto. Permite visualizar principalmente las actividades y dependencias entre las mismas.

Spline: Es una función matemática definida a trozos por polinomios. Se caracteriza por zonas de curvatura complejas que son simplificadas por el uso de funciones individuales que se unen para formar un único gráfico.

Imagenología: El concepto se utiliza para nombrar al conjunto de técnicas y procedimientos que permiten obtener imágenes del cuerpo humano con fines clínicos o científicos.

Protocolo REST: Sus siglas refieren a *Representational State Transfer* (o transferencia del estado de representación), el cuál es un tipo de arquitectura de software utilizada para el intercambio de mensajes en internet. Define un estándar o protocolo que especifica la forma en que los mensajes deben ser enviados entre las entidades.

Protocolo SOAP: Son siglas de *Simple Object Access Protocol* (o protocolo de acceso simple a objetos) que define la comunicación entre procesos a través de datos en formato XML. En nuestro caso esos procesos están definidos por el software de CINARTRO y un servidor externo.

Programación Orientada a Objetos: Es un paradigma de diseño que organiza el modelo de un lenguaje de programación mediante objetos en lugar de acciones o procedimientos lógicos. Cada objeto posee su propia identidad y propiedades, y es manipulado independientemente de otros objetos.

Patrones de diseño: Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software que fueron probadas, documentadas y están sujetas a determinados contextos.

Singleton: En términos de programación y bajo el paradigma de orientación a objetos, es un patrón de diseño que restringe a la instancias de una clase a un único objeto. Esto es de utilidad cuando sólo se necesita de un objeto para manejar todas las acciones de una clase.

Model View Controller: También conocido en español como Modelo-Vista-Controlador, es un patrón de arquitectura de software que divide la aplicación en tres partes que se comunican entre sí y son precisamente llamadas modelo, vista y controlador.

Apéndice A

Utilización del índice EMPI

En repetidas ocasiones dentro del contenido de los distintos capítulos, se mencionaron ciertos aspectos acerca del EMPI y cómo es utilizado dentro de la aplicación. En este espacio complementamos la descripción de esta herramienta y vemos detalles de su utilización.

El EMPI, por sus siglas en inglés de *Enterprise Master Patient Index*, es una base de datos utilizada por los sistemas de salud que mantiene información médica de los pacientes en forma consistente, precisa y demográficamente correcta. Como su nombre lo indica esta base de datos es utilizada dentro de una *Enterprise* u Organización, que puede ser desde un país, una cadena de instituciones o simplemente un conjunto de personas. Si bien las reglas y protocolos son definidos como estándares, el manejo de la información depende de cada subsistema.

El objetivo de esta herramienta es asegurar que cada paciente es representado solamente una vez a través de todos los sistemas de software usados en la organización. Para ello, cada paciente es ingresado a los datos junto a un identificador único dentro de la empresa lo que permite diferenciarlos unos de otros y a su vez recuperar el resto de la información de los mismos. Los datos esenciales de un perfil incluyen: nombre, género, fecha de nacimiento, número de identificación, dirección y teléfono de contacto. Adicionalmente se pueden incluir otro tipo de información requerida por cada organización y datos de su última atención médica.

Los sistemas EMPI existen para resolver principalmente problemas de consistencia dentro de una organización, donde múltiples sistemas puedan modificar la información de los pacientes o incluso utilizar distintos perfiles para un mismo individuo y así dar pérdida de información y generar confusiones en la realidad de los datos. Al utilizar el índice la información es modificada en un solo lugar, globalmente visible a todos los sistemas de software que de esta forma siempre tienen acceso a la última versión de los datos de un paciente en cuestión.

Existen diversos proveedores de sistemas EMPI en el mercado, la mayoría de ellos pagos pero también los hay en forma de Open Source. En el caso de CINARTRO, dentro de la segunda categoría, se utilizó Open EMPI [43].

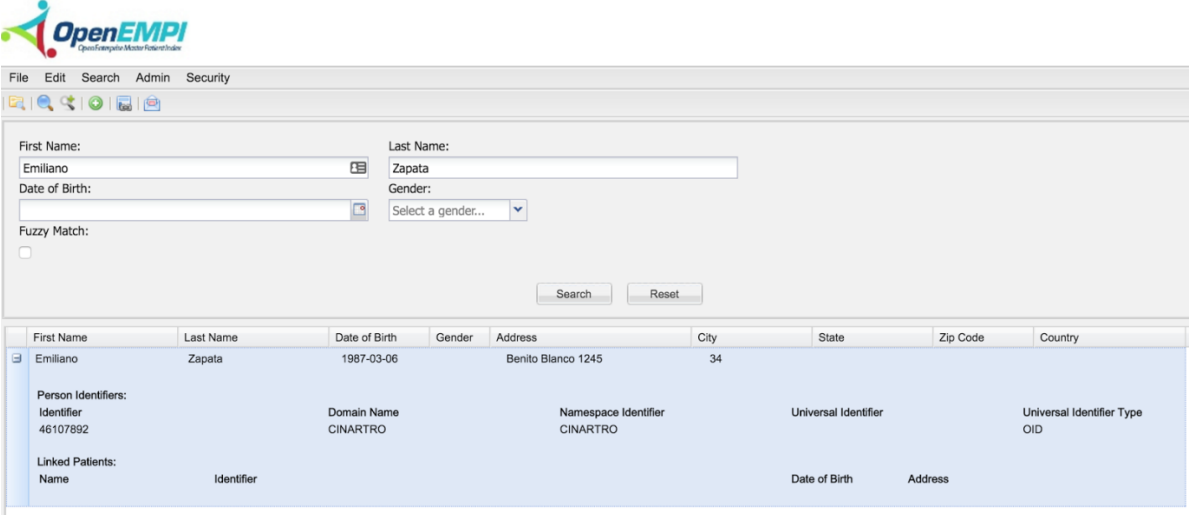
La utilización del EMPI se puede dividir en tres principales etapas. Una de ellas es configurar la plataforma de Open EMPI para su correcto funcionamiento, la segunda es la comprensión del conjunto de servicios que deben ser invocados para la interacción con los datos, lo que luego deriva en la última etapa que es la transferencia de datos propiamente dicha, esto es ingresar y buscar pacientes dentro de la base de datos.

Para la primer situación, consideramos que se tienen objetivos de carácter investigativo para utilizarlo por lo que existen dos opciones que son, ejecutar la plataforma localmente a un sistema o accesible a través de internet utilizando la “nube”.

En ambos casos la configuración es similar, la diferencia está en que para el segundo caso se debe proveer con una dirección del servidor que sea accesible desde el exterior para que cualquier software pueda hacer uso de los servicios del EMPI.

Para configurar la base de datos simplemente se debe descargar la máquina virtual desde la sección de descargas del sitio de OpenEMPI, la cual incluye los componentes necesarios para el correcto funcionamiento.

Una vez montada la maquina virtual en un servidor es posible tener acceso visual, para el manejo administrativo de los datos, utilizando un navegador web que apunte puerto utilizado para correr el programa. En la Figura 33 se muestra un ejemplo de lo mencionado. A través de esta consola, es posible realizar búsquedas, agregar y editar información de pacientes.



The screenshot shows the OpenEMPI administration console. At the top is the OpenEMPI logo. Below it is a menu bar with 'File', 'Edit', 'Search', 'Admin', and 'Security'. The main area contains a search form with fields for 'First Name' (Emiliano), 'Last Name' (Zapata), 'Date of Birth' (1987-03-06), and 'Gender' (Select a gender...). There is also a 'Fuzzy Match' checkbox and 'Search' and 'Reset' buttons. Below the form is a table of registered users.

First Name	Last Name	Date of Birth	Gender	Address	City	State	Zip Code	Country
Emiliano	Zapata	1987-03-06		Benito Blanco 1245	34			

Below the table, there are sections for 'Person Identifiers' and 'Linked Patients'. The 'Person Identifiers' section shows a table with columns: Identifier, Domain Name, Namespace Identifier, Universal Identifier, and Universal Identifier Type. The 'Linked Patients' section shows a table with columns: Name, Identifier, Date of Birth, and Address.

Figura 33: Consola de administración de una cuenta EMPI.

La consola permite visualizar, añadir y eliminar la información existente de los usuarios registrados.

Una vez instalada la maquina virtual es posible comenzar a utilizar los recursos de la plataforma, pero antes es necesario comprender la forma de hacerlo.

Existen dos formas de agregar nueva información y obtener los datos ya existentes. La primera consiste en utilizar el intercambio de consultas bajo la mensajería HL7 [60] para sistemas médicos, mientras que la segunda alternativa es utilizar la interfaz de servicios REST provistos por OpenEMPI.

El primer camino conduce a un extenso marco teórico que especifica diversos conceptos necesarios, no sólo para la utilización del sistema EMPI sino para el manejo de datos clínicos en general. La especificación de estos conceptos y metodología es publicada por la organización *Integrating the Healthcare Enterprise* (IHE) [61].

Un concepto fundamental en esto es el de perfiles PIX/PDQ. El perfil PIX (*Patient Identity Cross-referencing*) soporta la referencia cruzada de identificadores de pacientes que pertenecen a múltiples dominios de identificación. Esta referencia cruzada de identificadores puede ser usada por un servicio de consulta de identificación con el fin de correlacionar información de un paciente sin importar si este tiene distintos identificadores en diferentes dominios. Esto le permite al médico tener una vista más amplia de la información del paciente.

Por otro lado, el perfil PDQ (*Patient Demographics Query*) provee formas de consulta para recuperar la información de un paciente a partir de criterios de búsqueda para luego desplegarlos en la aplicación cliente.

Conociendo esto, el objetivo es construir mensajes en formato HL7 donde estos perfiles PIX/PDQ son secciones dentro de ese mensaje y describen datos sobre el paciente, ya sea para ingresar un nuevo individuo como para recuperar los datos de uno de ellos. El cuerpo de los mensajes, debe seguir las reglas del protocolo para realizar la comunicación con OpenEMPI satisfactoriamente. La Figura 34 muestra ejemplos de este tipo de mensajes.

```

1) Agregar Persona a un dominio
MSH|^~\&|INST3_SERVICES|INST3|EMPI_SERVICES|EMPI|201509091452||ADT^A04
^ADT_A01|10825|P|2.3.1
PID|||10003^^^INST1&2.16.858.0.1&ISO||KLUVER^PEDRO^KLAN|SAENZ|19500202
|F|||||||||333333333

2) Consultar Persona por ID
MSH|^~\&|OPENEMPI|OPENEMPI|EMPI_SERVICES|EMPI|20150928130008||QBP^Q23^
QBP_Q21|005056010223|P|2.5
QPD|IHEPIXQuery|823|10003^^^INST1&2.16.858.0.1&ISO
RCP|I

3) Consultar Persona por Datos Patronímicos
MSH|^~\&|OPENEMPI|OPENEMPI|EMPI_SERVICES|EMPI|20150928130008||QBP^Q22^
QBP_Q21|005056010223|P|2.5
QPD|IHE PDQ Query|494|@PID.5.1.1^PEREZ~@PID.5.2^GERALD
RCP|I

```

Figura 34: Ejemplo de mensaje PIX/PDQ en HL7.

Metadata de ejemplo en formato PIX/PDQ que describe las operaciones de: Agregar Persona, Consultar Persona por Identificador y Consultar Persona por Datos Patronímicos.

La segunda alternativa de comunicación, la cuál fue utilizada a nivel de implementación en CINARTRO, consiste en el envío de mensajes bajo el protocolo REST. Desde la aplicación se invocan las interfaces que permiten realizar todas las acciones para el manejo de usuarios. Considerando la implementación con objetivos puramente investigativos, se utiliza la plataforma OpenEMPI publicada por Salud.uy, que ofrece una interfaz como la descrita anteriormente. De esta forma, se cuenta con el acceso remoto de la información, permitiendo a los usuarios de CINARTRO manejar la información de los pacientes de forma distribuida.

Se tomó la decisión de construir una clase encargada de abstraer la comunicación con el servidor externo de Open EMPI. Esta clase *Singleton* denominada *OpenEmpiConnector*, implementa los métodos de alta y búsqueda de pacientes, para los cuales se reciben un conjunto de campos y se forma el mensaje en base a estos datos para luego realizar el pedido al servicio web correspondiente. En la Figura 35 se muestran los cabezales de dichas funciones. Un detalle a mencionar, es que previamente a invocar un servicio, es necesario obtener un *token* de sesión del servidor, que permita identificar la comunicación con la aplicación cliente. Por lo tanto, se realiza un llamado a este servicio de identificación, se obtiene el *token*, y este es añadido al conjunto de *headers* de la *request* HTTP que se desee realizar.

```
public ArrayList<HashMap<String, String>> searchPatient(String name, String lastName)

public HashMap<String,String> addPatient( String name, String lastName, String CI, String birthDate, String address,
String phoneNumber, String registerNumber, String patientIMC)
```

Figura 35: Métodos de búsqueda y creación de pacientes en EMPI.
Firma de las operaciones definidas para la búsqueda y alta de individuos en el sistema EMPI.

Apéndice B

Creación de un reporte CDA y utilización de repositorio XDS

A modo de conservar el aprendizaje y experiencia obtenidos, describimos a continuación con mayor detalle y en un más bajo nivel a como fuese mencionado anteriormente el proceso de creación de un reporte clínico CDA y cómo éste es subido a un repositorio XDS. Se incluye además código de ejemplo para esclarecer las explicaciones.

Un reporte CDA es un archivo de formato XML que se constituye por grupos de *tags* o identificadores. La información dentro del archivo puede contener cualquier tipo de datos clínicos, como información de altas médicas, patologías, historiales, reportes a través de imágenes, etc. El contenido soporta texto sin estructura y es posible incluir links que referencien a sitios externos. Algunos *tags* deben existir de forma obligatoria para asegurar la interpretación humana de la información mientras que otros, opcionales, funcionan como interpretación a nivel de software.

En el caso de CINARTRO la información a ser guardada implica los aspectos más importantes de un estudio realizado. Así por ejemplo se incluyen los datos del paciente, los datos del médico encargado de realizar la evaluación, las notas ingresadas por el médico como forma de diagnóstico y también se incluyen imágenes divididas en: 3 imágenes tomadas de la videofluoroscopia y la imagen con la representación en una gráfica del estudio ingresado. De esta forma, lo realizado a través del proceso es conservado a través de sus principales características que permiten recuperar la información.

A nivel de implementación el archivo resultante es construido simplemente como un string que concatena los metadatos estáticos y la información dinámica obtenida en cada estudio. La intención inicial fue utilizar la biblioteca JAXB, mencionada en el Capítulo 6, de forma que abstraiga el manejo de caracteres en el bajo nivel y permita manipular la estructura del documento fácilmente. Sin embargo una de las características del CDA son los atributos que deben ser definidos dentro de cada *tag* cuya sintaxis está definida en algunos casos con ciertos caracteres “especiales” que no pueden ser representados por esta biblioteca. Así por ejemplo el *tag* mostrado en la Figura 36 no logra ser reconocido por JAXB, debido a que una de las cosas que exige en los atributos de un XML es que no se tengan caracteres como “:”. De hecho, este no es el único caso, múltiples inconvenientes con símbolos terminaron por decidir resolver el problema de otro modo. Fue así que la solución alternativa se desarrolló en el bajo nivel controlando cada carácter del contenido. Se construyó un “*template*” de archivo CDA, que es compilado en tiempo de ejecución junto a la información del estudio en cuestión.

```
<ClinicalDocument xmlns="urn:hl7-org:v3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <typeId extension="POCD_HD000040" root="2.16.840.1.113883.1.3"/>
  <id extension="1.3.6.1.4.1.23650.4.3423.320100906093720000.3.2" root="1.2.276.0.24.0.5.0.1339551403.3.2"/>
  <code code="11528-7" codeSystem="2.16.840.1.113883" codeSystemName="LOINC" displayName="Reporte Radiologico"/>
  ....

```

Figura 36: XML no parseable por JAXB.

En Figura 37 y 38, se pueden ver partes de la estructura del CDA. En el primer caso el string utilizado en la implementación para generar parte del archivo CDA, y en el segundo la visualización del archivo XML obtenido.

```
"\t<recordTarget>\n"+
"\t\t<patientRole>\n"+
"\t\t\t<id extension=\"10087898-2\" root=\"RCICE^CI^EC\"/>\n"+
"\t\t\t<patient>\n"+
"\t\t\t\t<name>\n"+

/* given name */
"\t\t\t\t\t<given>\" + givenName + "</given>\n"+

/* family name */
"\t\t\t\t\t<family>\" + familyName + "</family>\n"+

"\t\t\t\t\t</name>\n"+

/* gender */
"\t\t\t\t\t<administrativeGenderCode code=\"\" + gender + \"\" codeSystem=\"2.16.840.1.

/* birth date */
"\t\t\t\t\t<birthTime value=\"\" + birthTime + \"\"/>\n"+

/* register number */
"\t\t\t\t\t<registerNumber value=\"\" + registerNumber + \"\"/>\n"+

/* patient imc */
"\t\t\t\t\t<imc value=\"\" + patientIMC + \"\"/>\n"+

"\t\t\t\t</patient>\n"+
"\t\t</patientRole>\n"+
"\t</recordTarget>\n"+

```

Figura 37: Representación interna del CDA.

Este String representa la implementación y representación a bajo nivel de lo que luego resulta en un archivo de Historia Clínica Electrónica CDA. Los datos dinámicos son concatenados junto a la metadata estática.

```

<recordTarget>
  <patientRole>
    <id extension="10087898-2" root="RCICE^CI^EC"/>
    <patient>
      <name>
        <given>Mario</given>
        <family>Torres</family>
        <suffix/>
      </name>
      <administrativeGenderCode code="M" codeSystem="2.16.840.1.113883.5.1"/>
      <birthTime value="Sat Mar 14 00:00:00 ART 1987"/>
      <registerNumber value="1122334455"/>
      <imc value="78"/>
    </patient>
    <providerOrganization>
      <id root="2.16.840.1.113883.19.5"/>
    </providerOrganization>
  </patientRole>
</recordTarget>
<author>
  <time value="2000040714"/>
  <assignedAuthor>
    <id extension="KP00017" root="2.16.840.1.113883.2.14.2.1"/>
    <assignedPerson>
      <name>
        <given>Jorge</given>
        <family>Barovero</family>
      </name>
    </assignedPerson>
    <representedOrganization>
      <id root="2.16.840.1.113883.2.14.2.1"/>
    </representedOrganization>
  </assignedAuthor>
</author>

```

Figura 38: Resultado en XML del CDA generado.

Una vez que la información es compilada y generada desde CINARTRO, se produce el archivo XML adecuado.

Otra de las características importantes del documento CDA es la utilización de imágenes en el contenido. En una primer solución, como salida de los estudios CINARTRO, las imágenes y el XML eran situados en un mismo directorio y dentro de la información de este último se añaden las referencias a las imágenes. Esta forma de referencia no es soportado para algunos sistemas. Para revertir esto, los datos se embeben en el XML transformándolos a un string en base64, una solución muy común para transformar multimedia a texto plano. La información de cada imagen simplemente es insertada dentro de un tag representativo y pasa a ser un dato más al igual que el resto de los elementos.

Un detalle respecto a esta solución, es que el tamaño del archivo final crece considerablemente, obviamente debido a la inclusión de las imágenes. Para dar una idea, un CDA que incluye las imágenes pesa alrededor de 7 MB.

Si bien esta es la solución actual en CINARTRO, otra de las opciones existentes para resolver el contenido de imágenes dentro del CDA es referenciar a las mismas a través de links accesibles desde internet que apunten a la ubicación en algún lugar de la “nube”. La gran ventaja de esto es que el CDA sigue siendo transportado simplemente a través de su par XML y que además es de contenido liviano al no contener los metadatos.

Una vez que CINARTRO genera un reporte CDA en base a un estudio realizado, comienza la etapa de guardado dentro del repositorio XDS. Este espacio nombrado en su larga expresión como *Cross Enterprise Domain Sharing*, es un sistema de estándares destinado al catálogo y compartimiento de registros de pacientes entre instituciones de salud.

La comunicación con este sistema se realiza a través de mensajes HTTP intercambiando mensajes bajo el protocolo SOAP. Precisamente esta forma de comunicación se caracteriza por la utilización de XML como tipo de datos para el contenido. Así, nuestro estudio CDA será agregado a estos mensajes para posteriormente ingresar al catálogo del repositorio.

A efectos de llevar a cabo nuestra implementación, se hace uso de los recursos publicados por Salud.uy que provee una interfaz de acceso a los métodos necesarios para el uso del XDS. Desde el punto de vista de implementación primero hubo que comprender la especificación de estos servicios para posteriormente invocarlos junto a nuestros datos.

Esta especificación, y en general todos los sistemas que funcionen utilizando el protocolo SOAP, provee un archivo XML denominado *Web Service Definition Language* (WSDL) que funciona como la documentación de los servicios publicados por el servidor describiendo de qué forma deben ser enviados los mensajes por los clientes para realizar las operaciones.

En el caso del repositorio, si bien existen varias operaciones que producen alteraciones sobre los datos existentes, podemos decir que contiene dos operaciones fundamentales. Una de ellas para ingresar un nuevo registro al sistema y la otra dedicada a la recuperación de información. Nuestro interés está enfocado en la primera de ellas ya que no es uno de los objetivos trabajar con la recuperación de registros para listarlos al usuario sino que simplemente se añaden para cada estudio realizado para eventualmente visualizarlos desde otro sistema externo.

El archivo WSDL nombra a la operación de ingresar nueva información como: **PROVIDEANDREGISTERDOCUMENTSET**, para la cual se exige un extenso contenido en el cuerpo de mensaje que permita identificar todos los aspectos involucrados en la transacción. Podemos encontrar entre los elementos más representativos a datos como: perfil del médico, paciente, entidad clínica, país donde se genera el contenido, e información sobre la jerarquía de los archivos, entre otros.

Finalmente para el envío del compilado se utilizó una librería nativa de java denominada SOAPXML, específica para manejo de comunicación mediante el protocolo SOAP. Cada uno de los datos del archivo final fueron añadidos bajo las operaciones provistas por la interfaz de la librería, la cual adicionalmente realiza el envío de información a través de internet.

Apéndice C

Contenido para Ingeniería de Muestra

Las siguientes imágenes fueron utilizadas durante la presentación de CINARTRO en el evento de Ingeniería de Muestra llevado a cabo entre los días 22 y 24 de Octubre de 2015. La primera de ellas (Figura 39), fue impresa en formato de póster y resume las principales características del proyecto. Se describen las personas involucradas y los logos de las instituciones, y dentro del contenido se mencionan los objetivos principales, una introducción desde el punto de vista de la lesión ocurrida, cómo son tomados los resultados para realizar un diagnóstico y también una mención sobre la aplicación de software.

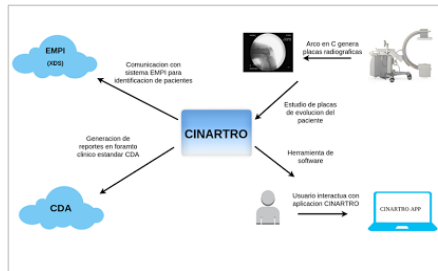
Por otra parte, las imágenes Figura 40 y 41, son las dos caras del folleto utilizado en la muestra para entregar al público interesado. Contiene una versión más extensa del resumen del proyecto respecto al póster, incluyendo entre otras cosas un resumen de las características del sistema.

CINARTRO: evaluación del ligamento cruzado anterior reconstruido en la rodilla de futbolistas

Marcio Rodríguez, Williams Olivera, Darío Santos, Franco Simini.

InCo – NIB – UIBLH – H.Clinicas

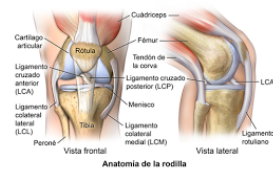
Ingeniería de Muestra, 22 - 24 de octubre 2015. Montevideo, Uruguay



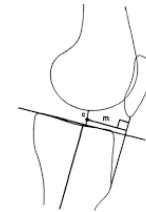
Objetivo

Determinación en forma interactiva del centro de rotación de la rodilla sobre imágenes fluoroscópicas. La finalidad del proyecto es implementar un equipo clínico integrando un aparato de RX portátil (arco en C) y su comunicación con el usuario. Generación de informe para la historia clínica electrónica en formato CDA y para PACS en forma DICOM. El proyecto se desarrolla en colaboración con la Unidad de Investigación en Biomecánica de la Locomoción Humana del Departamento de Rehabilitación (HC).

Anatomía y lesiones

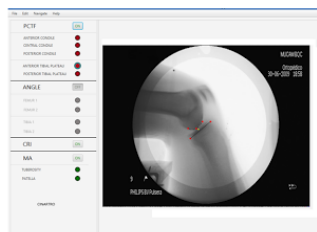
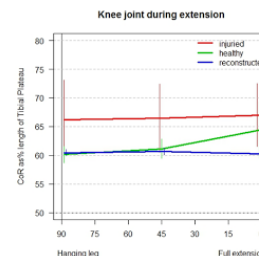


La rodilla es una de las mas complejas articulaciones del cuerpo humano, sustentada por varios tendones y ligamentos entre los que se encuentra el ligamento cruzado anterior (LCA). La rotura del LCA es una de las lesiones más comunes en los deportistas. La reparación requiere en la mayoría de los casos una operación que implica una recuperación de entre 1 y 2 años.



Diagnostico

A partir de la determinación del CRPC en cada una de las imágenes de una secuencia de videofluoroscopia del movimiento de la rodilla, se obtiene el desplazamiento de éste punto en todo el rango de movimiento de la rodilla. Dicho rango se determina en porcentaje del largo del platillo tibial. Los valores obtenidos ayudan a determinar el estado de los ligamentos cruzados luego de una operación y da un parámetro preciso para la comparación a lo largo de una recuperación.



Aplicación de Software

Cinartro es una herramienta de software que a partir de las imágenes obtenidas mediante rayos X y de modelos matemáticos ayuda a determinar el estado de recuperación de una rodilla tras una operación de LCA. El software está enfocado al uso clínico y genera información valiosa y precisa para el análisis de una rodilla afectada al exponer las diferencias existentes respecto a una rodilla sana. Genera un documento clínico estándar (CDA) con un reporte de los análisis realizados.



Figura 39: Póster de CINARTRO presentado en Ingeniería de Muestra.



CINARTRO

Evaluación del ligamento cruzado anterior reconstruido en rodilla de futbolistas

CINARTRO es un producto que permite determinar en forma interactiva el estado y evolución de una rodilla operada de ligamento cruzado anterior. Se determina el centro de rotación de la rodilla sobre imágenes de videofluoroscopia. La videofluoroscopia (VFC) es una serie temporal de imágenes de Rx de una estructura del cuerpo para su estudio.

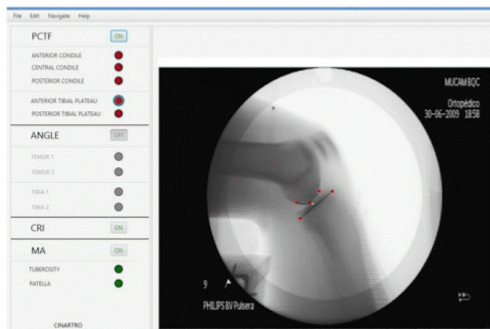
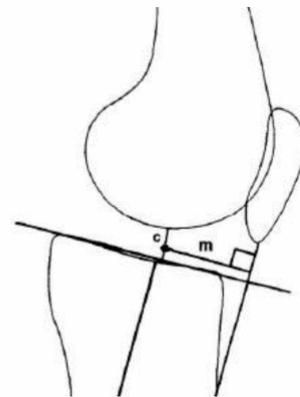


Imagen Rx de la rodilla capturada por **CINARTRO** y puntos elegidos por el operador

CINARTRO pone en práctica el método de determinación de parámetros de la mecánica articular (Centro de rotación instantáneo, centro de rotación por punto de contacto, brazo de palanca del cuádriceps) a partir de las imágenes de una secuencia de VFC del movimiento de la rodilla. El rango de desplazamiento de este punto a lo largo del movimiento de la rodilla es presentado como porcentaje del largo del platillo tibial. Los valores obtenidos ayudan a determinar la función dinámica del LCA, que se vuelve un parámetro preciso durante la recuperación.



Parámetros mecánicos: punto de contacto **c** y brazo de palanca **m**.

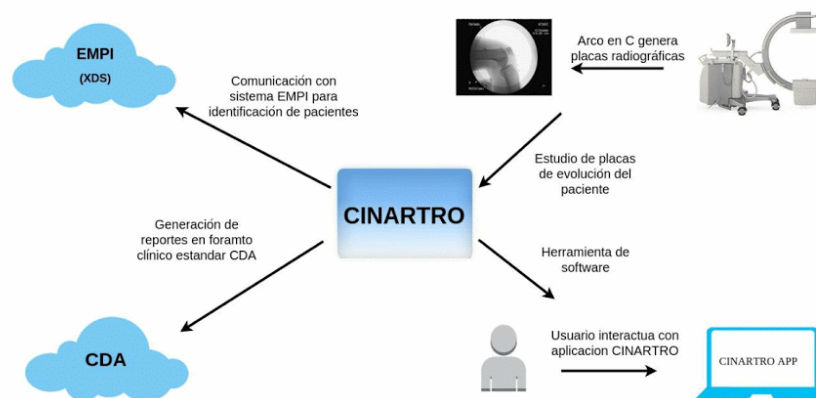
Rotura del Ligamento Cruzado Anterior

La rodilla es una de las articulaciones más complejas del cuerpo humano, con varios tendones y ligamentos entre los que se encuentra el ligamento cruzado anterior (LCA). La rotura del LCA es una lesión frecuente en los deportistas cuya reparación requiere una cirugía y posterior rehabilitación de la nueva estructura tendinosa colocada. No existe al día de hoy ningún método cuantitativo de evaluación durante el movimiento de la rodilla. **CINARTRO** es una propuesta de método clínico cuantitativo y de fácil uso.

Para procesar imágenes Rx **CINARTRO** aplica modelos geométricos y genera informes para la Historia Clínica Electrónica sobre el seguimiento de la rehabilitación de la rodilla afectada, comparándola con la rodilla sana. La identificación del paciente es consultada en el registro EMPI y la historia clínica almacenada en el repositorio XDS, ambos servicios de salud.uy

Figura 40: Folleto presentado en Ingeniería de Muestra.

A diferencia del póster, aquí se incluyen mayores detalles de los procedimientos y funcionalidades de **CINARTRO**.



CINARTRO en entorno clínico y servicios SALUD.UY: índice EMPI y repositorio de historias clínicas XDS

Características de CINARTRO

Parámetros de la mecánica articular determinados	- Punto de contacto tibiofemoral - Centro de rotación instantáneo - Brazo de palanca - Ángulo de flexión de la rodilla
Forma de interacción	El usuario marca puntos fiduciaros en cada imagen de la secuencia videofluoroscópica
Asistencia de marca de puntos	CINARTRO sugiere los puntos de las imágenes siguientes a las marcadas
Formato de entrada desde VFC	- Imágenes en formato BMP – PNG - Video en formato MP4
Formato de trabajo	Secuencias de imágenes BMP - PNG
Identificación de pacientes	Compatible con salud.uy índice EMPI (Repositorio XDS)
Seguimiento de la rehabilitación	Gráfica semestral durante el movimiento de la rodilla
Formatos de salida de historia clínica electrónica	- CDA, estándar de informes médicos - PDF
Exportación/Importación	- XML
Idiomas	- Múltiples con estructura simétrica - Español, inglés, portugués
Instalación	Standalone
Sistemas operativos	- Windows (versión 7+) - Linux (Ubuntu 14+) - MacOS (8+)

Desarrollado en colaboración con la unidad de investigación en biomecánica de la locomoción humana (UIBLH) de la Catedra de Rehabilitación del Hospital de Clínicas. Marcio Rodríguez y Williams Olivera, Prof. Franco Simini y Prof. Darío Santos - Octubre 2015

núcleo de ingeniería biomédica
de las Facultades de Medicina e Ingeniería de la Universidad de la República - Montevideo, URUGUAY
www.nib.med.edu.uy
Hospital de Clínicas piso 15 sala 2 11600 Montevideo URUGUAY +598 2487 1515 interno 2438

Figura 41: Folleto presentado en Ingeniería de Muestra.
Se listan el conjunto de características en detalle.

Apéndice D

Paper de CINARTRO

Las Figuras 42 a 46 muestran el *paper* de CINARTRO que está siendo realizado para su presentación en el Congreso Latinoamericano de Bioingeniería los días 26 al 28 de Octubre de 2016 [62]. El manuscrito contiene los aspectos más relevantes del producto y el trabajo realizado.

Knee Videofluoroscopy Kinematics Assisted Analysis Clinical Tool: CINARTRO

Williams Olivera¹, Marcio Rodriguez¹, Darío Santos², Franco Simini³

¹ Computer Engineering, University of the Republic, Montevideo, Uruguay.

² Department of Rehabilitation, Clinical hospital, University of the Republic, Montevideo, Uruguay.

³ Core Biomedical Engineering, Clinical hospital, University of the Republic, Montevideo, Uruguay.

Abstract— This work consists in the design and development of a desktop application in the Java language, to be used by doctors as a tool for studying the kinematics of the knee. X-ray images are taken from a videofluoroscopy and loaded into the software. For each image, the user places a set of points which determine the value of the Tibiofemoral Contact Point and Moment Arm. The software generates reports from the results in PDF, Excel and CDA format. CDA report is uploaded into an XDS repository, as part of the medical history of the patient. To face the problem of pincushion distortion in images, a gray tone recognition algorithm and distortion correction algorithm were implemented.

The software was tested for injured and healthy knee in 3 patients. A 1-minute average time was necessary to place all points in each image and precision errors introduced during the location of the points were detected.

The results of Tibiofemoral Contact Point show an average of 58.6% of displacement for healthy knee, and an average of 49% for injured knee. On the other hand, the Moment Arm average distance was 50.3 mm for healthy knee and 45.3 mm for injured knee.

Keywords— *Anterior Cruciate Ligament (ACL), Tibio-Femoral Contact Point (TFCP), Software, CDA.*

I. INTRODUCTION

The knee of a person is subjected in everyday life to continuous movements and strength exercises, so it is common the existence of instabilities in the expected evolution over a knee postoperative treatment, which ends resulting in delaying the recovery process.

One of the most common ways to verify the status of a reconstructed knee is through bending tests, or certain special moves around the knee, to identify inconsistencies and drawbacks. Many times, this depends on the perception and experience of each doctor; so to enhance and supplement this methodology, verification of results is proposed through quantitative methods.

As in every quantitative method, the aim of CINARTRO as a software program is the collection of numerical results that help doctors to determine the evolutionary condition of the patient after an ACL surgery.

The set of images obtained from a study by videofluoroscopy, is the source of input data to the software, which through a series of actions taken in conjunction with the user, allow to obtain relevant information as output. The value of using CINARTRO is complemented by the generation of reports that feed the patient's medical history and can be integrated into other information systems.

The procedure performed, pursues the theoretical methodology, mainly [1][2][3][4], internally implementing the geometrical and numerical calculations complexities. In particular, obtaining values for the TFCP in a flexion-extension movement has always been of great interest when studying the kinematics of the knee [1][2][5][6].

II. SPECIFICATION

The characteristics that define CINARTRO as an application software, address aspects covering various needs, including technical properties, usability, integration with external components, processed and generated data, among others. Table 1 summarizes the properties that form what we call the product specification, that is, the program features that have been defined a priori.

Table 1 Description of CINARTRO software specification.

Characteristic	Description
Calculus of interest	Value of TFCP and Moment Arm given a VFC image
Platform	Multiplatform: Windows, MacOS, Linux
Internationalization	Minimum: English, Spanish. easily add new languages
Target user	Medical knowledge, specially physiotherapist, exercise physiologist
External agents	EMPI [7], XDS [8], AGESIC [9], Salud.uy [10]
Output information	CDA [11] and PDF with clinical reports. Excel spreadsheets containing raw data

Figura 42: Paper CINARTRO. Página 1.

III. DESIGN AND ARCHITECTURE

Based on the properties determined in the specification, the design and architecture for CINARTRO is built. Figure 1 provides a graphic description of the components involved from a global point of view of the system around the software application.

The area of operation of CINARTRO, begins when the patient is subjected to X-ray shots using the C arc. The result of this activity produces a videofluoroscopy as output, showing a flexion-extension movement of the knee.

The media files of the videofluoroscopy are presented to the user as individual frames through the software application. There, the doctor must enter the patient data in question, which is then uniquely identified in the EMPI index. This ensures that the study data is related to a single individual.

Once the process is completed, a report in CDA format is created CDA for clinical history, which is published in an XDS repository in the cloud for later use.

Note that the repository EMPI, generating information in CDA format and the communication with the XDS are factors that make CINARTRO not an isolated software but, will join the channels for exchange of medical information and allows integration with other platforms and systems; being one of the most striking and interesting features.

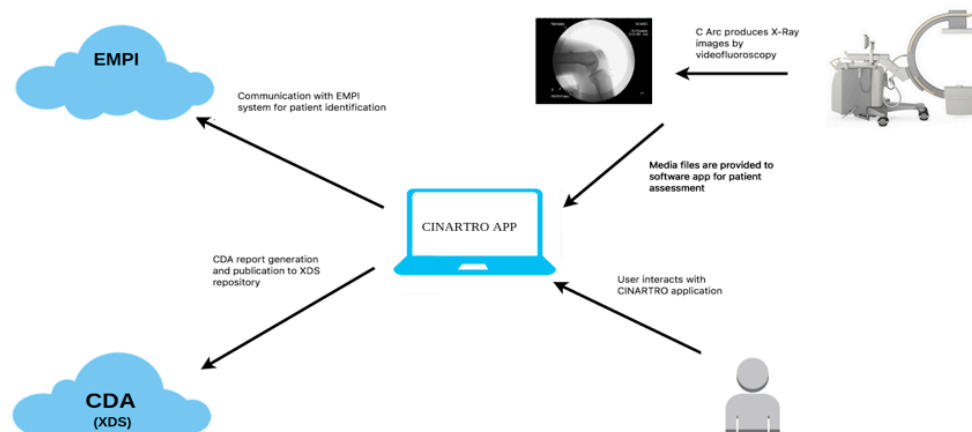


Fig. 1 Architecture components

Figura 43: Paper CINARTRO. Página 2.

IV. WORKFLOW INSIDE THE APPLICATION

In order to obtain results from videofluoroscopy images, the user must place a total of 11 points on each of the images that are presented. Thus,

CINARTRO will calculate the value of TFCP, the Moment Arm and the bending angle of movement for that state. This situation is reflected in Figure 2. Once processed all pictures, the user can view the fluctuation of these values through graphs.

As mentioned, the data can be obtained through Excel spreadsheets, PDF reports and CDA clinical reports that are automatically uploaded to an XDS repository. One of the most important features is the ability to compare different levels of evaluation, for example, studies of damaged knee, contralateral and rebuilt, which gives a better view of the results obtained.

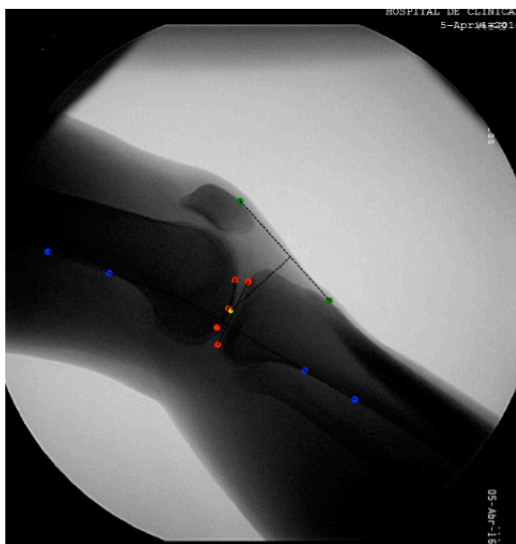


Fig. 2 Points marked by the user. Image taken from CINARTRO.

Is possible to apply the image distortion correction [1] by using a phantom. To do this, the user must upload the RX image of the phantom, under the same conditions that the videofluoroscopy was taken, and set the visible parts of the grid points, as seen in Figure 3, so that the software identifies the matrix points and can implement the corrections as explained in [1].

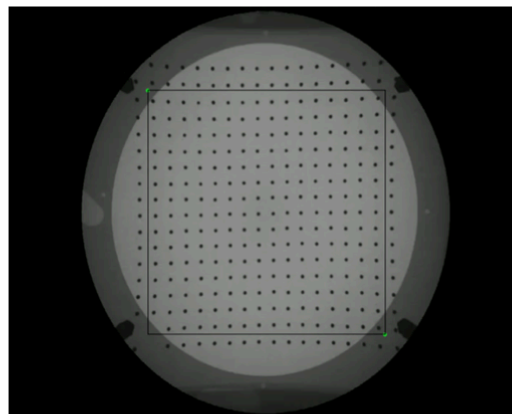


Fig. 3 Phantom configuration. Image taken from CINARTRO.

V. IMPLEMENTATION

From one perspective at a lower level, we present Table 2, with a compact of implementation features and libraries used to solve the various program activities. As general comments, it stands out the use of Java as the base language of the code and all libraries used, following the paradigm of Object Orientation. Another of the most important properties is the use of JavaFX framework that operates the graphics application components.

Table 2 Application features.

Feature	CINARTRO
Installation	Standalone (Desktop)
Programming Language	Java [12]
Graphic interface	JavaFX [13]
Graphic design tool	Scene Builder
Patient identification	EMPI (OpenEmpi) [7]
Studies output format	CDA (XML)
Math library	Apache Commons Math [14]
Data output / reports	Excel, PDF
PDF library	iText PDF [15]
Excel library	No, String CSV
XML management library	JAXB [16]
Video manipulation library	Xuggler [17]
HTTP messages exchange	HttpClient (Java Core)
Architecture patterns	POO, Singleton

The most important program classes represent instances of patient and physician as well, for which the Singleton design pattern is used to maintain a single instance of those classes. Other classes, maintain reference to images and sharp points on them. In addition, there are modules dedicated to generate reports and communication with external agent modules.

VI. TESTS AND RESULTS

It has been possible to meet the established proposals in the context of the specifications. While some features are subject to improvements, CINARTRO study achieves the goals initially set.

Tests based on videofluoroscopy images obtained from 3 patients were performed. The tests correspond to healthy knee and ALC injured knee for each patient.

The results of Tibiofemoral Contact Point show an average of 58.6% of displacement for healthy knee, and an average of 49% for injured knee. On the other hand, the Moment Arm average distance was 50.3 mm for healthy knee and 45.3 mm for injured knee. Tables 3 and 4 show the explained information.

Another of the significant variables in the calculations, is error correction through the use of a phantom. It stands, the result obtained with 0.235mm of error for a 40% utilization of phantom setting points, which is notoriously close to the value of 0.246mm, obtained by Baltzopoulos in [1].

The result also opens the way to new ideas and a great future work to improve the experience and increase the number of functionalities.

Table 3 Tibio femoral contact point results.

#Patient	100° - 110° (Flexion)%		135° - 145° %		170° - 180° (Extension) %		Average migration %	
	Healthy	Injured	Healthy	Injured	Healthy	Injured	Healthy	Injured
1	57	48	63	46	60	58	58	51
2	52	55	54	45	67	51	55	50
3	54	46	68	46	65	51	63	46

Table 4 Moment arm results.

#Patient	100° - 110° (Flexion)mm		135° - 145° mm		170° - 180° (Extension) mm		Average migration mm	
	Healthy	Injured	Healthy	Injured	Healthy	Injured	Healthy	Injured
1	43	43	47	51	44	46	45	48
2	43	46	51	57	41	54	47	52
3	44	47	45	54	43	49	44	51

Figura 45: Paper CINARTRO. Página 4.

REFERENCES

- [1] V. Baltzopoulos, "A videofluoroscopy method for optical distortion correction and measurement of knee-joint kinematics," *Clin. Biomech.*, vol. 10, no. 2, pp. 85–92, 1995.
- [2] E. Kellis and V. Baltzopoulos, "In vivo determination of the patella tendon and hamstrings moment arms in adult males using videofluoroscopy during submaximal knee extension and flexion," *Clin. Biomech.*, vol. 14, no. 2, pp. 118–124, 1999.
- [3] D. E. Tsaopoulos, V. Baltzopoulos, P. J. Richards, and C. N. Maganaris, "In vivo changes in the human patellar tendon moment arm length with different modes and intensities of muscle contraction," *J. Biomech.*, vol. 40, no. 15, pp. 3325–3332, 2007.
- [4] R. M. Erskine, C. I. Morse, S. H. Day, A. G. Williams, and G. L. Onambele-Pearson, "The human patellar tendon moment arm assessed in vivo using dual-energy X-ray absorptiometry," *J. Biomech.*, vol. 47, no. 6, pp. 1294–1298, 2014.
- [5] J. T. Bryant, H. W. Wevers, and P. J. Lowe, "One parameter model for error in instantaneous centre of rotation measurements," *J. Biomech.*, vol. 17, no. 5, pp. 317–323, 1984.
- [6] J. E. Giphart, C. a. Zirker, C. a. Myers, W. W. Pennington, and R. F. LaPrade, "Accuracy of a contour-based biplane fluoroscopy technique for tracking knee joint kinematics of different speeds," *J. Biomech.*, vol. 45, no. 16, pp. 2935–2938, 2012.
- [7] Sysnet International, "Open Enterprise Master Patient Index (OpenEMPI)," *Visto en Mayo 2016*. [Online]. Available: <http://www.openempi.org/>.
- [8] Integrating the Healthcare Enterprise (IHE), "Cross-Enterprise Document Sharing (XDS)," *Visto en Mayo 2016*. [Online]. Available: http://wiki.ihe.net/index.php/Cross-Enterprise_Document_Sharing.
- [9] Presidencia de la República, "Agencia de Gobierno Electrónico y Sociedad de la Información de Uruguay (AGESIC)," *Visto en Mayo 2016*. [Online]. Available: <http://www.agesic.gub.uy/>.
- [10] Agencia de Gobierno Electrónico y Sociedad de la Información de Uruguay (AGESIC), "SALUD.UY," *Visto en Mayo 2016*. [Online]. Available: <http://salud.gub.uy/>.
- [11] Health Level Seven International (HL7), "Clinical Document Architecture (CDA)," *Release 2.0, HL7 2005 Normative Edition*. 2005.
- [12] Oracle Corporation, "Java Software," *Visto en Mayo 2016*. v 8. [Online]. Available: <https://www.java.com/es/about/>.
- [13] Oracle Corporation, "JavaFX," *Visto en Mayo 2016*. v 8. [Online]. Available: <http://docs.oracle.com/javafx/>.
- [14] Apache Software, "Apache Commons Math," *Visto en Mayo 2016*. v 3.6.1. [Online]. Available: <https://commons.apache.org/proper/commons-math/>.
- [15] NV iText Group, "iText PDF," *Visto en Mayo 2016*. v 5.5.7. [Online]. Available: <http://www.itextpdf.com/>.
- [16] Oracle Corporation, "Java Architecture for XML Binding (JAXB)," *Visto en Mayo 2016*. v 2.2. [Online]. Available: <https://jaxb.java.net/>.
- [17] LLC ConnectSolutions, "Xuggle/Xuggler," *Visto en Mayo 2016*. v 5.2. [Online]. Available: <http://www.xuggle.com/xuggler/>.

Figura 46: Paper CINARTRO. Página 5.

Apéndice E

Curso Biomecánica de la Rodilla

En los días comprendidos entre las fechas 4 al 7 de abril de 2016 se llevó a cabo en el Hospital de Clínicas un curso denominado *Biomechanics of the Knee*. Expertos de cada área realizaron presentaciones referidas a la estructura de la rodilla desde diversos puntos de vista, tanto de la mecánica y locomoción, como de prótesis y placas radiográficas. Si bien no entraremos en profundidad en los temas, mencionamos una de las exposiciones realizadas que resultó de gran interés en el equipo así como la experiencia de presentar CINARTRO que también tuvo su espacio en el desarrollo del curso.

Una de las propuestas se centró en visualizar la compleja estructura puramente desde la mecánica. Se considera calcular la posición del Centro de Rotación Instantáneo (CRI) durante el movimiento articular de la rodilla.

En forma simplificada, la idea es considerar las velocidades como los factores de cálculo en el movimiento y obtener el centro de rotación como si fuera cualquier objeto con un eje fijo y otro en movimiento rotando a su alrededor. Así por ejemplo si un paciente realiza un movimiento de extensión, siendo el fémur la parte fija y la tibia rotando sobre él, se calcularían las velocidades obtenidas en las partes alta y baja de la tibia. Luego, el CRI para una posición dada está ubicado en la intersección de la recta totalmente paralela al fémur y la recta obtenida de unir las proyecciones de las magnitudes de las velocidades. En la Figura 42 vemos el bosquejo de lo mencionado.

Esto da muestra de lo amplio y complejo que puede resultar el estudio de la rodilla y cómo visualizar el mismo problema desde varias perspectivas, resulta en planteamientos realmente interesantes y considerables.

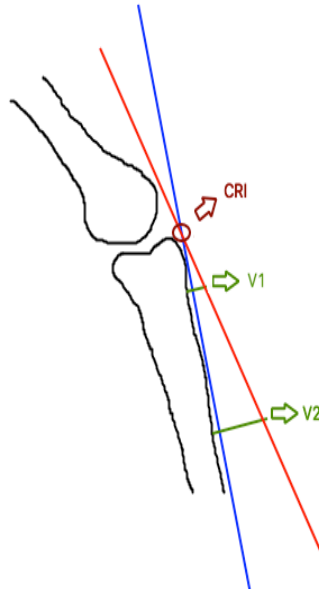


Figura 47: Cálculo del Centro de Rotación Instantáneo (CRI) utilizando velocidades.
Se considera el CRI como la intersección entre la línea paralela a la tibia y la línea que une las magnitudes de las velocidades de dos puntos en la tibia.

Durante el curso se realizó la presentación de CINARTRO, primero como propuesta teórica, dando ideas del trabajo ya existente en términos de investigación y luego exponiendo la aplicación de software realizada. Allí el programa fue instalado en la computadora de cada uno de los participantes del curso, beneficiando no sólo a la comprensión de los usuarios sino que también se dio paso a pruebas del estilo beta testing que permitieran descubrir mejoras y errores ya que aproximadamente 15 personas estaba haciendo uso de la aplicación simultáneamente. La experiencia fue muy enriquecedora ya que las preguntas realizadas, ideas y propuestas fueron extraídas por el equipo y luego terminarían afinando detalles en el software y diera algunas pautas para el trabajo a futuro.

Apéndice F

Gestión

El enfoque de este anexo se basa en describir de forma estadística la distribución de las horas de trabajo del equipo. Aclaremos primeramente que estos valores son tomados de un registro de horas que fue actualizado durante el desarrollo del proyecto. Si bien el trabajo en horas entre los miembros del equipo no fue igual para cada área, se promediaron las horas en cada caso con el objetivo de graficarlas como un elemento representativo sobre el total de horas.

En Figura 47 y 48 se presentan gráficos con la distribución de horas hombre y la dedicación en las distintas áreas de esfuerzo. La primera de ellas conteniendo los valores en horas y el segundo gráfico con una perspectiva de porcentajes entre las categorías. El total de horas promedio de cada integrante fue de aproximadamente 680 horas.

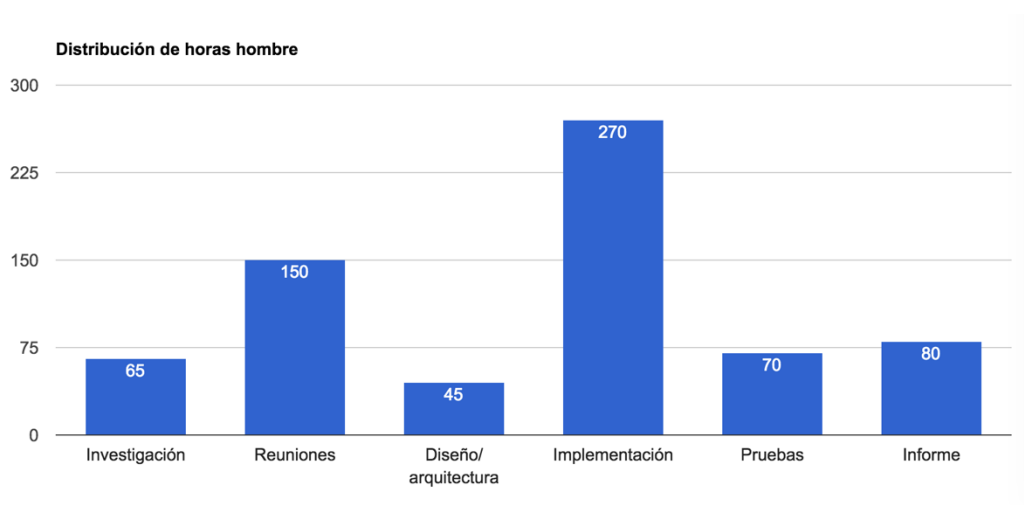


Figura 48: Gráfica del total de horas en promedio de los integrantes por área de trabajo.

Distribución en % de horas

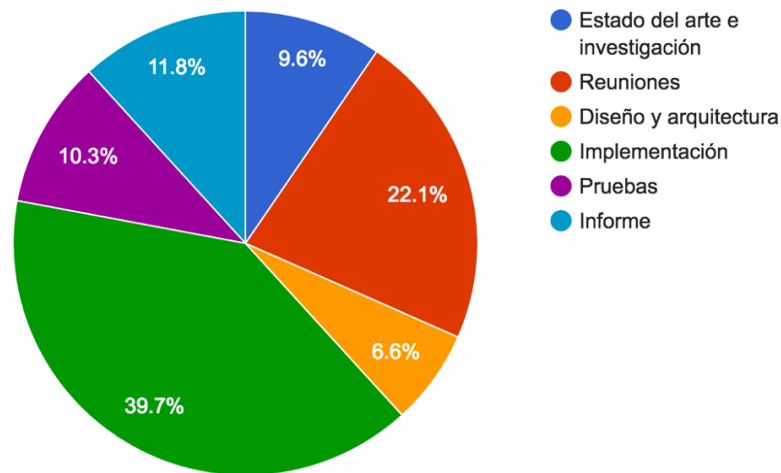


Figura 49: Distribución por área de trabajo.

A simple vista notamos que la mayor parte del tiempo empleado se produjo en la etapa de implementación con casi el 40% de horas totales. Esto remarca que el software como producto es una de las cosas más valorables de CINARTRO siendo el resultado de gran parte del esfuerzo. A través de la Figura 49, es posible observar la distribución de horas dentro del área de implementación. En mayor proporción encontramos las tareas relacionadas a los cálculos matemáticos, esto es, la lógica necesaria para el hallazgo de los puntos de interés.

Distribución de horas de implementación

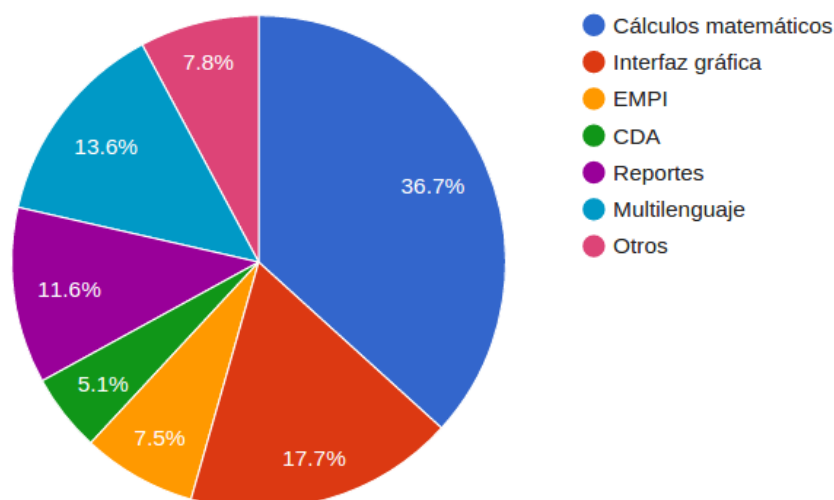


Figura 50: Distribución por categoría del esfuerzo en el área de implementación.

Por otro lado, se puede observar una considerable carga horaria en el área de reuniones, las cuales fueron fundamentales a lo largo de todo el proceso, manteniendo la organización de las prioridades y objetivos, y siendo el punto de partida de la mayoría de las decisiones, discusiones y puestas en común en el transcurso del tiempo. Esta área abarca las reuniones con tutores y reuniones de equipo así como también otras actividades como fueron una práctica para el aprendizaje sobre el índice EMPI, una charla sobre el repositorio XDS, la presentación de CINARTRO en Ingeniería de Muestra 2015 y la asistencia al curso de Biomecánica de la Rodilla en Abril de 2016.

El número restante en horas, se divide entre las demás etapas del proceso. Si bien no siguen un orden consecutivo en el transcurso del desarrollo, ya que algunas toman el foco de atención en varias partes, podemos identificar los aspectos de: estudio de estado del arte e investigación, que se da principalmente al comienzo del proyecto, y representa el análisis necesario para el entendimiento del problema y las bases para las siguientes etapas, esto a partir de lectura de papers y diversas publicaciones, las cuales fuesen descritas mayormente en el Capítulo 2. Este estudio del arte, que también se tuvo presente durante todo el proceso, se subsigue con el plan y definición de los requerimientos de CINARTRO, que pone en acuerdo los objetivos sobre los cuales se enfoca el resto del desarrollo.

Otra de las etapas identificables en cuanto a la dedicación de horas, es la realización de la arquitectura y diseño del sistema, en base a los requisitos definidos.

La última de las categorías son las pruebas realizadas. Tras lograr los primeros resultados de las etapas mencionadas, se realizaron múltiples pruebas tanto sobre el software como en términos cuantitativos, que permitieran comparaciones respecto a los objetivos iniciales.

Consideremos ahora, el análisis desde un punto de vista de la gestión, describiendo la forma en que el tiempo fue separado entre el equipo, las dificultades que surgieron y el orden de realización de las actividades.

Durante la primer etapa del desarrollo, el esfuerzo fue totalmente puesto en el estudio del estado del arte y la investigación de los conceptos desconocidos, ya que era necesario que se implantaran las bases necesarias para la comprensión de los aspectos teóricos en los cual se forma la idea de CINARTRO. Claramente, esto fue una de las situaciones en donde no era posible dividir el tiempo, y por lo tanto se presenta como uno de los cuellos de botella en el camino.

Lo mismo sucede para la definición de requerimientos y el planteamiento de la arquitectura del sistema, ya que las decisiones necesitaban de mutuo acuerdo y opinión, por lo que estos también fueron tiempos compartidos en el total de horas.

Respecto a las reuniones, aquellas que eran junto a los responsables, todos asistían a modo de puesta en común y muestra de avances, así como discusiones de problemas y planteamiento de ideas. Sin embargo, se paralelizaron las reuniones referentes a charlas, prácticas y clases que fueran específicas a un tema en particular, donde el mayor responsable de una tarea relacionada asistía para luego compartir experiencias y que a su vez permitiese poder avanzar en otros aspectos.

Al comenzar con la etapa de implementación, se contaba con una gran cantidad de trabajo que era necesario priorizar y administrar con el objetivo de optimizar el tiempo de desarrollo. Para ello, se utilizó la herramienta Trello [63], destinada precisamente a la organización de proyectos. Allí se crearon las tareas necesarias, con su responsable asignado, y un estado de progreso así como referencias y notas para la realización. De esta forma se hizo fácil mantener el orden de las cosas y evitar hacer re trabajo para los requerimientos.

De todos modos, la etapa de implementación no escapó a diversos obstáculos que necesitaron total dedicación del equipo, enfocado en una misma tarea, debido a su complejidad y también a la necesidad de cumplir con ciertas metas planeadas.

Finalizamos esta sección, con el siguiente diagrama de Gantt en Figura 50, que describe las diferentes etapas, las principales dependencias entre las actividades y resume con mayor claridad los aspectos de gestión descritos.

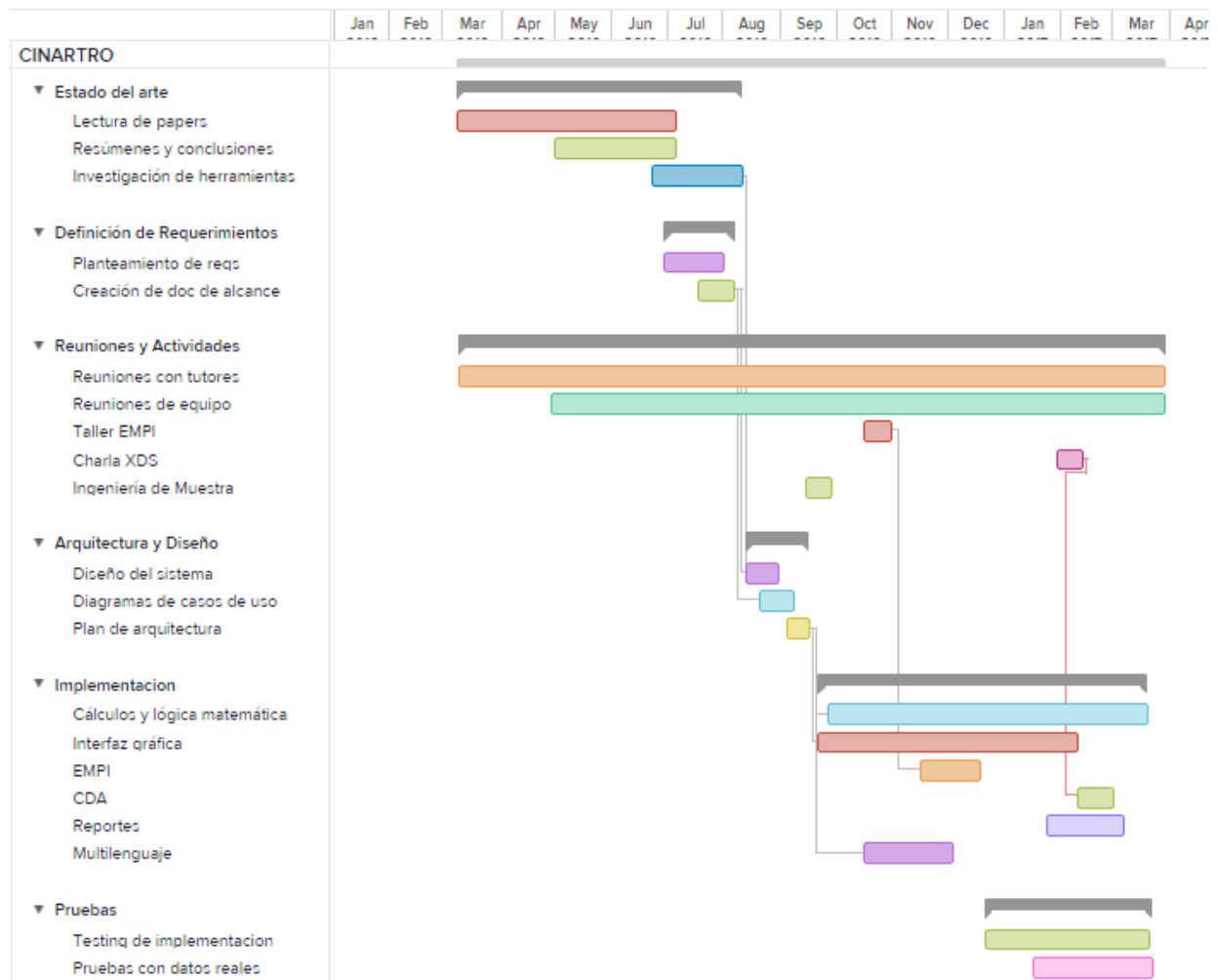


Figura 51: Diagrama de Gantt con la distribución de categorías de trabajo.

Apéndice G

Representación de clases

A continuación se muestra el conjunto completo de clases del sistema. Para cada una, se detallan los atributos así como la definición de los métodos. Se omitieron las conexiones entre las clases ya que las relaciones más importantes fueron descritas en el Capítulo 5.

application.cda::CinartroCDA
-xmlCDA: String -title: String -givenName: String -familyName: String -gender: String -birthTime: String -authorName: String -authorFamilyName: String -registerNumber: String -patientIMC: String -organization: String -diagnostic: String -notes: String -cArc: String -studyName1, studyName2, studyName3: String -studyDate1, studyDate2, studyDate3: String -image1, image2, image3, graphImage: String
+CinartroCDA(): ctor +export(File file): boolean +generateAndUploadCDA(): boolean -getCinartroData(): void -getStudy(int index): String -build(): void

Figura 52: Clase CinartroCDA.
Representación interna de los reportes clínicos electrónicos en formato CDA.

application.controller::CinartrolImageController
-instance = null: CinartrolImageController -images: List<CinartrolImage> -phantom: CinartroPhantom
-CinartrolImageController(): ctor +getInstance(): CinartrolImageController +getImages(): List<CinartrolImage> +setImages(List<CinartrolImage> images): void +addImage(String imageURI): void +addImageURI(String imageURI): void +getImageByIndex(int index): CinartrolImage +getImagesCount(): int +getCompletedImagesCount(): int +removeImage(CinartrolImage img): void +addPhantom(String imageURI): void +setPhantom(CinartroPhantom p): void +getPhantom(): CinartroPhantom +addVideo(File file, int numberOfVideoFrames): void +CRPCChart(boolean usePhantom): List<XYChart.Data<Number, Number> +momentArmChart(): List<XYChart.Data<String, Number> +momentArmCRPCChart(boolean usePhantom): List<Data<Number, Number> +imageItemsChart(boolean usePhantom): List<Data<Number, Number> +getDataForExcel(): ArrayList<HashMap<String, String>

Figura 53: Clase CinartroImageController.
 Manejador de las imágenes radiográficas que son posteriormente marcadas por el usuario.

application.controller::CinartroReportController
-instance = null: CinartroReportController -studies: List<CinartroReportStudy> -medical: CinartroMedical -patient: CinartroPatient -diagnostic: String -medicalHistory: String
-CinartroReportController(): ctor +getInstance(): CinartroReportController +setMedical(CinartroMedical medical): void +setPatient(CinartroPatient patient): void +getImages(): List<CinartroReportStudy> +addStudy(CinartroReportStudy s): void +getStudyByIndex(int index): CinartroReportStudy +getStudiesCount(): int +setData(String diagnostic, String medicalHistory): void +getMedicalHistory(): String +getMedical(): CinartroMedical +getPatient(): CinartroPatient +getDiagnostic(): String

Figura 54: Clase CinartroReportController.
 Manejador de reportes generados por CINARTRO.

application.controller::CinartroPersistenceController
-instance = null: CinartroPersistenceController -language: String -medicalInstitute: String -medicalName: String -medicalLastName: String -cArcModel: String -mainApp: MainApp
-CinartroPersistenceController(): ctor +init(MainApp mainApp): void +getInstance(): CinartroPersistenceController +save(): void +load(): void +setData(CinartroPersistenceController persistenceCtrl): void

Figura 55: Clase CinartroPersistenceController.
 Manejador de las operaciones relacionadas a persistir el estado de la aplicación.

application.controller::CinartroXMLController
+exportToXML(File fileToSave): void +importFromXML(File file): void +importFromXMLForReport(File file): void +importFromXMLold(Stage stage): void +exportToXMLold(Stage stage): void +savePreferencesToXML(): void +loadPreferencesFromXML(): void

Figura 56: Clase CinartroXMLController.
 Controlador que contiene la lógica para operar con los archivos XML.

application.controller::CinartroMedicalController
-instance = null: CinartroMedicalController -entity: CinartroMedical
-CinartroMedicalController(): ctor +getInstance(): CinartroMedicalController +getEntity(): CinartroMedical +setEntity(CinartroMedical patient): void

Figura 57: Clase CinartroMedicalController.
 Controlador Singleton que maneja la instancia del médico que hace uso de CINARTRO.

application.controller::CinartroPatientController
-instance = null: CinartroPatientController -patient: CinartroPatient
-CinartroPatientController(): ctor +getInstance(): CinartroPatientController +getPatient(): CinartroPatient +setPatient(CinartroPatient patient): void

Figura 58: Clase CinartroPatientController.
Clase Singleton encargada del manejo de Pacientes de CINARTRO.

application.controller::CinartroPdfController
-der, izq, sup, inf: float -tamanoDoc: Rectangle +font, fontbold, fontTitle, fontTable, fontDir, fontNote: Font
+CinartroPdfController(): ctor +generateReport(String pdfPath): boolean

Figura 59: Clase CinartroPDFController.
Clase que realiza el manejo y operaciones con archivos PDF.

application.controller::CinartroExcelController
+export(File file): boolean -build(): String

Figura 60: Clase CinartroExcelController.
Clase para el manejo de archivos Excel.

application.datatypes::AnglePointsDT
-anteriorFemur: Point.Double -posteriorFemur: Point.Double -anteriorTibia: Point.Double -posteriorTibia: Point.Double
+AnglePointsDT(): ctor +AnglePointsDT(Double anteriorFemur, Double posteriorFemur, Double anteriorTibia, Double posteriorTibia): ctor +getAnteriorFemur(): Point.Double +setAnteriorFemur(Point.Double anteriorFemur): void +getPosteriorFemur(): Point.Double +setPosteriorFemur(Point.Double posteriorFemur): void +getAnteriorTibia(): Point.Double +setAnteriorTibia(Point.Double anteriorTibia): void +getPosteriorTibia(): Point.Double +setPosteriorTibia(Point.Double posteriorTibia): void

Figura 61: Clase AnglePointsDT.
Este Datatype contiene los atributos y operaciones para el cálculo de los ángulos en las imágenes.

application.datatypes::SquareDT
-RealTopLeft: Point.Double -RealTopRight: Point.Double -RealBottomRight: Point.Double -RealBottomLeft: Point.Double
+SquareDT(): ctor +SquareDT(Double realTopLeft, Double realTopRight, Double realBottomRight, Double realBottomLeft): ctor +getRealTopLeft(): Point.Double +getRealTopRight(): Point.Double +getRealBottomRight(): Point.Double +getRealBottomLeft(): Point.Double +isInterior(int i, int j): boolean +isInterior(double i, double j): boolean +getGraficSquare(int screenWidth, int screenHeight): Rectangle +fromImgToGuiCoords(int width, int height): void

Figura 62: Clase SquareDT

Datatype con atributos y operaciones para crear coordenadas de trabajo sobre las imágenes.

application.datatypes::CorrectionPointsDT
~pointsInRealWorld: List<Point.Double> ~pointsInImage: List<Point.Double>
+CorrectionPointsDT(List<Double> pointsInRealWorld, List<Double> pointsInImage): ctor +getPointsInRealWorld(): List<Point.Double> +getPointsInImage(): List<Point.Double>

Figura 63: Clase CorrectionPointsDT

Este Datatype contiene la información necesaria para aplicar el algoritmo de corrección.

application.logic::CinartroRect
-m: double -n: double -isVertical: boolean -isHorizontal: boolean -A, B, C: double -p1, p2: Point.Double
+CinartroRect(Point.Double _p1, Point.Double _p2): ctor -CinartroRect(double coefM, Point.Double p, boolean isH, boolean isV): ctor +isVertical(): boolean +isHorizontal(): boolean +getP1(): Point.Double +getP2(): Point.Double +valuate(double x): double +getDistance(Point.Double p): double +getPerpendicularByPoint(Point.Double p): CinartroRect +intersection(CinartroRect other): Point.Double +getMiddlePoint(): Point.Double +angleBetweenRectsInDegrees(CinartroRect femur): double

Figura 64: Clase CinartroRect.

Representación interna de un Recta en CINARTRO.

application.logic::CinartroCore
-umbralChangeCoords = 6.5: double ~polin: UnivariateFunction ~min, max: double
+CinartroCore(): ctor -combine(double[] a, double[] b): double[] +getFemurInterpolate(): double[] +getSplinePoints(Point.Double femur1, Point.Double femur2, Point.Double femur3): List<Point.Double> -getListFemurInterpolate(UnivariateFunction polin, boolean rotate, int pointsQuantity): List<Point.Double> -invertPoint(Point.Double p): Point.Double +getCRPC_Rect(Point.Double femur1, Point.Double femur2, Point.Double femur3, Point.Double tibia1, Point.Double tibia2): CinartroRect +getMomentArm(CinartroRect tendon, Point.Double centro): CinartroRect +actualKneeAngle(Point.Double femur1, Point.Double femur2, Point.Double tibia1, Point.Double tibia2, boolean wiwhPhantomaPoints): double +getCRI(Point.Double tibiaAnterior1, Point.Double tibiaPosterior1, Point.Double tibiaAnterior2, Point.Double tibiaPosterior2): Point.Double +getReuleauxCRI(Point.Double tibiaAnterior1, Point.Double tibiaPosterior1, Point.Double tibiaAnterior2, Point.Double tibiaPosterior2): Point.Double +getCRPCPositionInPlateauLength(Point.Double femur1, Point.Double femur2, Point.Double femur3, Point.Double tibia1, Point.Double tibia2): double +getCRISkipping(CinartroImageController ic, CinartroPhantomValues values): List<String> -getFemurNearPoint(double minRango, double maxRango, UnivariateFunction polin, CinartroRect tibia): Point.Double

Figura 65: Clase CinartroCore.

Esta clase contiene la lógica que realiza los cálculos más importantes como el PCTF, BM y la representación de segmentos y splines.

application.logic::CinartroPhantomCore
-HSB_UMBRALE = 0.2: double -RECURSION_STEPS_LIMIT = 20: int -PIXELS_BETWEEN_UMBRALE = 10.0: double
+getPhantomaPoints(BufferedImage img, SquareDT sqDT, boolean fixRotation, double widthFit, double heightFit): CinartroPhantomValues -convertToBlackAndWhite(BufferedImage img, SquareDT sqDT): BufferedImage -getPointsUbication(BufferedImage img, double widthFit, double heightFit): List<List<Point.Double>> -insertPoint(Point.Double p, List<List<Point.Double>> matrix): void -insertInOrder(List<Point.Double> actualList, Point.Double p): void -getCloudPoints(int i, int j, boolean[] visitados, BufferedImage img, List<Point.Double> lista, int depth, List<Point.Double> actualPointColud): void -averagePoint(List<Point.Double> lista): Point.Double +correctPoint(Point.Double p, CinartroPhantomValues values): Point.Double -generate_X_matrix(List<Point.Double> calibrationPoints): double[][] -generate_real_world_coords(List<Point.Double> calibrationPoints, boolean X): double[]

Figura 66: Clase CinartroPhantomCore

Clase que contiene la lógica y operaciones relacionadas al Fantoma.

application.model::CinartrolImage
-imageURI: String -base64Image: String -edgeImageURI: String -posteriorCondile: Point.Double -anteriorCondile: Point.Double -centralCondile: Point.Double -posteriorTibialPlateau: Point.Double -anteriorTibialPlateau: Point.Double -patella: Point.Double -tuberosity: Point.Double -posteriorFemur: Point.Double -anteriorFemur: Point.Double -posteriorTibia: Point.Double -anteriorTibia: Point.Double -//All points settet isCompleted: boolean
+CinartrolImage(): ctor +CinartrolImage(String imageURI): ctor -setBase64Image(URL file): void +getBase64Image(): String +getImageFromBase64(): ByteArrayInputStream +getBytesFromBase64(): byte[] +getImageURI(): String +setImageURI(String imageURI): void +getEdgeImage(): ByteArrayInputStream +setEdgeImageURI(String edgeImageURI): void +getPatella(): Point.Double +setPatella(Point.Double patella): void +getTuberosity(): Point.Double +setTuberosity(Point.Double tuberosity): void +getAnteriorCondile(): Point.Double +setAnteriorCondile(Point.Double anteriorCondile): void +getCentralCondile(): Point.Double +setCentralCondile(Point.Double centralCondile): void +getPosteriorCondile(): Point.Double +setPosteriorCondile(Point.Double posteriorCondile): void +getAnteriorTibialPlateau(): Point.Double +setAnteriorTibialPlateau(Point.Double anteriorTibialPlateau): void +getPosteriorTibialPlateau(): Point.Double +setPosteriorTibialPlateau(Point.Double posteriorTibialPlateau): void +getAnteriorFemur(): Point.Double +setAnteriorFemur(Point.Double anteriorFemur): void +getPosteriorFemur(): Point.Double +setPosteriorFemur(Point.Double posteriorFemur): void +getAnteriorTibia(): Point.Double +setAnteriorTibia(Point.Double anteriorTibia): void +getPosteriorTibia(): Point.Double +setPosteriorTibia(Point.Double posteriorTibia): void +setPoint(PointTypes type, Point.Double position): void +getPoint(PointTypes type): Point.Double -setIsCompleted(): void +isCompleted(): boolean

Figura 67: Clase CinartrolImage.

Atributos y operaciones de cada imagen en CINARTRO. Contiene la información de los puntos marcados por el usuario.

application.model::CinartroPatient
-firstName: String -lastName: String -birthDate: Date -CI: String -address: String -phoneNumber: String -id: String -dateCreated: String -patientIMC: String -registerNumber: String +CinartroPatient(): ctor +CinartroPatient(String firstName, String lastName): ctor +CinartroPatient(String id, String firstName, String lastName, String CI, Date birthDate, String address, String phoneNumber, String dateCreated, String patientIMC, String registerNumber): ctor +setId(String id): void +getId(): String +getFullName(): String +getFirstName(): String +setFirstName(String firstName): void +getLastName(): String +setLastName(String lastName): void +getBirthDate(): Date +getBirthDateAsString(): String +setBirthDate(Date birthDate): void +getCI(): String +setCI(String ci): void +getAddress(): String +setAddress(String address): void +getPhoneNumber(): String +setPhoneNumber(String phoneNumber): void +setDateCreated(String dateCreated): void +getDateCreated(): String +getPatientIMC(): String +setPatientIMC(String patientIMC): void +getRegisterNumber(): String +setRegisterNumber(String registerNumber): void +firstNameProperty(): SimpleStringProperty +lastNameProperty(): SimpleStringProperty +CIProperty(): SimpleStringProperty +addressProperty(): SimpleStringProperty

Figura 68: Clase CinartroPatient.
Clase que contiene los datos del paciente.

application.model::CinartroMedical
-instituteName: String -medicalName: String -medicalLastName: String -patientHistory: String -currentStudyName: String -savedDate: String -cArcModel: String +CinartroMedical(): ctor +CinartroMedical(String instituteName, String medicalName, String medicalLastName): ctor +setSavedDate(): void +getSavedDate(): String +getInstituteName(): String +setInstituteName(String instituteName): void +getMedicalName(): String +setMedicalName(String medicalName): void +getMedicalLastName(): String +setMedicalLastName(String medicalLastName): void +getMedicalShortName(): String +setPatientHistory(String patientHistory): void +getPatientHistory(): String +getCurrentStudyName(): String +setCurrentStudyName(String currentStudyName): void +getcArcModel(): String +setcArcModel(String cArcModel): void

Figura 69: Clase CinartroMedical.
Clase que contiene los datos del médico que hace uso de CINARTRO.

application.model::CinartroReportStudy
-images: List<CinartrolImage> -phantom: CinartroPhantom -patient: CinartroPatient -medical: CinartroMedical
+CinartroReportStudy(): ctor +getImages(): List<CinartrolImage> +setImages(List<CinartrolImage> images): void +addImage(String imageURI): void +getImageByIndex(int index): CinartrolImage +getImagesCount(): int +removeImage(CinartrolImage img): void +getPatient(): CinartroPatient +setPatient(CinartroPatient patient): void +getMedical(): CinartroMedical +setMedical(CinartroMedical medical): void +getPhantom(): CinartroPhantom +setPhantom(CinartroPhantom phantom): void +CRPCCChart(boolean usePhantom): List<XYChart.Data<Number, Number>»

Figura 70: Clase CinartroReportStudy.

Contiene los datos que luego son mostrados en un reporte generado por CINARTRO.

application.model::CinartroPhantomValues
-//in mm DISTANCE_BETWEEN_PHANTOMA_POINTS = 10: double -//in pixels where 0,0 is in the middle (GUI coords) pointsInImage: Point.Double[][] -//in real (mm) world where 0,0 is buttom left pointsInRealWord: Point.Double[][] -columnsCount: int -rowsCount: int ~fixRotation: boolean ~imageWidth: int ~imageHeight: int
+CinartroPhantomValues(): ctor +CinartroPhantomValues(List<List<Point.Double>» columns, boolean fixRotation, int imageWidth, int imageHeight): ctor +printMatrixs(): void +getCorrectionPoints(Point.Double p): CorrectionPointsDT +phantomAverageError(): double +getPointCount(): int

Figura 71: Clase CinartroPhantomValues.

Clase que contiene la matrix de datos extraída de un Fantoma luego de aplicar el algoritmo de reconocimiento de grises.

application.model::CinartroPhantom
-imageURI: String -base64Image: String -values: CinartroPhantomValues
+CinartroPhantom(): ctor +CinartroPhantom(String imageURI): ctor +getImageURI(): String +setImageURI(String imageURI): void +getBase64Image(): String +setBase64Image(String base64Image): void +getImageFromBase64(): ByteArrayInputStream +getBytesFromBase64(): byte[] -setBase64Image(URL file): void +getValues(): CinartroPhantomValues +setValues(CinartroPhantomValues values): void

Figura 72: Clase CinartroPhantom.
Clase que contiene los datos de la imagen original del Fantoma.

application.utils::ScalingImage
-scrollOffset: Point2D -zoomPane: StackPane -scroller: ScrollPane -scrollContent: Group ~startClickX, startClickY: double -onMousePressedEventHandler: EventHandler<MouseEvent> -onMouseDraggedEventHandler: EventHandler<MouseEvent>
+ScalingImage(): ctor +getHScroll(): double +getVScroll(): double +createZoomPane(StackPane group): ScrollPane +figureScrollOffset(): Point2D +repositionScroller(double scaleFactor): void +moveImage(KeyEvent event): void

Figura 73: Clase ScalingImage.
Clase que realiza el manejo para la interacción de la imagen cuando es presentada al usuario.

application.utils::OpenEmpiConnector
-instance = null: OpenEmpiConnector -empiKey = "": String -baseUrl = "https://laboratorio.hcen.salud.uy/openempi-webapp-web-2.2.8/": String
-OpenEmpiConnector(): ctor +getInstance(): OpenEmpiConnector +getEmpiKey(): String -authenticateApp(): void +searchPatient(String name, String lastName): ArrayList<HashMap<String, String> +addPatient(String name, String lastName, String CI, String birthDate, String address, String phoneNumber, String registerNumber, String patientIMC): HashMap<String, String> +editPatient(String id, String name, String lastName, String CI, String birthDate, String address, String phoneNumber, String dateCreated): void +loadXMLFrom(String xml): Document

Figura 74: Clase OpenEmpiConnector.
Realiza la comunicación con el sistema EMPI.

application.utils::XDSRepositoryConnector
-instance = null: XDSRepositoryConnector -endpoint = "https://laboratorio.hcen.salud.uy/RepositorioXDSCAP/adocumentrepository_services": String -oidHospital = "39211000999139": String -oidMedical = "123456789": String
-XDSRepositoryConnector(): ctor +getInstance(): XDSRepositoryConnector +uploadCDA(String CDA): boolean -createSOAPRequest(String CDA): SOAPMessage -printSOAPResponse(SOAPMessage soapResponse): void -getOID(): String

Figura 75: Clase XDSRepositoryConnector.
Realiza la comunicación con el repositorio XDS.

application.utils::VideoThumbnails
-outputFilePrefix, inputFilename: String -mVideoStreamIndex = -1: int -mLastPtsWrite = Global.NO_PTS: long -MICRO_SECONDS_BETWEEN_FRAMES: long -imagesURI: ArrayList<String>
+getImages(String videoURI, double secondsBetweenFrames): ArrayList<String>

Figura 76: Clase VideoThumbnails.
Clase que realiza las operaciones para la extracción de imágenes a partir de un video.

Manual de Usuario

Para el uso del software Cinartro

2016

v 1.0

Autores

Marcio Rodriguez

Williams Olivera

Índice

Requerimientos	3
Inicio y pantalla principal	4
Configuración	5
Registrar un paciente	6
Carga de datos	7
Edición de puntos	9
Visualización de gráficas	13
Guardado y carga de estudios	14
Creación de reportes	15
Uso del fantoma	16

Requerimientos

Para un óptimo rendimiento de las funcionalidades de la aplicación, se recomienda que el computador utilizado, cumpla como mínimo con las características de hardware y de software para alguna de las opciones mostradas debajo.

Propiedad de hardware	Mínimo
Memoria RAM	3 GB
Procesador	1.8 GHz
Disco Duro	1 GB libre

Sistema Operativo	Mínima versión
Windows	7
Mac	Yosemite
Linux	Ubuntu 12.04 / Mint 17

Tipo de archivo	Extensión soportada
Video	mp4
Imágenes	.bmp .jpg .png

Se recomienda adicionalmente para una mejor experiencia, contar con un *mouse* durante el uso.

Inicio y pantalla principal

Como toda aplicación de escritorio, CINARTRO debe ser iniciado desde su correspondiente ejecutable, instalado previamente en el computador. Una vez iniciada la aplicación, veremos algo como la Figura 1.



Figura 1: Vista principal al iniciar la aplicación.

Esta es la pantalla principal, donde es posible iniciar todas las acciones dentro del programa. Se muestra la fecha y hora actuales así como la versión actual de la aplicación.

El contenido se divide principalmente en dos secciones; una de ellas es el panel de información, el cual resume los datos de lo realizado hasta el momento por el usuario y la otra sección muestra los accesos directos a las utilidades más importantes.

El panel de información de la izquierda se separa a su vez en tres partes que son: datos del paciente con el que se está trabajando, el médico que hace uso de la aplicación y el progreso en el estudio actual. Como es de esperar, inicialmente todo está vacío porque aún no hemos configurado ninguna información. En los siguientes apartados, se muestra cómo se completan esos datos que luego aparecerán en dichas secciones del panel.

Por otro lado, el menú de acciones ubicado al centro de la página, provee acceso directo a ocho funcionalidades claves dentro del programa que se detallan a continuación. Si bien estos iconos están disponibles solo en esta vista, la *Barra de Menú* al tope de la ventana, ofrece todas las acciones y navegaciones existentes, incluidas las mencionadas anteriormente.

Configuración

La mejor forma de comenzar la primera vez que se utiliza la aplicación, es configurando el perfil del usuario que va a realizar los estudios en el futuro. Esta información es de suma importancia ya que los datos del médico y la institución para la cuál se realiza el análisis, serán añadidos a los reportes generados una vez completado el flujo de trabajo. Esto sólo será necesario hacerlo la primera vez, o en caso que se desee cambiar el usuario.

Para realizar esta configuración, se debe acceder a la *Barra de Menú* => *Editar* => *Preferencias* donde veremos lo que se muestra en la Figura 2.



Informacion Medica	
Nombre Institucion	Hospital Clinicas
Nombre del Medico	Jorge
Apellido del Medico	Barovero
Modelo Arco en C	Philips Pulsera
<button>Guardar</button>	

Figura 2: Preferencias del usuario

Una vez completados los datos, se selecciona la opción *Guardar*.

Notar que existe una pestaña para cambiar el idioma en la aplicación, esto modifica todos los textos visibles en todas las pantallas. Sólo es necesario realizarlo una vez.

Si volvemos a la pantalla principal, utilizando *Barra de Menú* => *Navegar* => *Inicio*, veremos que la sección con la información del médico ha si actualizada con los datos ingresados.

Registrar un paciente

La información del paciente es de suma importancia en el programa, pues los estudios realizados posteriormente estarán ligados, así como al médico, a determinado paciente que es a quien pertenecen las imágenes evaluadas.

Cabe mencionar, que no necesariamente se está siguiendo un orden estricto, estas acciones para registrar a un paciente se podrían haber realizado al final de los otros procedimientos.

Desde la pantalla inicial, accionamos el botón de *Nuevo Paciente*, lo que nos lleva a un formulario como el que se muestra en Figura 3(a).

El médico deberá llenar como mínimo los campos mandatorios y luego guardar los datos.

Una vez hecho esto, pueden suceder dos cosas: o bien el flujo es redireccionado a la pantalla principal con un mensaje de éxito, o de lo contrario, sucede lo que se muestra en la Figura 3(b). Esto quiere decir que es posible que esa persona ya haya sido registrada previamente en la organización debido a que existen coincidencias en base a su nombre y apellido. Para este último caso, el usuario deberá decidir, eligiendo dentro del listado y considerando el resto de los datos, si se trata de la misma persona o si es un nuevo individuo.



El formulario, titulado "Registro de paciente", contiene los siguientes campos de entrada:

- *NOMBRE:
- *APELLIDO:
- *CI:
- F.NAC:
- DIRECCION:
- NUM TEL:
- N° REGISTRO:
- IMC:

En la parte inferior izquierda hay un botón "Guardar". En la parte inferior derecha, se indica "(*) OBLIGATORIO". A la derecha del formulario se encuentra un icono de una carpeta con un documento.

Figura 3(a): Formulario para registro de un paciente.



Registro de paciente

*NOMBRE:

*APELLIDO:

*CI:

F.NAC:

DIRECCION:

NUM TEL:

N° REGISTRO:

IMC:

(*) OBLIGATORIO

Estos pacientes coinciden con su búsqueda

Nombre	Apellido	CI	Direccion
Ismael	Alcoba	48585859	Bvar Artigas 1234

Figura 3(b): La información ingresada coincide con los datos de un paciente ya registrado.

Luego de elegida una de las opciones el usuario es finalmente llevado a la pantalla inicial, donde el panel izquierdo es completado con los datos del paciente.

Carga de datos

Las imágenes de rayos X son lo que dan sentido a la aplicación; por ello, para comenzar a evaluarlas, previamente deben ser cargadas al programa. Para hacerlo, existen dos opciones: en formato de video o como imágenes.

Para el primer caso, asumiendo que nos encontramos en la pantalla inicial, accionamos la opción de *Abrir Video* lo cuál muestra el explorador de archivos local para que el usuario seleccione el archivo correspondiente.

Cuando el video se ha elegido, se mostrará un cuadro de diálogo como el de Figura 4, pidiendo al usuario que ingrese el número de imágenes a extraer del video. CINARTRO se encarga internamente de guardar el número de *frames* elegidos de forma equidistante a la duración del video.

Si bien la elección es libre, se recomienda utilizar un número cercano a 30 para obtener mejores resultados.

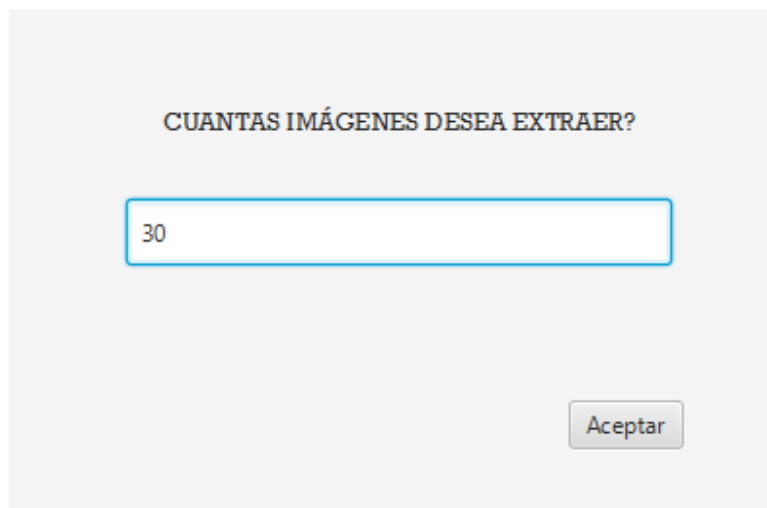
The image shows a light gray rectangular box containing a form. At the top, the text "CUANTAS IMÁGENES DESEA EXTRAER?" is centered in a bold, black, sans-serif font. Below this text is a white rectangular input field with a thin blue border. Inside the input field, the number "30" is displayed in a black, sans-serif font. To the right of the input field, there is a gray button with rounded corners and the word "Aceptar" in a black, sans-serif font.

Figura 4: El usuario debe ingresar el número de imágenes a extraer a partir de un archivo de video.

La segunda opción para carga de datos, es subir las imágenes directamente en caso que ya se tengan de forma individual. Esta vez elegimos la opción de *Abrir Imágenes* desde el menú y nuevamente veremos el explorador de archivos para seleccionar múltiples imágenes.

Luego de seleccionados los datos de entrada, ambas opciones conducen al usuario a la vista previa de imágenes. Ver Figura 5. Allí, el usuario deberá dar un nombre al *estudio* y tiene la posibilidad de realizar ciertas acciones.

Es posible añadir más imágenes con la acción de *Añadir Más*, lo que abre nuevamente el explorador local para seleccionar archivos adicionales.

También es posible reordenar las imágenes, utilizando movimientos de *drag and drop* sobre las vistas en miniatura. Idealmente la secuencia de imágenes debe seguir un orden creciente en cuanto al ángulo de movimiento, o sea, desde la flexión hacia la extensión, por ello se provee la opción de alterar el orden en caso que este tipo de comportamiento no esté reflejado al subir las imágenes.

Por último, es posible remover imágenes, simplemente seleccionando una o varias que se deseen eliminar del estudio y luego accionando el botón *Borrar*.

Una vez listo, se presiona *Continuar*.

Mencionemos, que el acceso a esta pantalla está siempre disponible desde la *Barra de Menú* así como también desde la pantalla principal.

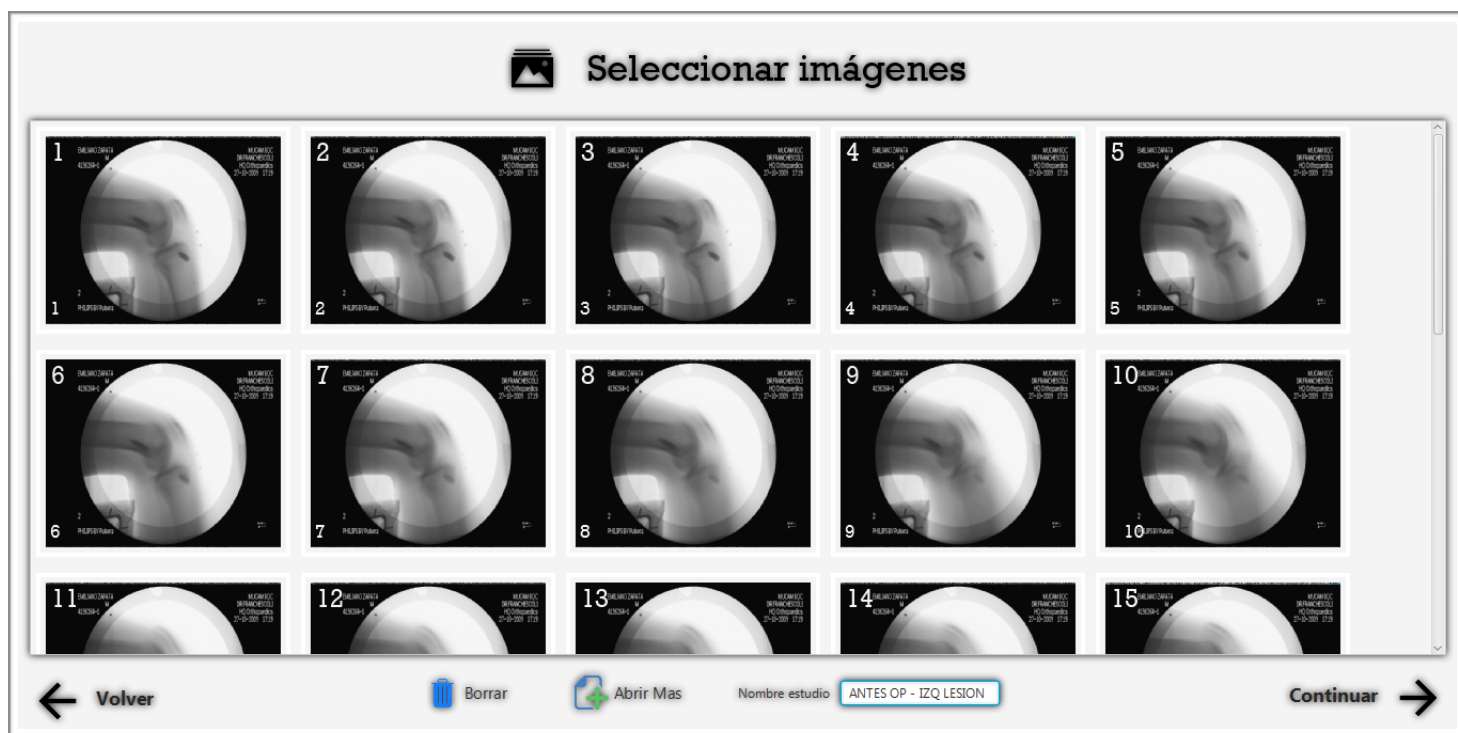


Figura 5: Vista previa de imágenes cargadas al programa.

Edición de puntos

Cuando las imágenes han sido confirmadas, el usuario es llevado a una nueva pantalla, como la que se muestra en Figura 6, donde realizará la mayor parte del trabajo. En la parte central de la vista, se presenta la imagen actual sobre la cual debe posicionar la serie de puntos. Mientras que en el panel izquierdo, se muestra la tabla de puntos a marcar, ciertos valores calculados y el número actual de la imagen.

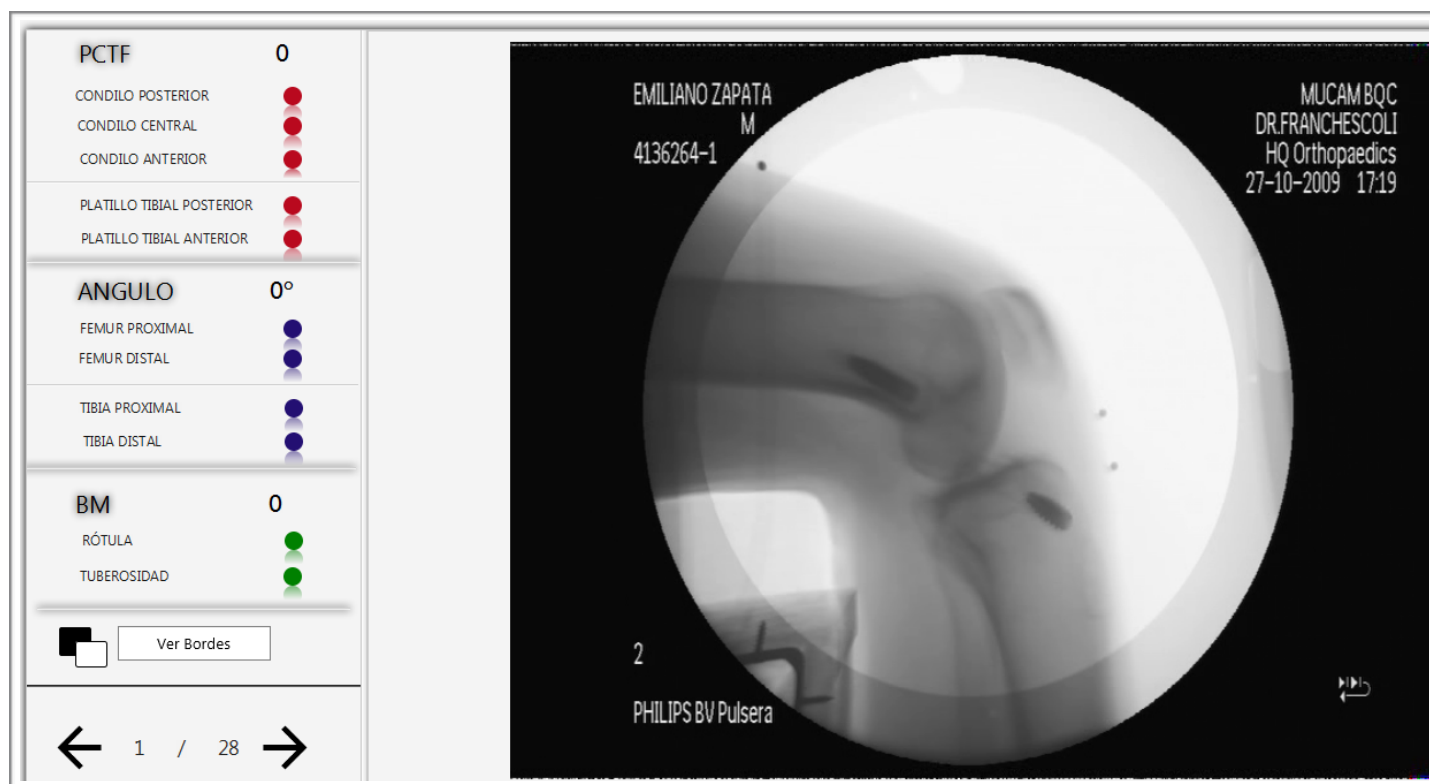


Figura 6: Pantalla de edición de puntos.

Para determinar un punto sobre la imagen, el usuario deberá primeramente seleccionarlo desde el panel, donde al hacer *click*, éste comienza a parpadear. Luego, también utilizando el *click del mouse* se selecciona el lugar exacto donde el punto debe ser colocado dentro de la imagen.

Esta acción debe ser realizada para cada uno de los 11 puntos del panel.

Se espera que el usuario, tenga conocimientos de la tarea que está realizando. Si bien el operador puede cambiar de un momento a otro, cada punto (descrito por su nombre) pertenece a un área dentro de la imagen sobre la que se espera sea posicionado.

Las Figuras 7(a), 7(b) y 7(c), ilustran mediante un ejemplo la posición “ideal”, dada una imagen, que se espera para cada uno de los puntos.

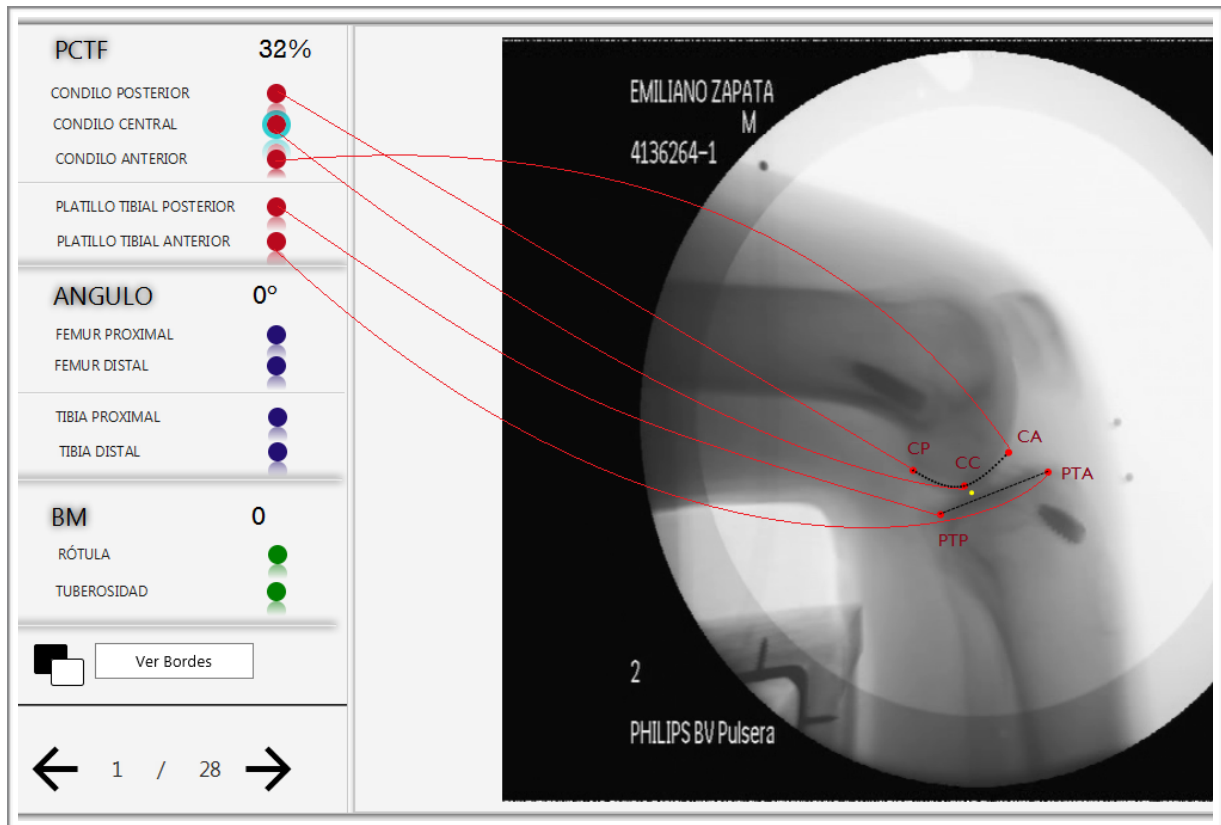


Figura 7(a): Puntos (rojos) que determinan el valor del PCTF

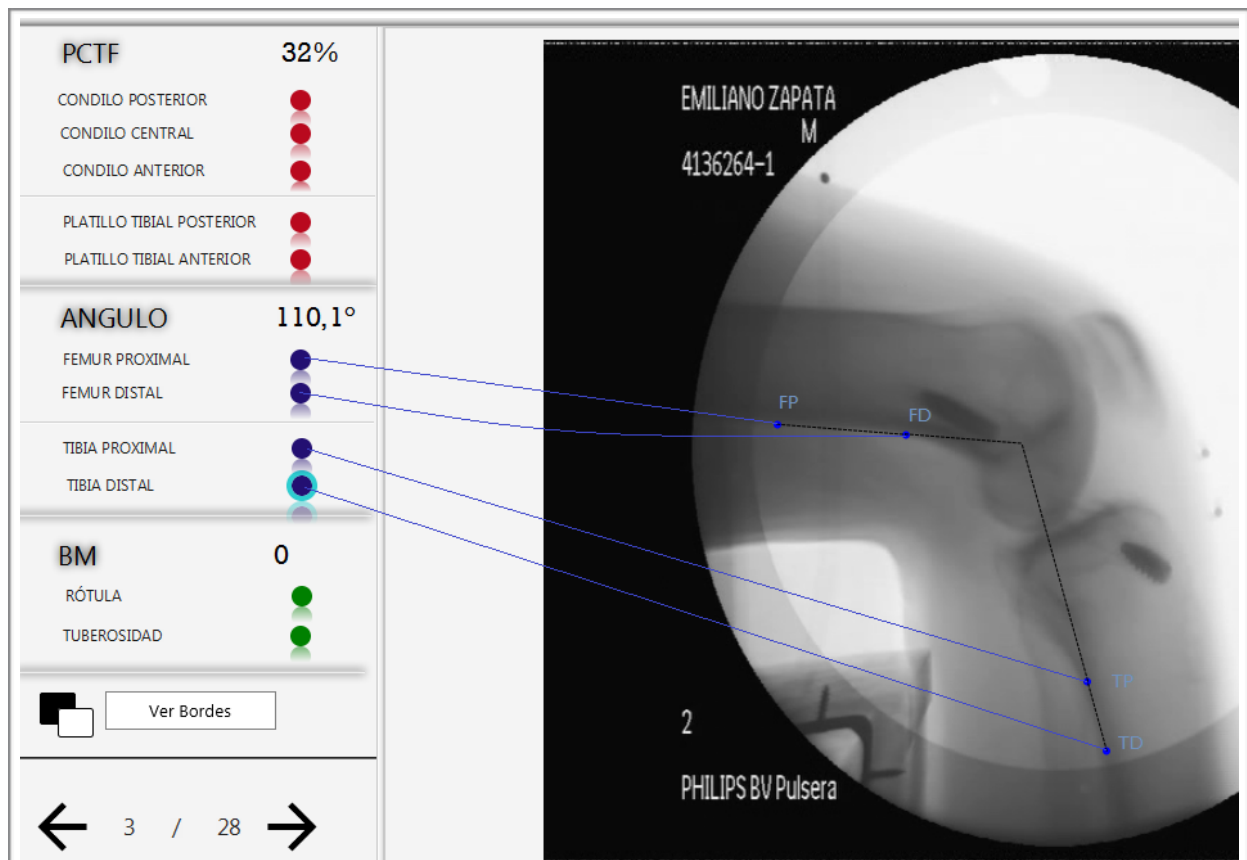


Figura 7(b): Puntos (azules) que determinan el valor del ángulo de flexión.

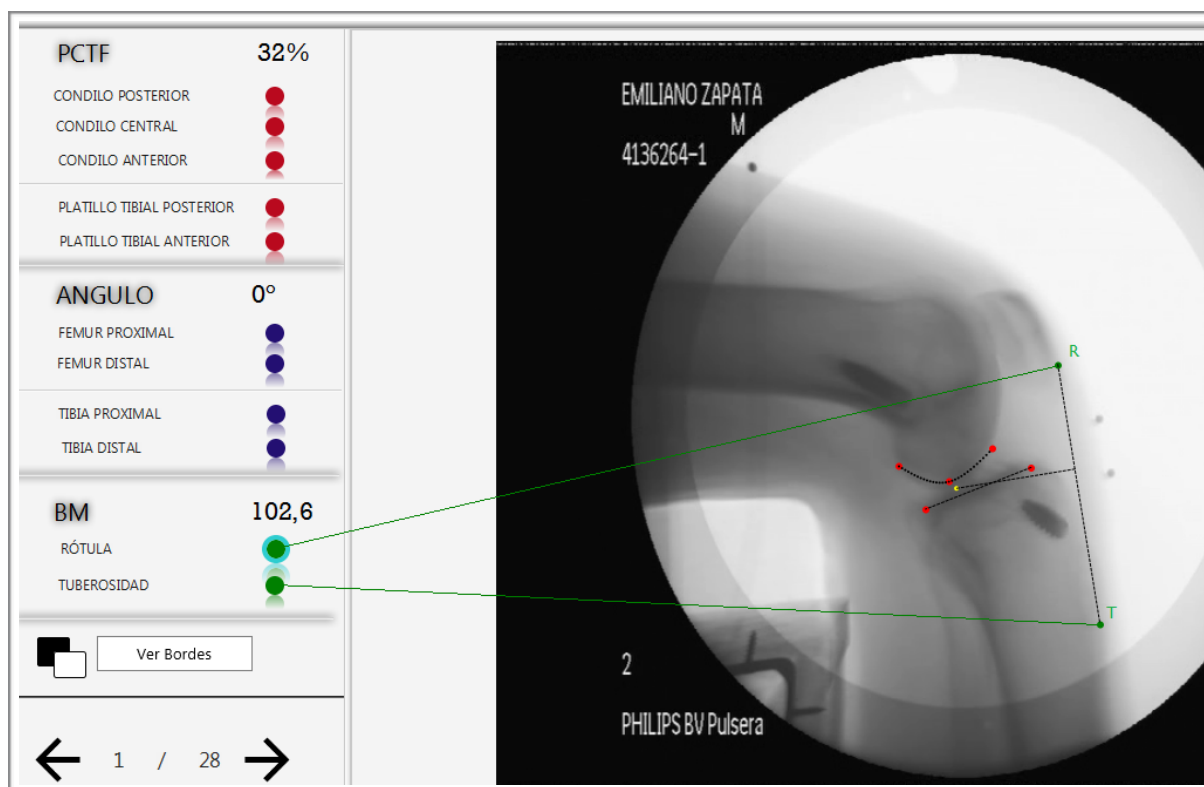


Figura 7(c): Puntos (verdes) que junto al PCTF determinan el valor del brazo de momento.

A medida que los puntos son ubicados, los cálculos correspondientes se van realizando, así por ejemplo los puntos rojos permiten calcular el valor del *PCTF* (*Punto de contacto tibio-femoral*), el cual es mostrado con un punto amarillo sobre la imagen. Los puntos azules, permiten calcular el *ángulo* de extensión en el que se encuentra la pierna, y los puntos verdes el valor del *Brazo de Momento*.

Una vez que todos los puntos son ubicados, se ve algo como lo mostrado en Figura 8. Luego, el usuario puede pasar a la siguiente imagen utilizando las *flechas del panel*. La nueva imagen, a diferencia de la primera, contiene los puntos seleccionados de la imagen anterior a modo de guía para su nueva configuración de puntos. Se espera, que de una imagen a otra, la posición de los puntos varíe muy poco, por lo que el usuario, sólo deberá mover mínimamente los puntos de guía.

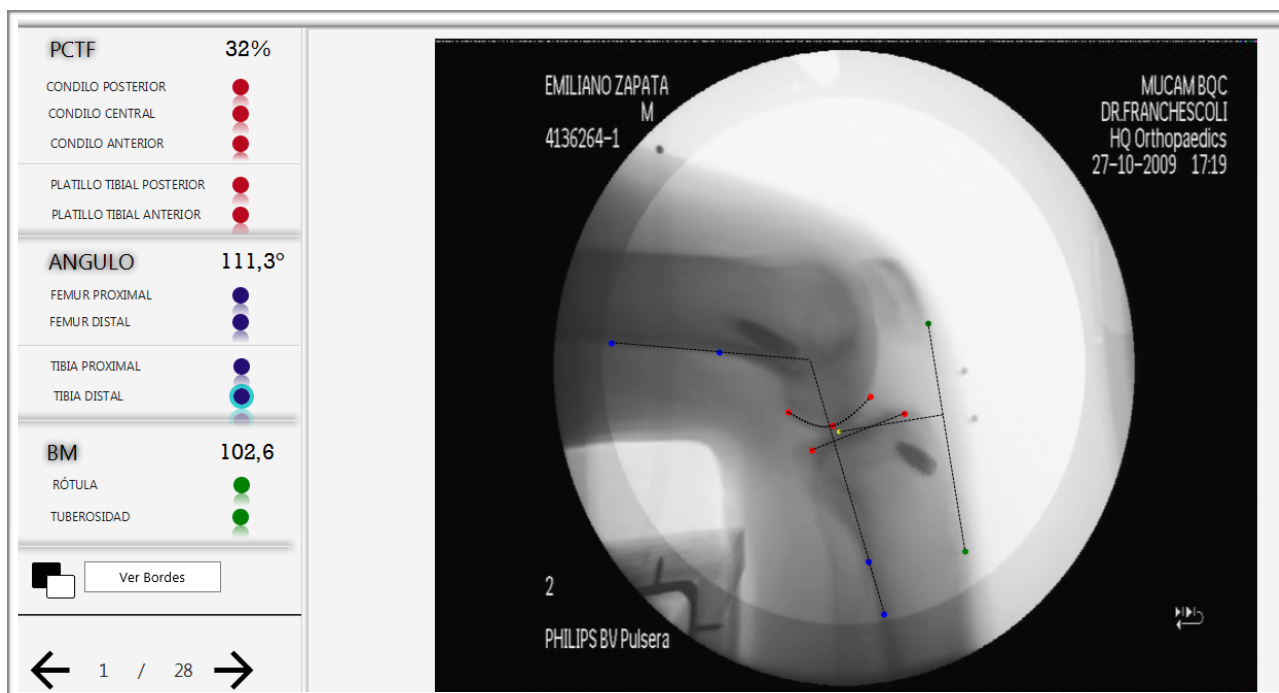


Figura 8: Vista de una imagen luego que todos los puntos han sido ubicados.

Para mayor comodidad, el usuario podrá mover un punto (luego de que es seleccionado desde el panel) utilizando las flechas de su teclado *A*, *S*, *D* y *W*, haciendo que cada punto se traslade muy lentamente en cualquier sentido para así lograr mejor precisión.

Otras herramienta disponibles, son el *zoom* en el área de la imagen; que se acciona utilizando el *scroll* del mouse donde el usuario podrá acercar y alejar la imagen en caso que algunas zonas no sean totalmente nítidas a simple vista. También está la utilidad de activar la opción de *contraste* sobre la imagen, donde se aplica un efecto sobre la misma que permite resaltar los bordes.

Si volvemos al menú principal, veremos que la sección de estudio ahora contiene información que muestra el progreso en la edición de puntos, detallando el nombre del estudio así como el total de imágenes y el número de completadas sobre el total.

Es posible regresar a la última imagen editada desde el menú central.

Visualización de gráficas

Una vez que los puntos fueron configurados, tiene sentido visualizar cómo ha sido la variación de los valores de interés a lo largo del total de imágenes. Para ello, desde el menú central o desde la *Barra de Menú*, podemos navegar hacia la visualización de gráficas. Allí, veremos algo como en Figura 9, donde se muestran tres gráficos diferentes: el superior, muestra el valor del *PCTF* respecto del ángulo; el inferior izquierdo, muestra el valor del *brazo*

de momento respecto al ángulo, y el inferior izquierdo, el valor de cada ángulo en cada imagen. Este último, se espera que sea siempre creciente, mostrando el movimiento de flexión-extensión.

Accionando el *click* sobre cualquiera de las gráficas, se muestra una nueva pantalla con dicho gráfico en toda la pantalla para visualizarlo con mayor detalle.

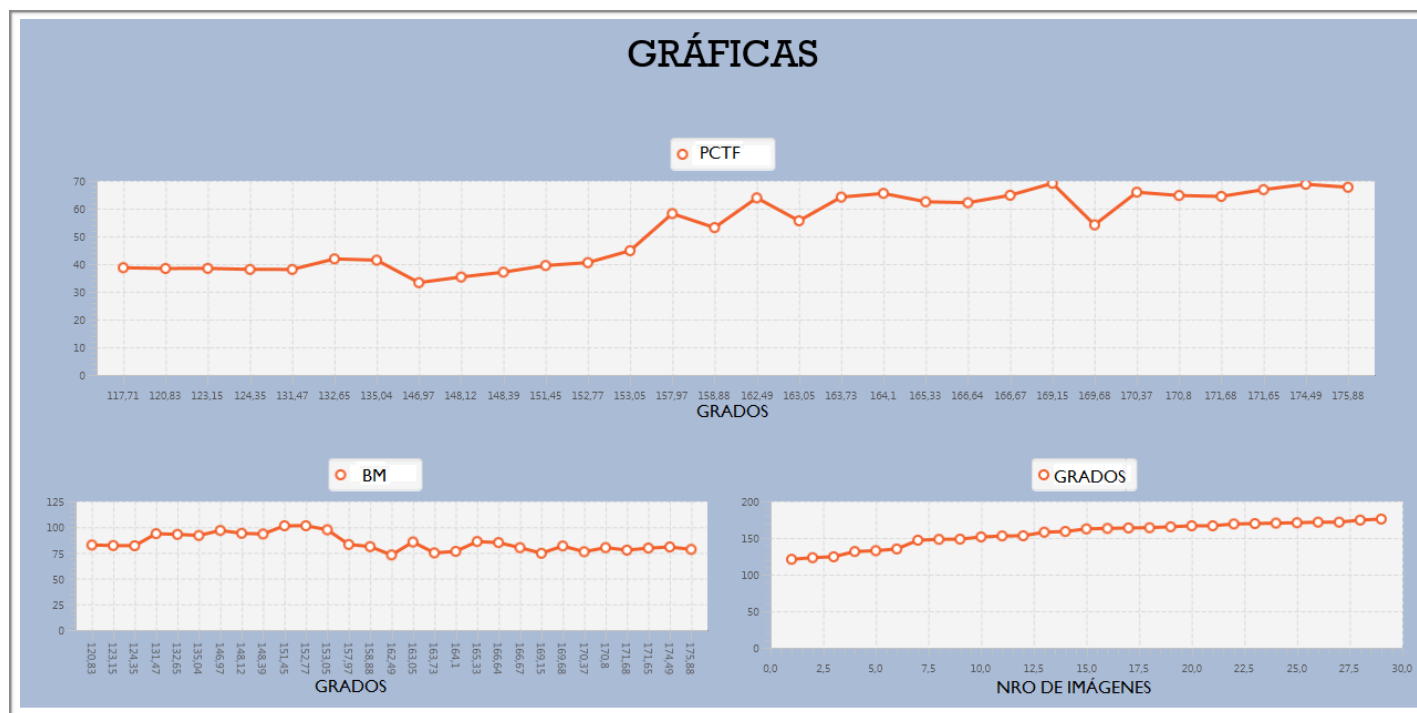


Figura 9: Visualización de gráficos a partir de los valores calculados.

Guardado y carga de estudios

CINARTRO permite guardar el avance de lo realizado en el programa, persistiendo la información necesaria para volver al estado actual en el futuro.

Para guardar los cambios, ir a *Barra de Menú* => *Archivo* => *Guardar*, y luego seleccionar la dirección de destino.

De manera similar, desde *Barra de Menú* => *Archivo* => *Abrir*, se podrá seleccionar un estudio existente, y la aplicación se actualizará mostrando los datos allí guardados.

Creación de reportes

El usuario puede exportar la información de los estudios de diferentes formas. Una de ellas, es hacerlo como una planilla Excel, que contiene los datos crudos con los valores de los puntos de interés para todas las imágenes. Para exportar un archivo excel, simplemente hay que seleccionar la opción de *Exportar Excel* desde el menú principal y elegir una ubicación de destino.

Otra opción, es realizar un reporte en formato PDF con la información más relevante del estudio realizado. Es posible incluso, realizar comparaciones de hasta 3 estudios diferentes. En el archivo resultante, se puede ver la información del médico, paciente, notas de diagnóstico, 3 imágenes de cada estudio (se toman la primera, la última y una del medio), la gráfica conteniendo los valores de variación del *PCTF* en cada estudio y un espacio para la firma del médico, en caso que el reporte quiera ser utilizado como registro.

Para lograr esto, desde el menú principal, se elige la opción de *Nuevo Reporte*, lo cual lleva a una nueva pantalla donde el usuario deberá elegir el archivo de un estudio previamente guardado. Cuando este es subido, se tiene una vista previa con las imágenes seleccionadas por el sistema como se ve en Figura 11.



Figura 11: Creación de un nuevo reporte. Selección de estudios previos.

Luego, el usuario podrá opcionalmente añadir estudios adicionales, en cuyo caso será para “compararlos” entre sí o de lo contrario, finalizar la carga de archivos.

El siguiente paso es simplemente para visualizar la comparación de gráficas, que será agregada al reporte final. Al dar siguiente, se presenta un formulario con la información, ya existente, del médico y paciente. Ver Figura 12. Allí el usuario, tiene la posibilidad de escribir notas sobre el reporte así como comentarios de diagnóstico.

NUEVO REPORTE

Paciente		Medico	
*NOMBRE:	Ismael	*NOMBRE:	Jorge
*APELLIDO:	Alcoba	*APELLIDO:	Barovero
*CI:	48585859	*INSTITUTO:	Hospital Clinicas
IMC:	87		
N° REGISTRO:	1122334455		
F.NAC:	2016-02-18		
DIAGNOSTICO:			

Historia clínica

Guardar

Figura 12: Resumen del reporte. Se habilitan campos para ingresar historia clínica y diagnóstico.

Cuando todo es confirmado, el operador podrá seleccionar dónde guardar el PDF. Además, toda esta información es compilada en un reporte de formato CDA clínico estándar, el cual es subido, paralelamente a la generación del PDF, a un repositorio XDS para mantener registro de la historia clínica.

Uso del fantoma

Otra de las funcionalidades existentes en CINARTRO, es la corrección de errores de distorsión en las imágenes de rayos X mediante la utilización de la imagen de un fantoma. Accediendo a *Barra de Menú => Archivo => Abrir Fantoma*, se puede seleccionar una imagen de fantoma como la de Figura 13.

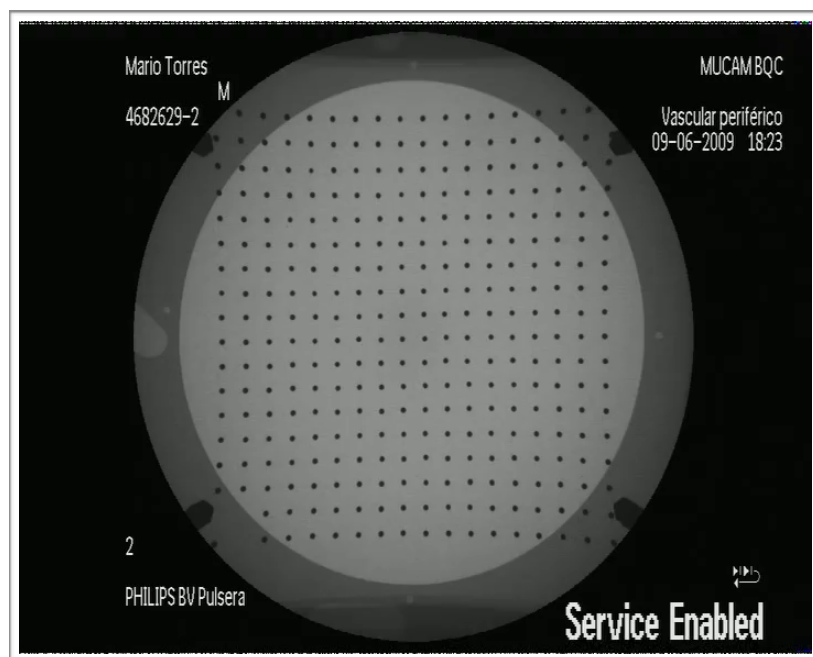


Figura 13: Ejemplo de imagen fluoroscópica de un fantoma.

Posteriormente, se presenta la pantalla mostrada en Figura 14, y como las instrucciones lo indican, el usuario deberá marcar dos puntos sobre la imagen, de forma que el recuadro que los une, contenga la mayor cantidad de puntos visibles posible. De esta forma se evitan las zonas distorsionadas o demasiado oscuras de la imagen. CINARTRO reconoce la grilla de puntos y guarda los valores internamente.

Esta información del fantoma, es utilizada para corregir los valores en los reportes generados.

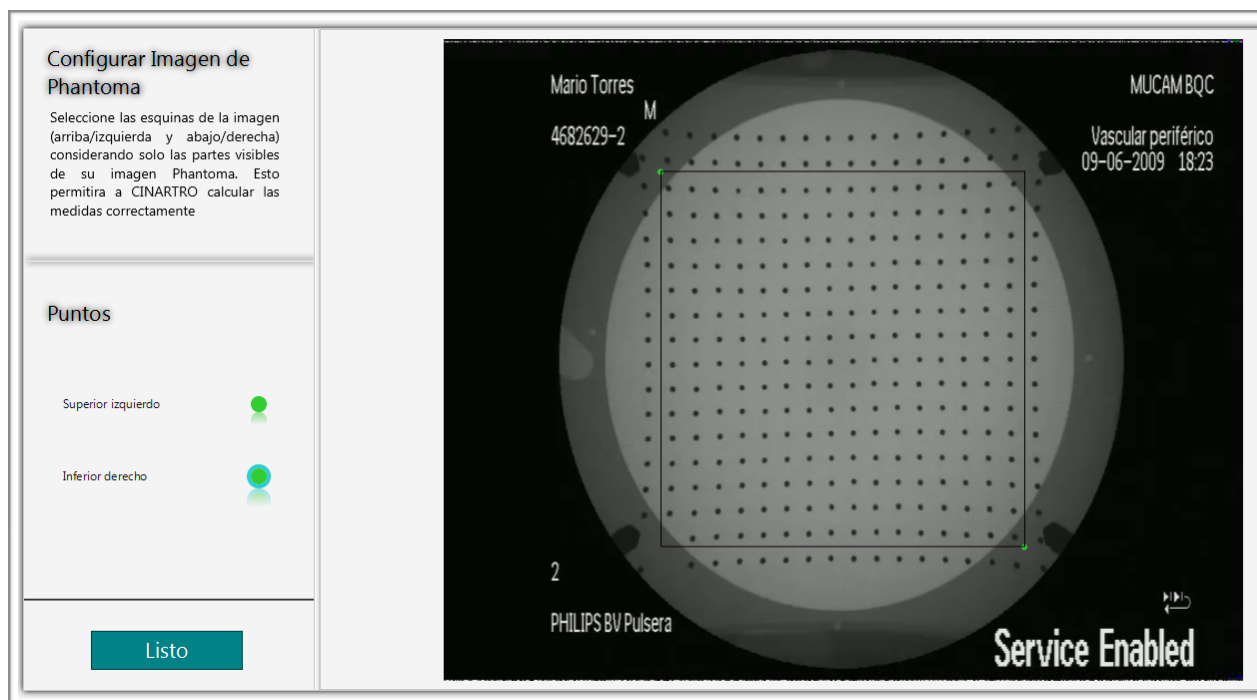


Figura 14: Selección de puntos visibles del fantoma para posterior corrección de la distorsión.