



Universidad de las Ciencias Informáticas.

Facultad: 5

Título: KINETIC, un nuevo enfoque de la evaluación postural mediante realidad aumentada.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autor:

José Luis Simón Peña.

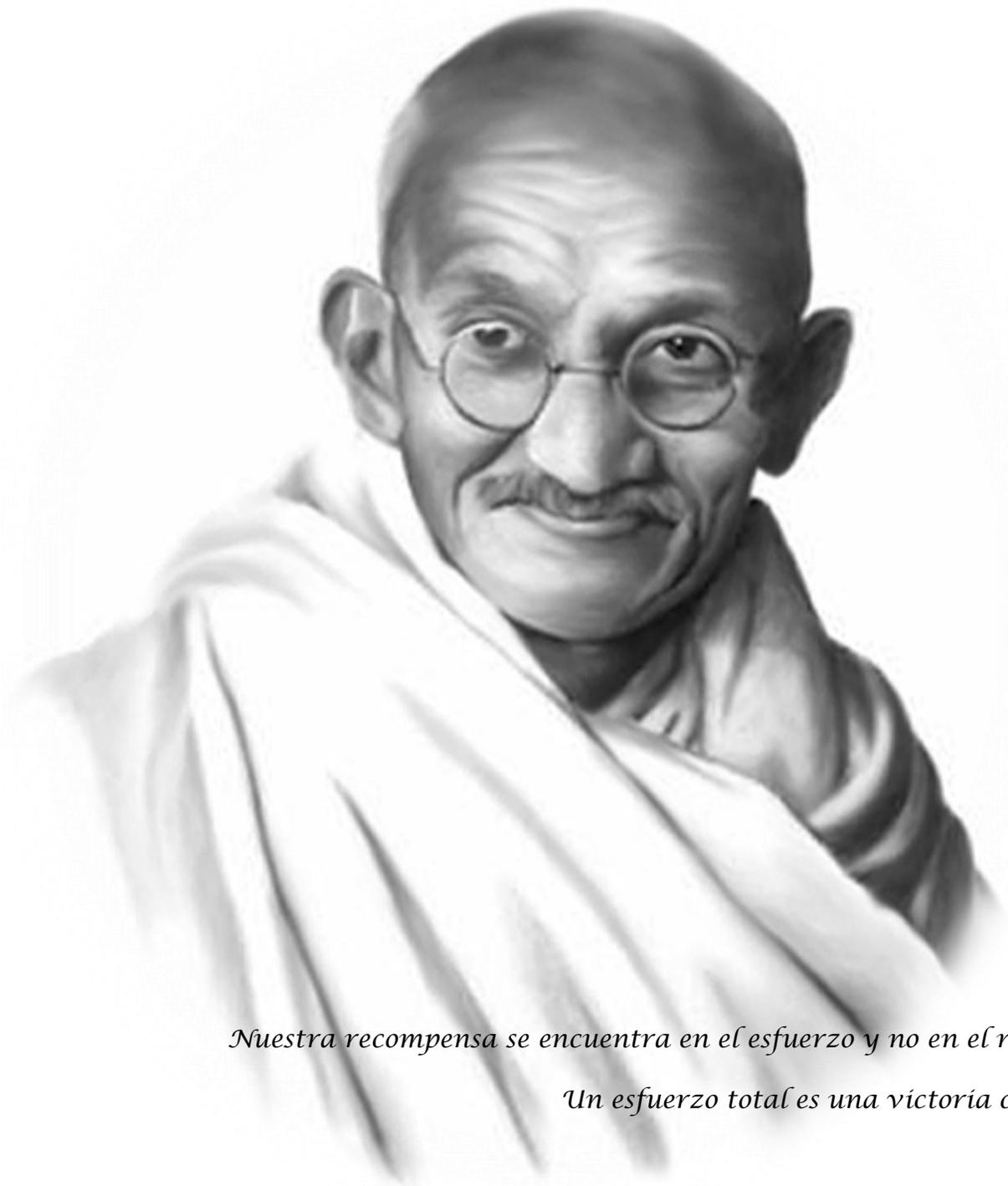
Tutores:

Ing. Mileydi Moreno Mirabal.

Ing. Ernesto de la Cruz Guevara Ramírez.

La Habana, Cuba. Junio 2012

“Año 54 de la Revolución”.



Nuestra recompensa se encuentra en el esfuerzo y no en el resultado.

Un esfuerzo total es una victoria completa.

Mahatma Gandhi

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

José Luis Simón Peña.

(Autor)

Ing. Ernesto de la Cruz Guevara Ramírez.

(Tutor)

Ing. Mileydi Moreno Mirabal

(Tutor)

DATOS DE CONTACTO

Tutor: Ing. Ernesto De La Cruz Guevara Ramírez.

Graduado como Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2008. Actualmente pertenece al Centro de Informática Industrial (CEDIN) de la Facultad 5 de la UCI.

Empresa: Universidad de Ciencias Informáticas UCI.

Dirección: carretera a San Antonio Km. 2 1/2 Torrens, Boyeros La Habana.

e-mail: elguevara@uci.cu

Tutor: Ing. Mileydi Moreno Mirabal

Graduada como Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2008. Actualmente pertenece al Centro de Informática Industrial (CEDIN) de la Facultad 5 de la UCI.

Empresa: Universidad de Ciencias Informáticas UCI.

Dirección: carretera a San Antonio Km. 2 1/2 Torrens, Boyeros La Habana.

e-mail: mmirabal@uci.cu

AGRADECIMIENTOS

A mis padres por haberme apoyado y guiado en cada paso de la vida.

A mis hermanas que siempre han estado a mi lado.

A mi familia que siempre ha confiado en mí.

A mis tutores, por su apoyo incondicional y su trabajo incansable por formarnos.

*A todos mis amigos tanto dentro y fuera de la universidad por la amistad que me
han brindado.*

A mis compañeros de grupo que siempre me apoyaron.

*A Fidel, Ernesto, Rohelís, José Andrés, Nayía, Sonia, Isidro y todos los que de una
forma u otra me brindaron su apoyo en el desarrollo del trabajo de diploma.*

*A mis profesores de toda la vida, a la facultad y a la revolución por haberme
dado la oportunidad de formar parte de este proyecto.*

DEDICATORIA

A mi mamá, mi papá y mis hermanas.

A mis abuelos y familia en general.

RESUMEN

La evaluación de la postura ha sido una preocupación constante por los médicos especialistas en rehabilitación y se ha visto beneficiada por la evolución de los instrumentos de medición para este fin. En este trabajo se trata de resolver la problemática que enfrentan los especialistas del Centro Nacional de Rehabilitación Hospital Julio Díaz de utilizar un método uniforme en la realización de las evaluaciones posturales. Para resolver el problema de investigación se hace un análisis de los principales métodos e instrumentos de evaluación postural utilizados internacionalmente para conocer las tendencias, resaltando una creciente actividad en la utilización de software. Luego de haber analizado los principales conceptos de Realidad Aumentada para delimitar el dominio de la solución, se propone una solución basada en dicha tecnología aprovechando las características que ofrece esta. En el trabajo se implementa un prototipo de software para apoyar a los especialistas con la realización de mediciones que utiliza técnicas de Realidad Aumentada. En la solución se introduce información virtual mediante análisis de imágenes y video tomados del contexto de una escena real, donde el proceso de la evaluación pudiera tener lugar. Se realiza un caso de estudio utilizando una maqueta con mediciones conocidas para comprobar los resultados del *software* implementado, obteniéndose resultados aceptables.

Palabras clave: evaluación postural, rehabilitación, biomecánica, mediciones, realidad aumentada, visión por computadoras.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	4
1.1. Conceptos asociados al dominio del problema	4
1.1.1. Postura	4
1.1.2. Visión por computadoras	5
1.1.3. Realidad Virtual (RV)	6
1.1.4. Realidad Aumentada (RA)	6
1.1.5. Tracking.....	7
1.2. Desarrollo de la evaluación postural	7
1.2.1. Sistemas de rehabilitación relacionados	8
1.3. Características de los sistemas de RA.....	10
1.4. Dispositivos de visualización.....	10
1.5. Framework de desarrollo	13
1.5.1. Designers' Augmented Reality Toolkit (DART)	13
1.5.2. Distributed Wearable Augmented Reality Framework (DWARF).....	13
1.5.3. Studierstube	14
1.5.4. Augmented Reality Toolkit (ARToolKit).....	14
1.6. Motor gráfico.....	15
1.6.1. OpenSceneGraph (OSG).....	15
1.6.2. OSGART	17
1.7. Metodologías de desarrollo de <i>software</i>	18
1.7.1. Las metodologías tradicionales o pesadas	18
1.7.2. Metodologías ágiles.....	18
1.8. Conclusiones parciales	21
CAPÍTULO 2. PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN	22
2.1. Soluciones técnicas	22

2.2.	Descripción de la solución	24
2.2.1.	Descripción del seguimiento	25
2.2.2.	Organización de la escena según OSG	26
2.2.3.	Integración del seguimiento y la escena	27
2.2.4.	Procedimientos matemáticos	29
2.3.	Modelo del dominio.....	31
2.3.1.	Glosario de términos del dominio.....	32
2.4.	Exploración.....	32
2.4.1.	Actores del sistema	32
2.4.2.	Lista de reserva del producto.....	33
2.4.3.	Historias de usuarios	33
2.4.4.	Diseño de caso de prueba	36
2.5.	Planificación	36
2.5.1.	Plan de entrega	36
2.5.2.	Estimaciones del esfuerzo	37
2.5.3.	Plan de iteraciones	37
2.5.4.	Plan de duración de iteraciones.....	38
2.6.	Conclusiones parciales	38
CAPÍTULO 3. CONSTRUCCIÓN DE LA SOLUCIÓN.....		39
3.1.	Diseño del sistema	39
3.1.1.	Tarjetas CRC.....	39
3.1.2.	Patrones de diseño.....	40
3.1.3.	Modelo de datos	40
3.2.	Estándar de codificación.....	41
3.3.	Desarrollo de Iteraciones	42
3.4.	Implementación	42
3.5.	Pruebas de aceptación	44
3.6.	Pruebas de validación.....	48

3.7. Conclusiones parciales del capítulo	50
CONCLUSIONES	51
RECOMENDACIONES	52
REFERENCIAS BIBLIOGRÁFICAS	53
ANEXOS	56
ANEXO 1. Tarjetas CRC	56
ANEXO 2. Tareas por historia de usuario iteración 1	58
ANEXO 3. Tareas por historia de usuario iteración 2	60
ANEXO 4. Pruebas de aceptación iteración 2	63
GLOSARIO DE TÉRMINOS	70

INTRODUCCIÓN

Nuestro país no es ajeno a los avances tecnológicos existentes en el mundo, por lo que ha destinado importantes recursos para el desarrollo de la investigación y producción de software. La inserción de nuestro país en un mercado internacional cada vez más competente y exigente, demanda la liberación de productos de software con elevada calidad. Como una estrategia para el desarrollo de software para la sociedad cubana surge la Universidad de las Ciencias Informática (UCI) que integra la actividad productiva con la de formación de los futuros profesionales.

En la UCI, la producción se concentra en el desarrollo de múltiples proyectos investigativos y productivos agrupados en diversos centros en toda la universidad. En estos se han obtenido reconocidos resultados en esferas como la salud, educación, software libre, automatización de procesos, entre otros. Actualmente el Centro de Informática Industrial (CEDIN) cuenta con un Departamento de Visualización y Realidad Virtual, existe una línea de visualización científica que investiga varios temas referentes a los gráficos por computadoras, entre ellos cómo desarrollar software con técnicas de Realidad Aumentada. El CEDIN, en conjunto con el Centro Nacional de Rehabilitación “Hospital Julio Díaz” (CNRHJ), desarrolla una investigación conjunta en la que se propone la Realidad Aumentada como un método de solución para asistir al personal médico especializado en la realización del proceso de evaluación postural de los pacientes.

La evaluación de la postura humana no debe ser invasiva en ninguna de sus variantes y no debe ofrecer al paciente ninguna disconformidad física bajo circunstancias normales. Habría que realizar este proceso en una habitación previamente preparada con una serie de requisitos para que el paciente se sienta cómodo. Las evaluaciones posturales frecuentes podrían detectar alteraciones y ayudar en la prevención de sus consecuencias duraderas.

El proceso de evaluación de la postura de los pacientes del hospital “Julio Díaz” se realiza fundamentalmente a través de la observación clínica. Este proceso está caracterizado por un importante componente subjetivo que introduce determinados márgenes de imprecisión, cualitativos y cuantitativos en las mediciones. Usualmente no se cuenta con los instrumentos adecuados para realizar las evaluaciones y no se logra un criterio estandarizado como lenguaje común entre los especialistas. Esta situación nos permite formular como **problema científico**: la necesidad de mejorar el proceso de evaluación postural de los pacientes del Centro Nacional de Rehabilitación Hospital Julio Días.

Por lo antes expuesto se plantea como **objeto de estudio**: *Software* de rehabilitación, y como **campo de acción**: Los *software* de rehabilitación para el análisis de la postura con Realidad Aumentada.

En aras de darle solución al problema científico antes expuesto se formuló como **objetivo**: elaborar un prototipo de aplicación informática, para el apoyar a los especialistas en rehabilitación en el proceso de análisis de la postura de los pacientes del Centro Nacional de Rehabilitación Hospital Julio Díaz, con la utilización de técnicas de Realidad Aumentada.

Para darle cumplimiento al objetivo trazado se determinó realizar las **tareas de investigación** que se muestran a continuación:

- Analizar bibliografía sobre el análisis de la postura mediante técnicas tradicionales y no tradicionales para conocer los trabajos relacionados.
- Analizar los procedimientos computacionales involucrados en el análisis de la postura para conocer su estructura y sus características básicas.
- Analizar el vínculo entre las técnicas de visión por computadoras, Realidad Virtual y Realidad Aumentada con las técnicas de análisis de la postura para la elaboración de la solución.
- Revisar analíticamente posibles bibliotecas de *software* que se puedan reutilizar para implementar los procedimientos de análisis de la postura.
- Elaborar la propuesta de solución a la problemática planteada, incluyendo las respectivas validaciones de los resultados obtenidos.

El proceso de investigación y elaboración de este trabajo se guió por la utilización de los métodos teóricos y empíricos que a continuación se mencionan:

Teóricos

Histórico – Lógico: este método permitió la consulta y evaluación de la bibliografía para determinar los conceptos fundamentales que se relacionan con los temas que se abordan en el trabajo, así como para conocer los trabajos relacionados con el mismo.

Analítico – Sintético: mediante este método se pudo hacer un análisis para obtener conocimientos generales, buscar la esencia de lo que existe en el mundo acerca de evaluación de la postura y los rasgos que lo caracterizan y lo distinguen.

Empíricos

Revisión de Documentos: La revisión de los trabajos relacionados con el tema de investigación sirvieron para profundizar en el campo de acción, y se logró la elaboración del documento de tesis, lo cual se refleja en la bibliografía consultada y referencias bibliográficas.

Estructura de la tesis

Capítulo 1. “Fundamentación teórica”: en este capítulo se enuncian las definiciones y los conceptos fundamentales del tema en cuestión. Se hace un estudio de los trabajos relacionado con el proceso de evaluación de la postura humana. A demás, se profundiza en el estudio de las herramientas y tecnologías utilizadas para el desarrollo de sistemas de realidad aumentada en el mundo, lo cual constituyó la base teórica necesaria para llevar a cabo la presente investigación.

Capítulo 2. “Propuesta y descripción de la solución”: en este capítulo se obtiene una visión más general de la solución. Se seleccionan las herramientas que formarán parte del ambiente de desarrollo, se define la estructura principal del prototipo funcional y se explica de forma detallada la solución. Además, se detallan los pasos de la metodología XP seleccionada para el desarrollo de la solución.

Capítulo 3. “Construcción de la solución”: en este capítulo se tratan los aspectos fundamentales del diseño, implementación y prueba de la solución para la elaboración del prototipo funcional.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se exponen conceptos relacionados con la postura humana y la realidad aumentada, así como las características y funcionamiento de esta última. Se exponen los trabajos relacionados con el proceso de evaluación de la postura humana. Además, se profundiza en el estudio de las herramientas y tecnologías utilizadas para el desarrollo de sistemas de realidad aumentada en el mundo, lo cual constituyó la base teórica necesaria para llevar a cabo la presente investigación.

1.1. Conceptos asociados al dominio del problema

Para lograr un mejor entendimiento de los temas que se abordarán en la presente investigación, se mostrarán una serie de conceptos identificados durante el proceso investigativo, así como la descripción de cada uno de estos facilitando así la comprensión de los temas abordados.

1.1.1. Postura

La postura deriva del latín “positura” que quiere decir modo en que está puesto, ya sea un objeto, un animal o un ser humano, también ha sido definida por autores del tema como la posición del cuerpo en cualquier modo o entorno y está basada en la posición de la columna vertebral y todas las articulaciones del sistema osteomuscular¹. (1)

Algunos ejemplos de postura son la posición de sentado, ver Figura 1 a), la posición de pie, ver Figura 1 b), y la posición de acostado, ver Figura 1 c). Mantener una mala postura trae como consecuencia que el organismo tienda a compensar el equilibrio generando contracciones musculares adicionales, que alteran la biomecánica² corporal, producen cambios degenerativos precoces y lesiones de sobrecarga, observándose con mayor frecuencia en la población deportivamente activa (2).

¹ Relacionado con músculos, los huesos, los tendones, los ligamentos, las articulaciones y los cartílagos.

² Ciencia que estudia la aplicación de las leyes de la mecánica a las estructuras y los órganos de los seres vivos.

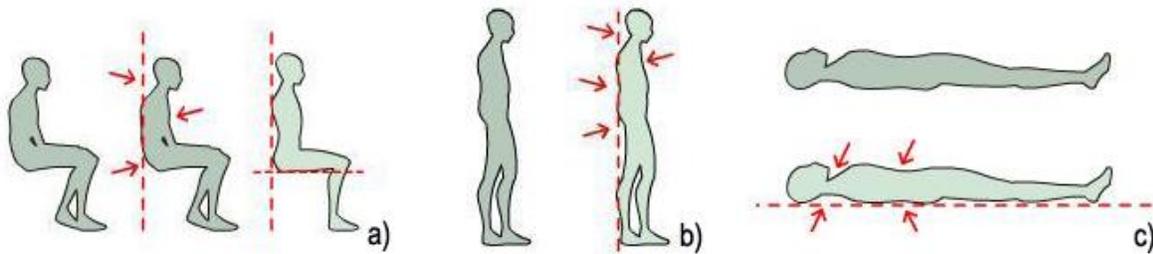


Figura 1. Posturas Humanas. a) posición de sentado, b) posición de pie, c) posición de acostado.

Los defectos posturales son fallas en la integración de los mecanismos que regulan la postura, estos son adquiridos a partir de malos hábitos posturales o defectos estructurales que son reflejados más tarde los daños a otras estructuras. En la mayoría de los casos en la edad adulta los problemas se intensifican si no son tratados en edades tempranas (3).

1.1.2. Visión por computadoras

La visión artificial, visión por computador o visión técnica, se puede describir según (4) como un sub-campo de la inteligencia artificial. Es el proceso de extracción de información del mundo real a partir de imágenes utilizando como herramienta una computadora. Desde un punto de vista técnico, un sistema de visión por computador es un sistema autónomo que realiza alguna de las tareas que el sistema de visión humano es capaz de realizar. El propósito de la visión artificial es programar un computador para que "entienda" una escena o las características de una imagen.

Tener la capacidad de simular el efecto de la visión humana a través de un sistema de percepción electrónico se torna un tanto difícil. Esto se debe a que el ojo humano tiene la capacidad de percibir la luz reflejada por los objetos en un escenario de tres dimensiones (3D), mientras que los sistemas electrónicos analizan estos objetos a partir de proyecciones en imágenes de dos dimensiones (2D). La reducción de una dimensión produce la pérdida de gran cantidad de información, lo que eleva notablemente la dificultad del proceso de visión humano.

En los últimos años se han logrado grandes avances en esta área, principalmente en el desarrollo de aplicaciones especializadas en la detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes (por ejemplo, caras humanas), la evaluación de los resultados, el registro de diferentes imágenes de una misma escena u objeto, hacer concordar un mismo objeto en diversas

imágenes, el seguimiento de un objeto en una secuencia de imágenes, mapeo de una escena para generar un modelo tridimensional de la escena entre otros.

1.1.3. Realidad Virtual (RV)

En la informática lo virtual es visto como “algo simulado”, creado por un ordenador para realizar un determinado fin. Teniendo en cuenta lo expresado por (5) la RV es un sistema interactivo que permite sintetizar un mundo tridimensional ficticio, crea así una ilusión de la realidad en la que es sumergido el usuario. Se puede decir que es una técnica fotográfica de 360⁰ que permite al usuario atrapado en este mundo moverse en todas direcciones. La RV puede ser considerada en varios aspectos como la interfaz definida entre los seres humanos y el ordenador.

Básicamente consiste en tratar de simular todo lo que una persona es capaz de percibir en el mundo real, como son los gráficos, el sonido e incluso sensaciones de aceleración o movimiento. Todas estas sensaciones diferentes son combinadas y presentadas al usuario de forma tal que se sienta inmerso en el universo ficticio generado por el ordenador, hasta el punto de dejar de percibir la realidad y ser engañado al sentirse que se encuentra en otro universo.

1.1.4. Realidad Aumentada (RA)

El concepto de RA es dado a conocer por Pierre Wellner en 1993. Donde define un concepto opuesto al conocido por RV. La RA se caracteriza por traer al mundo real componentes digitales que añaden información virtual a la información física ya existente. A continuación se da a conocer la definición dada por Ronald Azuma y mayoritariamente aceptada:

“La RA es una variación de Ambientes Virtuales (AV), o RV como es comúnmente llamada. Las tecnologías de RV sumergen al usuario dentro de un ambiente sintético. Mientras está inmerso, el usuario no podrá ver el mundo real a su alrededor. En contraste, la RA permite al usuario ver el mundo real, con objetos virtuales superpuestos o compuestos con el mundo real. La RA suplementa la realidad, antes que reemplazarla completamente” (6). La RA logra la inserción de elementos virtuales en el mundo real valiéndose de las técnicas de *tracking* existente.

1.1.5. Tracking

El *tracking* es el seguimiento o rastreo que consigue disponer permanentemente de un sistema de referencia virtual alineado con el sistema de referencia del objeto real, y que por lo tanto realiza la misma transformación sólido rígida que este último (traslación y rotación) (7).

El *tracking* 3D basado en visión por computadoras puede ser descompuesto en dos pasos principales; el primero es el procesamiento de imagen, para extraer alguna información de las imágenes, y el otro la estimación de la posición en sí mismo. Una de las formas de hacer *tracking* más exitosas ha sido la adición de marcadores en la escena, lo cual tributa considerablemente a los dos pasos descritos con anterioridad. Esta afirmación se sustenta porque los marcadores se convierten en características fáciles de extraer de las imágenes; además proveen otros elementos como las medidas, que son factibles y confiables para la estimación de la posición.

Los marcadores se definen como: un cuadrado externo negro con un cuadrado interno blanco que a su vez contiene una figura que debe ser lo más asimétrica posible como se muestra en la Figura 2.



Figura 2. Marcador cuadrado.

1.2. Desarrollo de la evaluación postural

La evaluación de la postura humana es una actividad que ha sido efectuada por profesionales de diferentes especialidades, quiroprácticos, masajistas, fisioterapeutas, entre otros; en áreas como la medicina física y rehabilitación. Por tal motivo muchos profesionales han tratado de estandarizar o hacer más ameno su trabajo para poder realizar diagnósticos precisos y emitir una serie de tratamientos que conduzcan a la mejoría de sus pacientes. Los trabajos realizados en el afán de cuantificar los resultados de la evaluación postural van desde las listas de chequeo a través de datos experimentales recogidos por

medios convencionales como plomadas (instrumento de construcción) hasta la utilización de programas de computadoras (8).

La utilización de instrumentos mecánicos como el indicador de la palpación, ha brindado soluciones muy originales que combinan la facilidad y ventajas propioceptivas³ de la palpación con la objetividad y la fiabilidad de las mediciones de espesor. El procedimiento de este indicador está basado en la palpación del cuerpo con los dedos de la mano. El instrumento mide la distancia de los dedos que palpan en centímetros y la inclinación de los dedos en grados. Una regla de cálculo única le da la diferencia de altura entre los dos puntos de referencia que palpan tanto en centímetros como en pulgadas.

Al igual que el instrumento anterior, se puede encontrar un amplio conjunto de instrumentos que se utilizan para hacer evaluación postural con sus distintas funcionalidades, los cuales son: conjuntos de evaluación de la postura (*Posture Evaluation Kit*) (9) que constituye una rejilla graduada suspendida del techo que sirve para realizar mediciones sobre el cuerpo de los pacientes; escoliómetros, antropómetros largos y pequeños, calibrador de pecho, indicadores de circunferencia de dedos y cintas antropométricas de Gulick. Todos estos instrumentos en su conjunto son utilizados para hacer estimaciones precisas de la postura de los pacientes y plantear tratamientos de rehabilitación para los pacientes.

1.2.1. Sistemas de rehabilitación relacionados

Además de los instrumentos mecánicos, el *software* de computadoras es una alternativa utilizada en el mundo entero con el propósito de apoyar dichas mediciones. Con el desarrollo de las tecnologías de las computadoras y la inserción de estas en el campo de la salud, la visión por computadoras ha desempeñado un papel fundamental en el desarrollo de aplicaciones informáticas que son utilizadas por diferentes especialistas para realizar el diagnóstico a sus pacientes.

La compañía Biotonix (8) es una de las empresas líderes en el mercado de *software* para la rehabilitación y la evaluación de la postura, con una amplia gama de productos certificados por la Asociación Internacional de Quiroprácticos (ICA por sus siglas en inglés). Entre los productos se encuentra el BioPrint™ fundamentado en conceptos firmes de las ciencias biomecánicas y la física, caracterizado por procedimientos de corrección o rehabilitación de la columna vertebral y la pelvis. Otro producto es el FitPrint™ que se basa en el cálculo y corrección de la simetría y la apariencia de la postura; de la misma

³ Informan sobre la situación del cuerpo en el espacio, la postura del cuerpo, los movimientos, el equilibrio, la actitud entre otros.

FUNDAMENTACIÓN TEÓRICA

forma la compañía Bionix cuentan con productos de *software* para tareas similares pero con propósitos más detallados.

Existen otras compañías de *software* que distribuyen productos para realizar acciones similares en cuanto a la evaluación de la postura; por ejemplo *PostureCo* (10) ofrece productos basados en el mercado de imágenes de rayos X para hacer sus estudios con gran precisión con el *software PostureRay* y evaluaciones de la postura a través de imágenes fotográficas con el *PostureScreen*. Según Rossi, Ph.D. Mark en "*Postural evaluation*" (1) muchos especialistas prefieren realizar el proceso de evaluación postural utilizando fotografías de la postura, mientras que otros prefieren utilizar imágenes de Rayos X, sobre la base de que estas modalidades conducen a resultados de mayor precisión que un simple examen de observación visual, no obstante es una técnica invasiva.

La compañía BTS⁴ (11) se dedica a la producción de sistemas informáticos para asistir al personal médico que labora en el área de rehabilitación y medicina física, entre sus trabajos se debe destacar el *software* BTS Digivec que permite la evaluación de la dinámica postural y del comportamiento estabilométrico⁵ del paciente. El cual cuenta con un sistema de realidad aumentada para el análisis en tiempo real del movimiento dinámico y de la postura.

Los trabajos vistos hasta el momento derivados de la bibliografía consultada dan una visión del estado actual del uso de la informática en la salud, en especial en la medicina física y la rehabilitación; además de ofrecer una vista de la tendencia a aprovechar las facilidades de la automatización de actividades de diagnóstico y tratamiento de enfermedades, con altos niveles de fiabilidad a través de *software* de computadoras. Aprovechando esta cobertura, con este trabajo se pretende contribuir con el proceso de introducir la utilización de técnicas de RA para realizar procesos de medición en la evaluación postural. Estos dispositivos pueden ir desde una computadora personal hasta dispositivos móviles o asistentes personales dotados con cámaras.

Una de las ventajas más prominentes que ofrece la RA para *software* de evaluación postural, es que se puede complementar la realidad con información virtual; que los especialistas estimen de interés práctico y para enriquecer los diagnósticos que se realizarán sobre la base de los resultados obtenidos, después de un proceso de mediciones y estimación de la postura de los pacientes. Las técnicas de RA, basadas en

⁴ *Bioengineering - Integrated Solution.*

⁵ La postura de los pacientes y su evolución en el tiempo.

métodos de visión por computadoras, en la comunidad científica internacional demuestran ser una solución viable para su utilización en procesos de evaluación de la postura (12).

1.3. Características de los sistemas de RA

Los sistemas de RA generalmente inician con el registro de señales provenientes del mundo real. Estas señales son procesadas dejando las imágenes preparadas para el proceso de segmentación o extracción de objetos y el reconocimiento de patrones. Este proceso nos permite conocer donde hay que remplazar un objeto real por uno virtual, cual objeto virtual es el que va a remplazar y en qué posición y perspectiva. Todo este proceso requiere de gráficos en tiempo real y permitir la interactividad, por tanto es necesario garantizar con anticipación la correspondencia entre patrones, rasgos geométricos del entorno, la posición tridimensional y la perspectiva de dibujado para agilizar el proceso.

Después de conocerse las propiedades del dibujado se establecen y aplican una serie de transformaciones al objeto virtual (parámetros de la cámara virtual, iluminación, entre otros) y se pasa al proceso de composición de la señal de video de salida. Esta nueva señal de video donde se mezclan ambos mundos es enviada al monitor.

La ejecución de todo este proceso resulta en un sistema RA con las siguientes características (6):

- Combina los elementos reales y virtuales.
- Interactivo en tiempo real.
- Registrado en Tres-Dimensiones (Información almacenada en 3D).

Para el desarrollo de aplicaciones de RA, es necesario contar con dispositivos capaces de garantizar la combinación del mundo real y virtual así como los demás retos que nos plantea el desarrollo de estos sistemas.

1.4. Dispositivos de visualización

En el artículo "*A Survey of Augmented Reality*" escrito por R. Azuma en 1997 se hace alusión a la existencia de dos tecnologías, la basada en óptica y la basada en video. Cada una de estas tecnologías a la hora de su implementación cuenta con sus respectivos dispositivos, los cuales se deben tener en cuenta a la hora de seleccionar entre ambas tecnologías a utilizar, según el área de aplicación que tenga la solución a desarrollar.

Head Mounted Display (HMD) Óptico transparente

Este dispositivo se basa en la colocación de mezcladores ópticos parcialmente transparentes frente a los ojos del usuario, de tal forma que el usuario es capaz de ver el mundo real a través de ellos. A su vez estos mezcladores permiten reflejar imágenes virtuales provenientes de una especie de proyector (6). En la Figura 3 se muestra un diagrama conceptual de un HMD Óptico transparente.

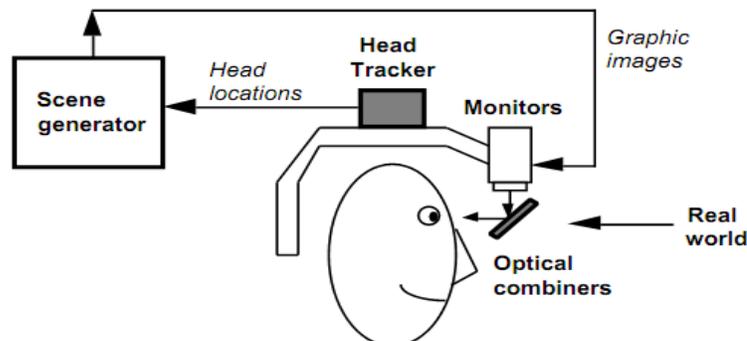


Figura 3. Diagrama conceptual de un HMD óptico transparente.

Por lo tanto los mezcladores de los HMD Ópticos funcionan como un par de gafas solares, que solo deja entrar una parte de la luz del mundo real; cuando se le suministra energía proveniente de los proyectores del HMD además serán capaces de reflejar una parte de esa luz.

Head Mounted Display (HMD) Con visión a través de video

A diferencia de la tecnología óptica, el HMD con visión a través de video es un HMD con vista cerrada combinado con una o dos cámaras montadas sobre la cabeza del usuario, por tanto el usuario sólo será capaz de visualizar el mundo de forma indirecta a través de las cámaras de video. Antes de mostrar las imágenes capturadas por las cámaras al usuario, estas son combinadas o mezcladas con las imágenes gráficas provenientes del generador de escena. Luego esta mezcla es enviada a los monitores que se encuentran frente a los ojos del usuario. En la Figura 4 se muestra un diagrama conceptual de un HMD con visión a través de video (6).

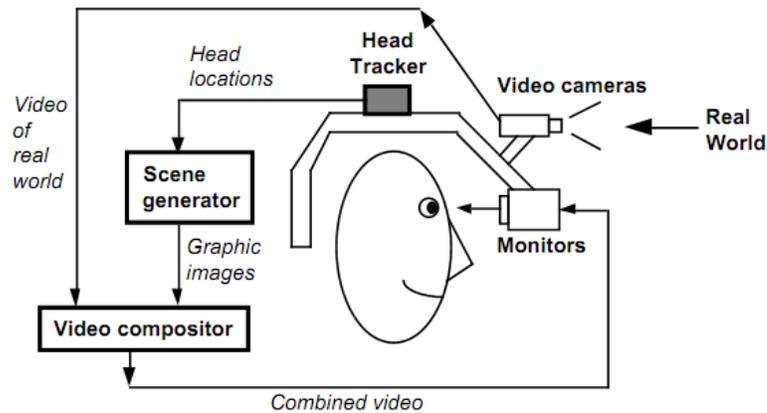


Figura 4. Diagrama conceptual de un HMD con visión a través de video.

Dispositivos alternativos basados en monitores

Para el desarrollo de sistemas de RA también se puede utilizar los dispositivos alternativos basados en monitores, los cuales surgieron como una alternativa de los HMD con visión a través de video. A diferencia de los HMD con visión a través de video, la mezcla de las imágenes proveniente de las cámaras de video del mundo real y las imágenes virtuales provenientes del generador de escena, se mostrarán al usuario en el monitor del computador (6). En la Figura 5 esto se aprecia gráficamente.

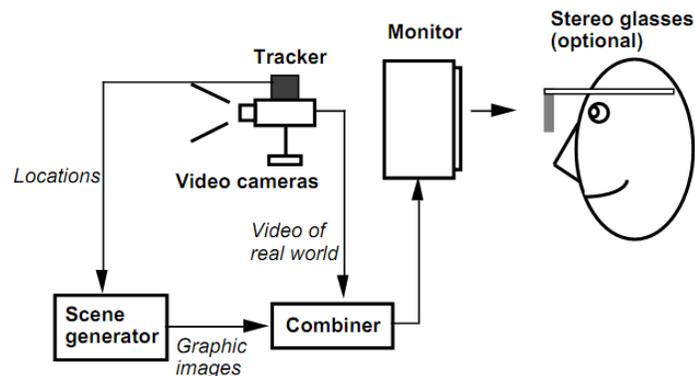


Figura 5. Diagrama conceptual de un dispositivo basado en monitor.

La mayoría de los prototipos de aplicaciones médicas mundialmente utilizan enfoques de video, por la flexibilidad en la mezcla de ambos mundos (el real y el virtual) y por las estrategias de registro adicionales que estos ofrecen (6).

Después del análisis de los diferentes dispositivos que son empleados en los sistemas de RA se puede decir que ambas tecnologías (la óptica y la de vídeo) juegan sus papeles en función del contexto en que son aplicadas. Esto justifica que la elección de una u otra tecnología dependen del área y los requisitos de la aplicación. Por tanto la tecnología de video y el dispositivo de visualización alternativo basado en monitor se ajustan más a las necesidades de la investigación, ya que una razón determinante por la cual no se seleccionó ninguno de los dos primeros dispositivos es el precio al que se venden en el mercado internacional actualmente.

1.5. Framework de desarrollo

Para elaborar una aplicación de RA existen dos variantes, una es realizarlas totalmente partiendo del inicio, mediante la implementación de cada una de las herramientas que se necesitan para desarrollar la aplicación. La otra variante es utilizar herramientas existentes que agilizan el proceso de desarrollo. En esta investigación se siguió la segunda variante. A continuación se analizan algunas de las herramientas más importantes para el desarrollo de aplicaciones de Realidad Aumentada.

1.5.1. Designers' Augmented Reality Toolkit (DART)

La Herramienta de Realidad Aumentada para Diseñadores (DART por sus siglas en inglés) está diseñada para dar soporte a la construcción rápida de prototipos de aplicaciones de RA. DART se construyó en la parte superior de Macromedia (ahora Adobe) Director, fue desarrollado en su mayoría en un lenguaje interpretado de *scripts*, y puede ser modificado por los diseñadores para satisfacer sus necesidades. El seguimiento de los marcadores en tiempo real lo logra al hacer uso de la biblioteca ARToolKit.

DART está liberado bajo una licencia libre que solo impone una restricción, que no puede ser vendida en sí misma o con pequeños cambios; puede ser útil para la implementación de aplicaciones multipropósito aunque se sustenta de herramientas privativas.

El hecho del modelo de licencia empleado por las herramientas sobre las que está sustentada y por estar discontinuado el proyecto se decide no utilizar esta herramienta.

1.5.2. Distributed Wearable Augmented Reality Framework (DWARF)

Desde el punto de vista de la arquitectura de *software*, para los creadores de la Plataforma de Realidad Aumentada Distribuida para Dispositivos Portátiles (DWARF por sus siglas en inglés), los sistemas de RA pueden ser divididos en un conjunto de componentes, de forma tal que cada uno contribuye con el

sistema completo con una funcionalidad dedicada. Los autores de esta herramienta se centraron fundamentalmente en encontrar una solución donde fuera posible la construcción rápida de prototipos de aplicaciones con varios investigadores en diferentes computadoras.

La arquitectura de DWARF está basada en el concepto de colaboración de servicios distribuidos. Para llevar a cabo esta arquitectura, los autores de DWARF utilizan el estándar CORBA (*Common Object Request Broker Architecture*), que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación a métodos remotos bajo un paradigma orientado a objetos.

A pesar de que este *framework* ofrece muchas ventajas, mencionadas anteriormente, no se ajusta a los propósitos de la investigación, porque originalmente está construido para sistemas distribuidos y para el uso de dispositivos móviles.

1.5.3. Studierstube

El sistema *Studierstube* fue uno de los primeros sistemas de RA colaborativa. Utiliza la RA colaborativa para embeber imágenes generadas por computadora en el entorno de trabajo real. Una de sus funcionalidades es que múltiples usuarios se congregan en una habitación y pueden experimentar un espacio virtual compartido poblado de datos tridimensionales.

La licencia de *Studierstube* es privativa y utiliza como dispositivos de visualización los *HMDs* ópticos con transparencia o pantallas de proyección (usando proyectores) para combinar los gráficos con la vista del mundo real del usuario. El reconocimiento de marcadores se hace a través de *stbTracker*⁶, una evolución de la biblioteca *ARToolKitPlus*, a su vez, una extensión de *ARToolKit*.

La razón fundamental por la que se decide no utilizar *Studierstube* en el desarrollo de la investigación es por el modelo de licencia. A demás por la utilización del *HMDs* óptico como dispositivo de visualización el cual es muy caro en el mercado internacional.

1.5.4. Augmented Reality Toolkit (ARToolKit)

Herramienta de Realidad Aumentada (*ARToolKit* por sus siglas en inglés) es una biblioteca de *software* para el desarrollo de aplicaciones de RA. La cual hace uso de los algoritmos de visión por computadoras para resolver el problema del *tracking* de los marcadores. Las funcionalidades de *ARToolKit* permiten

⁶ Es una biblioteca de la visión por computadora para la detección y representan la estimación de los marcadores de referencia en 2D.

calcular la posición real de los marcadores físicos y su orientación relativa a la cámara en tiempo real para hacer *tracking*, permiten el fácil desarrollo de una amplia gama de sistemas de RA. Entre las características de ARToolKit se incluyen (13):

- *Tracking* de orientación/posición de una sola cámara.
- Posee la habilidad de utilizar cualquier marcador o patrón. (Siempre que sea cuadrado).
- Código de calibración de cámara muy fácil.
- Suficientemente rápida para ser usada en aplicaciones de RA.
- Superpone objetos virtuales 3D en marcadores reales (basado en algoritmos de visión por computadoras).
- Es multiplataforma, soportada en distribuciones SGI IRIX, Linux, MacOS y Windows OS.
- Estrategia de patrones de marcadores extensible (el número de marcadores es directamente proporcional a la eficiencia).
- Interfaz de programación de aplicaciones (API) simple y modular, escrita en lenguaje C.
- Utiliza GLUT para el aspecto de la manipulación de ventanas y eventos, también para la biblioteca de video dependiente de hardware y el API estándar en cada plataforma.
- Utiliza OpenGL (*Open Graphics Library*) para las tareas de dibujo (*rendering*).
- Biblioteca liberada bajo licencia GPL distribuida con el código fuente completo para uso no comercial.
- Tiene una comunidad de desarrollo muy activa.

La biblioteca ARToolKit presenta un conjunto de características que la hacen un fuerte adversario frente a las demás analizadas anteriormente. A demás cabe destacar que es utilizada por las herramientas anteriores para realizar los trabajos de seguimiento.

1.6. Motor gráfico

El motor gráfico es el componente principal de un video juego u otra aplicación interactiva con gráficos en tiempo real; es la parte de un programa que controla, gestiona y actualiza los gráficos 3D.

1.6.1. OpenSceneGraph (OSG)

OSG (14) es un conjunto de herramientas gráficas multiplataforma de código abierto para el desarrollo de aplicaciones gráficas de alto rendimiento, tales como simuladores de vuelo, juegos, realidad virtual y

FUNDAMENTACIÓN TEÓRICA

visualización científica. OSG se basa en el concepto de Grafo de Escena, proporcionando un marco orientado a objetos como una abstracción de *OpenGL*. Libera al desarrollador de la implementación y optimización de gráficos a bajo nivel y proporciona muchas utilidades adicionales para el desarrollo rápido de aplicaciones gráficas. Se enfoca primordialmente en lo que el programador quiere hacer y no en como lo quiere lograr, para ellos el programador solo debe especificar la organización del grafo de escena que representa la escena que necesita representar gráficamente.

Características:

OSG es escrito totalmente en C++ estándar y OpenGL, hace pleno uso de la Biblioteca de Plantillas Estándar (por sus siglas en inglés STL), su implementación está basada en patrones de diseño de software, utiliza el modelo de desarrollo de código abierto y se centra en las necesidades de los usuarios finales. Los puntos fuertes de OpenSceneGraph son:

Productividad: Posee un conjunto de bibliotecas adicionales que hacen posible desarrollar aplicaciones gráficas de alto rendimiento muy rápidamente. Combina las experiencias de otros motores gráficos con modernos métodos de ingeniería de *software* para lograr el diseño de una biblioteca robusta y extensible.

Extensiones Soportadas: para la lectura y escritura de archivos la biblioteca proporciona una amplia variedad de formatos por medio de un mecanismo extensible de *plugins* dinámicos a través del módulo **osgDB**. Actualmente incluye más de cincuenta complementos para cargar formatos 3D y formatos de imágenes.

Portabilidad: el núcleo de OSG está diseñado para tener una dependencia mínima de cualquier plataforma. Esto permitió que OSG fuese portado rápidamente a un amplio rango de plataformas. Originalmente fue desarrollado en IRIX, luego fue portado a Linux, Windows, *FreeBSD*, Mac OSX, Solaris, HP-UX, AIX y hasta *PlayStation2*.

Soporte multi-lenguaje: OSG está disponible además en los lenguajes Java, Lua y Python. Además tiene una capa de introspección, que ofrece las características necesarias para construir los recubrimientos del lenguaje con el que se necesite trabajar.

1.6.2. OSGART

OSGART (15) es una biblioteca desarrollada en C++, multiplataforma que le da la posibilidad de crear fácilmente aplicaciones de RA mediante la combinación de la biblioteca ARToolKit de seguimiento con la biblioteca de dibujo OSG.

Funciones fundamentales OSGART:

- Alto nivel de integración de entrada de video (video objeto, *shaders*).
- Soporte de múltiples marcadores basados en características físicas o tecnologías de seguimiento (marcador basado en múltiples rastreadores)
- Registro fotométrico, basado en técnicas para la oclusión y el dibujo de sombras.

Características:

- Diseño Orientado a Objetos.
- Componentes de video se y de *tracking* a través de complementos.
- Soporte de múltiples entradas de vídeo (en directo, un archivo de red).
- Integración a alto nivel de objetos de vídeo (vídeo de fondo)
- Concepto de marcador genérico, (incluye los de ARToolKit).
- Soporte de múltiples tecnologías de seguimiento basadas en marcador o reconocimiento de características naturales.
- Fácil desarrollo de aplicaciones interactivas.

La biblioteca OSGART ofrece a los desarrolladores de sistemas de RA conjunto de ventajas y características que se deben tener en cuenta a la hora de diseñar un sistema de este tipo. La posibilidad de integrar la biblioteca de seguimiento ARToolKit y el motor de gráficos OSG la convierten en una potente biblioteca para el desarrollo de sistemas de RA.

1.7. Metodologías de desarrollo de *software*

En 1996 PIATTINI define a las metodologías de desarrollo como “un conjunto de procedimientos, técnicas, herramienta, y un soporte, documental que ayuda a los desarrolladores a realizar un nuevo *software*” (16). Las mismas persiguen tres necesidades principales:

- Mejores aplicaciones, que conducen a una mejor calidad.
- Un proceso de desarrollo controlado.
- Un proceso normalizado en una organización, no dependiente del personal.

Las metodologías guían a los desarrolladores al crear un nuevo producto de *software*, pero los requisitos de un *software* a otro son tan variados y cambiantes, que ha promovido la variedad de estas metodologías. Las mismas se podrían clasificar en dos grandes grupos las tradicionales y las ágiles.

1.7.1. Las metodologías tradicionales o pesadas

Las metodologías se centran en el trabajo sobre el proceso de desarrollo del *software*, con el fin de conseguir un *software* más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto *software*. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas, notaciones para la modelación y documentación detallada ya que pretende prever todo de antemano. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Entre las metodologías pesadas existentes se encuentran:

- RUP (Rational Unified Procces).
- MSF (*Microsoft Solution Framework*).
- *Win-Win Spiral Model*.

1.7.2. Metodologías ágiles

Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que contribuyen a la entrega del proyecto menos complicada y más satisfactoria tanto para los clientes como

FUNDAMENTACIÓN TEÓRICA

para los equipos de entrega. Seguidamente se hará énfasis a las ideas principales de toda metodología ágil:

- Se encarga de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados.
- Es mucho más importante crear un producto *software* que funcione que escribir mucha documentación.
- El cliente está en todo momento colaborando en el proyecto.
- Es más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan.

Entre las metodologías ágiles existentes se pueden encontrar:

- XP (Extreme Programming).
- Scrum.
- Crystal Methodologies.

A continuación se muestra la Tabla 1 **Error! No se encuentra el origen de la referencia.** comparativa entre metodologías ágiles y tradicionales de desarrollo de *software* (17):

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Menos énfasis en la arquitectura del <i>software</i>	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.

FUNDAMENTACIÓN TEÓRICA

Grupos pequeños y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
	La arquitectura del <i>software</i> es esencial y se expresa mediante modelos.

Tabla 1. Metodologías ágil vs metodologías tradicionales.

A continuación se muestra Tabla 2 comparativa entre la metodología ágil XP (18) y la tradicional RUP (19):

XP	RUP
Características	
Pruebas unitarias continuas.	Programación por equipos.
Interacción con el usuario final.	Interacción con el usuario estratégico.
Simplicidad del código.	Está integrado en todo el ciclo de vida.
Selección de la Metodología	
Los requisitos cambian durante el proceso de desarrollo.	Comunicación entre el equipo.
Grupos pequeños de programadores.	Complejidad de desarrollo de acuerdo con el tamaño del proyecto.
Recomendable emplearlo en proyectos a corto plazo.	
Ventajas	
Satisfacción del programador.	Mayor documentación.
Simplicidad (Implica resolver en cada momento las necesidades actuales).	Es modelado y guiado por casos de uso.
Retroalimentación.	Es centrado en la arquitectura.
Programación organizada.	Iterativo e incremental.

Tabla 2. Metodología XP vs RUP.

Luego de ver realizado el análisis de las principales características que presentan tanto las metodologías ágiles como las tradicionales, se avizora que la metodología ágil XP se ajusta más a las necesidades de desarrollo de la solución.

1.8. Conclusiones parciales

En este capítulo se analizaron los conceptos relacionados con la evaluación de la postura humana y la realidad aumentada. Se hizo un estudio de los trabajos relacionados con el tema de la investigación, con el objetivo de ver los aspectos más importantes acerca del tema en cuestión. Se analizaron los dispositivos de visualización y las características más importantes de los principales *framework* de desarrollo de aplicaciones de RA. Con el desarrollo de este capítulo se trataron los aspectos teóricos necesarios para llevar a cabo la confección de la solución que propone la investigación.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

CAPÍTULO 2. PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

En el presente capítulo se realiza la propuesta de solución. Se seleccionan las herramientas que formarán parte del ambiente de desarrollo, se define la estructura principal del prototipo funcional y se explica de forma detallada la solución. Además se describirán los pasos de la metodología XP seleccionada para el desarrollo de la solución; esto incluirá la elaboración del modelo del dominio, la fase de exploración y la fase de planificación.

Kinetic es el nombre se le decide establecer a la solución, proveniente del término Kinesiología. La Kinesiología (del griego kenosis, movimiento, estudio) refiere a las técnicas terapéuticas que ayudan a que una persona recupere el movimiento normal de ciertas partes de su cuerpo. La kinesiología no es lo más adecuado para diagnosticar enfermedades, pero si puede ser útil para detectar, y así poder tratar adecuadamente, problemas musculares o esqueléticos.

2.1. Soluciones técnicas

La fundamentación teórica expuesta en el capítulo 1, sirvió de sustento para seleccionar el conjunto de herramientas y bibliotecas más adecuado para el desarrollo de Kinetic. Las mismas se describen a continuación:

Biblioteca de desarrollo: se seleccionó OSGART como biblioteca de desarrollo, ya que la misma ofrece una serie de facilidades y ventajas para la elaboración de sistemas de RA, expuestas en el epígrafe (1.6.2). Otra de las razones por la cual se seleccionó fue debido a que combina la biblioteca de seguimiento ARToolKit con la biblioteca de dibujado OSG, de las cuales se expusieron sus características en el epígrafe (1.5.4) y (1.6.1) respectivamente.

Dispositivo de Visualización: en el caso de esta investigación se seleccionó como dispositivo de visualización, el dispositivo alternativo basado en monitor como variante de la tecnología de video. Esta selección se justifica teniendo en cuenta que el sistema de salud pública cubano por lo general no puede costear los productos caros existentes en el mercado internacional. Según las necesidades de nuestro sistema nacional de salud, esta variante es de las más atractivas en términos de costo-beneficio. No obstante según (12) esta variante es la recomendada para el desarrollo de un sistema de RA capaz de asistir al personal médico.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

Lenguaje de programación: se seleccionó el lenguaje de programación C++. Este es un lenguaje de programación, diseñado a mediados de los años 1980, por *Bjarne Stroustrup*, como extensión del lenguaje de programación C. Se puede decir que C/C++ es un lenguaje que abarca tres paradigmas de la programación:

- La programación estructurada.
- La programación genérica.
- La programación orientada a objetos.

Existen variadas razones que hacen que lenguaje C++ tenga gran aceptación entre los programadores dedicados a los gráficos por computadoras. El lenguaje C++ es muy potente y rápido, sólo superado por el ensamblador. Es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear *software* de sistemas operativos, aunque también se utiliza para crear otro tipo de aplicaciones; puesto que es posible utilizar sus características de bajo nivel para realizar implementaciones óptimas, es catalogado como un lenguaje muy eficiente. Esta característica hace que sea ideal para desarrollar programas de gráficos por computadoras que requieren un alto grado de rapidez y optimización.

Herramienta de desarrollo: se selecciona Microsoft Visual Studio.NET 2008 como entorno integrado de desarrollo (IDE), ya que el mismo cuenta con herramientas auxiliares de completamiento de códigos como el Visual Assist lo que ayuda mucho a la codificación de la aplicación. Aunque este IDE despunta sobre los existentes para el desarrollo de aplicaciones por los índices de productividad que se pueden alcanzar, su utilización no implicó atar la solución a una plataforma específica.

QT Framework: se selecciona QT Framework (20) por ser un *framework* de desarrollo multiplataforma con una biblioteca de más de 400 clases, las cuales encapsulan toda la infraestructura necesaria para desarrollar aplicaciones robustas. Presenta utilidades para la construcción rápida de aplicaciones completas incluyendo la interfaz gráfica de usuario, con apariencia y aspectos nativos de las plataformas soportadas. Está separado en 13 módulos y la API que brinda incluye funciones para conectividad a bases de datos, programación en red, integración con gráficos 3D, desarrollo multi-hilos y lectura/escritura de archivos XML.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

Gestor de base de dato: se decidió emplear el gestor de bases de datos PostgreSQL (21). El PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente; utiliza un modelo cliente/servidor y usa multiprocesos como alternativa a la programación multi-hilos para garantizar la estabilidad del sistema.

Sus características técnicas lo hacen uno de los sistemas de gestión de bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 15 años, y durante este tiempo de desarrollo; la estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta. PostgreSQL funciona muy bien con grandes cantidades de datos y con una alta concurrencia de usuarios accediendo al sistema.

Metodología de ingeniería de *software*: se selecciona como metodología de ingeniería de *software* la metodología de desarrollo de *software* ágil XP, aprovechando las características expuestas en el capítulo anterior en el epígrafe (1.7.2).

Herramienta *case*: se seleccionó Visual Paradigm (VP) (22) debido a que esta herramienta posee licencia gratuita y comercial. Es un producto de mucha calidad, configurable para el uso en varios idiomas, multilinguaje y ofrece compatibilidad entre ediciones. Esta herramienta soporta Lenguaje Unificado de Modelado (UML) y provee el modelado de procesos de negocios.

2.2. Descripción de la solución

El desarrollo de Kinetic se centra fundamentalmente en la gestión de las evaluaciones y la interacción del especialista con las mismas. No se trata de obtener una aplicación completamente lista y liberada para utilizar en una consulta médica; sino demostrar a través de un prototipo de aplicación que los procedimientos para realizar las mediciones de la postura de un paciente utilizando técnicas de RA son posibles.

Existe una serie de condiciones iniciales que se deben tener en cuenta para realizar las mediciones en la solución que se propone. Primeramente se debe disponer de una cámara de video con capacidad de captura con una velocidad superior a 25 imágenes por segundo y luego un marcador que se coloca en una posición adyacente y fija al objeto o individuo sobre el que se desean hacer las mediciones. Tanto el objeto a medir como el marcador deben estar en el campo de visión de la cámara para el correcto funcionamiento del prototipo. El marcador presente en cada imagen se detecta, utilizando técnicas de

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

análisis y seguimiento de imágenes, con el objetivo de establecer el sistema de referencia global por medio del cual se posicionan las entidades geométricas (líneas verticales, horizontales y puntos) que se insertarán en la escena. Luego se realizan las mediciones de cálculo de distancia y ángulo, a partir de las posiciones de las entidades geométricas existentes en la escena. A continuación se expresan de forma más detallada todo el proceso.

2.2.1. Descripción del seguimiento

El proceso de medición que realiza Kinetic se vale de las ventajas que ofrece el *tracking* de ARToolKit. La biblioteca ARToolKit debe disponer de una imagen proveniente de una cámara que realiza la captura de video en tiempo real; esta imagen debe contener en su interior al menos un marcador totalmente visible para el correcto funcionamiento del *tracking*. Por tanto se decide acondicionar como datos de entrada al programa una imagen proveniente de la cámara de video.

Luego, utilizando técnicas de análisis de imágenes, se hace el seguimiento para calcular la transformación (posición y orientación) del marcador detectado con respecto a la cámara (ver Figura 6). Esto implica que se dispondrá permanentemente de un sistema de referencia virtual alineado con el sistema de referencia del objeto real, aplicándole las mismas transformaciones de traslación, rotación y escalado.

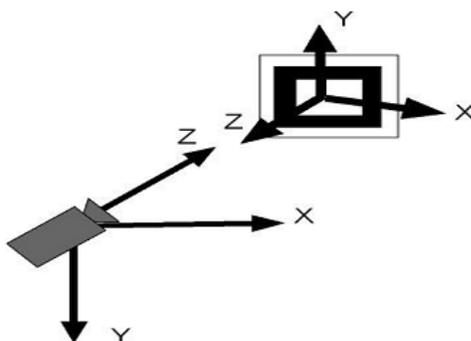


Figura 6. Relación cámara marcador.

Para realizar el proceso de seguimiento, el sistema debe conocer cierta información inicial proveniente de archivos de configuración. En los archivos, se almacenan los parámetros de la cámara, del dispositivo de visualización utilizado y los patrones de los marcadores que intervienen en la solución. Esta información es generada en una etapa previa al proceso de seguimiento. La información aumentada que representa las mediciones, se incorpora en la escena haciendo uso de las ventajas que ofrece OSGART, lográndose

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

así el aumento de la realidad dentro del sistema. Las funciones básicas, clases y métodos para operar sobre los datos visualizados en el escenario 3D, que representan la escena aumentada se realizan utilizando las funcionalidades del motor OSG. Para ello se hace necesario definir los nodos principales que van a componer el grafo de la escena, así como los elementos adicionales (instancias de clases) que ayudan en la gestión de la escena gráfica. Los nodos que componen el grafo de escena se describen en el siguiente epígrafe.

2.2.2. Organización de la escena según OSG

La organización de la escena de la solución aprovecha las ventajas del grafo de escena de OSG. De esta forma las entidades que aparecen derivadas de los conceptos que se manejan en la aplicación, se representan mediante las diferentes clases que provee *OpenSceneGraph*. La estructura de la escena se representa como una jerarquía de nodos a través de un grafo, como requiere OSG con un nodo raíz en el nivel cero que sirve como punto de entrada a la escena.

OSG propone la clase **Node** como la base para todos los nodos que pueden aparecer en el grafo de escena. Estos nodos pueden ser nodos de grupo o nodos hojas según la función que desempeñan dentro del grafo. La clase **Group** es la clase base para cualquier nodo intermedio que tiene hijos, no almacena geometría, sino información de configuración necesaria para organizar y dibujar una geometría, así como los estados que controlan su apariencia. Esta es una clase clave en la organización espacial de la escena que se propone.

La geometría que representa visualmente las mediciones introducidas en la escena aumentada, se representan en el grafo de escena utilizando instancias de la clase **Geode** (o nodo de geometría). Este tipo de nodo se representa como una hoja en el grafo de la escena y guarda la información que se va a visualizar. No tiene hijos como los nodos de agrupación, pero contiene objetos que se pueden dibujar definidos en OSG con la clase **Drawable**, los cuales contienen la geometría para la representación visual. La clase **Geometry** (un tipo de **Drawable**) se utiliza para almacenar los vértices y los comandos específicos de las figuras geométricas.

También se utiliza la clase **MatrixTransform** (Nodo grupo) que contiene una matriz para transformar la geometría de sus hijos, lo que permite a los objetos de la escena ser rotados, trasladados y escalados. Esta matriz aprovecha la notación de coordenadas homogéneas y está en correspondencia con las matrices de transformación de 4x4 que utiliza el estándar *OpenGL*.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

La utilización de OSG para el desarrollo de la solución permite al usuario interactuar y modificar el escenario gráfico de forma dinámica y en tiempo de ejecución. En la solución esto se logró mediante las clases **Callbacks** que define OSG para desencadenar acciones que modifican dinámicamente los nodos utilizando el patrón de diseño *Callback* (23). Estas son asignadas a objetos de tipo **Node** y **Drawable** y definen funciones que se ejecutan durante los procesos de dibujado y selección de visibilidad de OSG. Cuando se produce el evento al que se asoció una **Callback**, esta se registra para continuar invocándose por el sistema automáticamente mientras se recorre el grafo de la escena. OSG proporciona una interfaz básica para cualquier clase que necesite manejar eventos (de teclado y del *mouse* de manera independiente a la plataforma) de la Interfaz Gráfica de Usuario (GUI) a través de la clase **GUIEventHandler**. En la guía de inicio rápido de OSG (24) existe una información más detallada del grafo de escena que utiliza OSG.

En la solución se utilizaron las clases **Callbacks** para insertar, eliminar, así como trasladar y rotar las mediciones. Cada una de las mediciones se representó mediante geometrías. Las mediciones horizontales y verticales se representaron como pares de cilindros paralelos utilizando un **Drawable** para definir la geometría del cilindro y las transformaciones de cada cilindro se representaron mediante nodos **MatrixTransform**. Las mediciones angulares se representaron utilizando **Drawable** para definir la geometría de las esferas que delimitan el ángulo y las transformaciones de cada esfera se representaron mediante nodos **MatrixTransform**. De esta forma se formaron subgrafos como se muestran en la Figura 8 para cada una de las mediciones, que se insertan y se eliminan del grafo principal de la escena a medida que el usuario inserta y elimina mediciones.

2.2.3. Integración del seguimiento y la escena

La estructura del grafo de la escena de la solución que se propone está constituida de forma general por un visualizador representado por el **Viewer** que recorre una jerarquía de nodos para dibujar la información que estos representan. A través del **Viewer** el usuario es capaz de interactuar de forma dinámica y en con el sistema. La jerarquía de nodos está conformada por un nodo de grupo (raíz del grafo), el cual tiene como hijo un nodo cámara. Este nodo cámara se encarga de gestionar el video de fondo, proveniente de la cámara de video, a través de un nodo hijo de tipo **VideoLayer**. De igual modo, el nodo cámara contiene otro nodo hijo de tipo **MatrixTransform** que representa las transformaciones (traslación, rotación y escalado) de sus nodos hijos. Estas transformaciones se calculan a partir del marcador que está presente

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

en la imagen utilizando los algoritmos de *tracking* de ARToolkit. Estos datos de transformación se modifican dinámicamente y se actualizan en el grafo de escena por medio de **Callbacks** que se asocian al nodo **MatrixTransform** que sirve de referencia para insertar las mediciones en la imagen ver Figura 7.

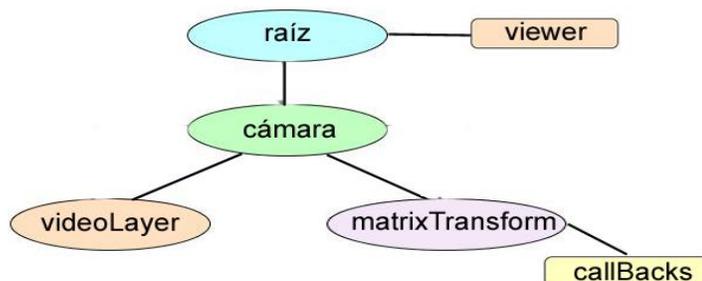


Figura 7. Grafo de la escena.

A partir de este nodo **MatrixTransform** es que se van a introducir los demás nodos que representan geoméricamente las mediciones que se van a insertar en la escena. Con el objetivo de mejorar la organización y comprensión del grafo de escena, se utilizan nodos para agrupar las estructuras de las mediciones horizontales, verticales y de ángulo como se muestra en la Figura 8. Estos nodos de grupo tienen como hijos las representaciones de cada una de las mediciones que se van a insertar en la escena. La estructura de grafo de escena en cada una de las mediciones presenta la ventaja que los objetos dibujables pueden ser compartidos y se muestran como si fueran visualmente independientes, consiguiendo un ahorro razonable de memoria en la medida que se insertan muchas mediciones. En todo momento se dispondrá de la posición de todos los objetos presentes en la escena accediendo a su **MatrixTransform** asociada, ver Figura 9 y a partir de aquí realizar los cálculos de mediciones.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

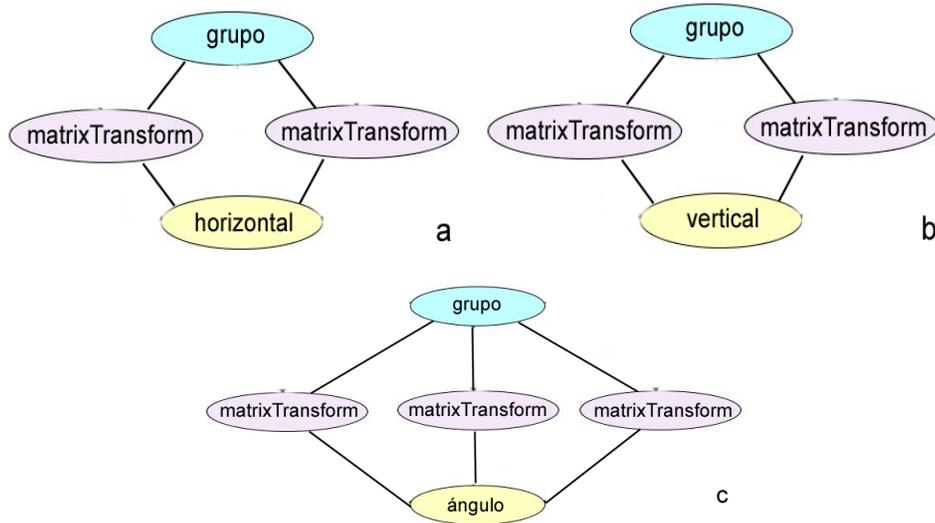


Figura 8. a) grafo medición horizontal, b) grafo medición vertical, c) grafo medición de ángulo.

La organización que se propuso para la escena permitió trabajar de forma sencilla con los datos numéricos que representan las transformaciones de las mediciones. De esta forma los cálculos de la distancia para las mediciones verticales y horizontales, así como el cálculo de la magnitud del ángulo en las mediciones angulares se realiza utilizando los valores que se almacenan en la última columna de las matrices de transformación, ver Figura 9.

$$\begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figura 9. Matriz de transformación.

2.2.4. Procedimientos matemáticos

Cálculo de distancia: para el cálculo de la distancia entre dos entidades geométricas el sistema utiliza la llamada distancia euclidiana, está basada en el teorema de Pitágoras y plantea que la distancia euclídea entre los puntos $P(p_1, p_2, \dots, p_n)$ y $Q(q_1, q_2, \dots, q_n)$, del espacio euclídeo n -dimensional, se define como:

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}. \quad (1)$$

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

Por tanto podemos definir que la distancia euclidiana entre dos puntos $P (p_1, p_2, p_3)$ y $Q (q_1, q_2, q_3)$ del espacio euclídeo 3-dimensional se define por:

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \quad (2)$$

Dónde:

- d_e es la distancia euclídea entre el punto P y Q.
- p_1, p_2 y p_3 es la posición del punto P en el espacio.
- q_1, q_2 y q_3 es la posición del punto Q en el espacio.

Cálculo de ángulo:

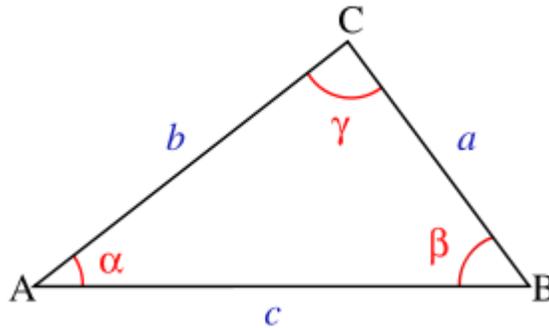


Figura 10. Cálculo de ángulo.

El ángulo que se forma entre tres entidades geométricas en la escena se calcula a partir de “el teorema del coseno” o también conocido como “teorema de Pitágoras generalizado” que plantea lo siguiente “Dado un triángulo formado por los puntos ABC, siendo α, β, γ , los ángulos, y a, b, c , los lados respectivamente opuestos a estos ángulos entonces”:

$$c^2 = a^2 + b^2 - 2ab \cos \gamma \quad (3)$$

Por tanto podemos definir que el ángulo γ que se forma entre los lados a y b está dado por:

$$\gamma = \arccos \frac{a^2 + b^2 - c^2}{2ab} \quad (4)$$

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

Los algoritmos de gráficos por computadoras hoy en día son capaces de brindar información muy exacta y con márgenes de error aceptables. Aunque no se descarta la posibilidad de la existencia de posibles errores con las mediciones, en función de la proyección perspectiva o cónica, pero que no es objetivo del desarrollo de esta investigación conocer tal margen de error, eso se determinará en futuras investigaciones.

2.3. Modelo del dominio

La metodología XP plantea que para un mejor entendimiento de las tareas, flujos y métodos de desarrollo de las funcionalidades se pueden crear diagramas, siempre que su creación no implique mayor esfuerzo que la implementación del mismo. Siguiendo este principio se elaboró el modelo del dominio Figura 11, el cual ayuda a comprender los conceptos más importantes del contexto y enlaza estos objetos unos con otros. La identificación y asignación de un nombre para estos objetos ayuda a desarrollar un glosario de términos que permitirá comunicarse mejor con los que trabajan en el sistema.

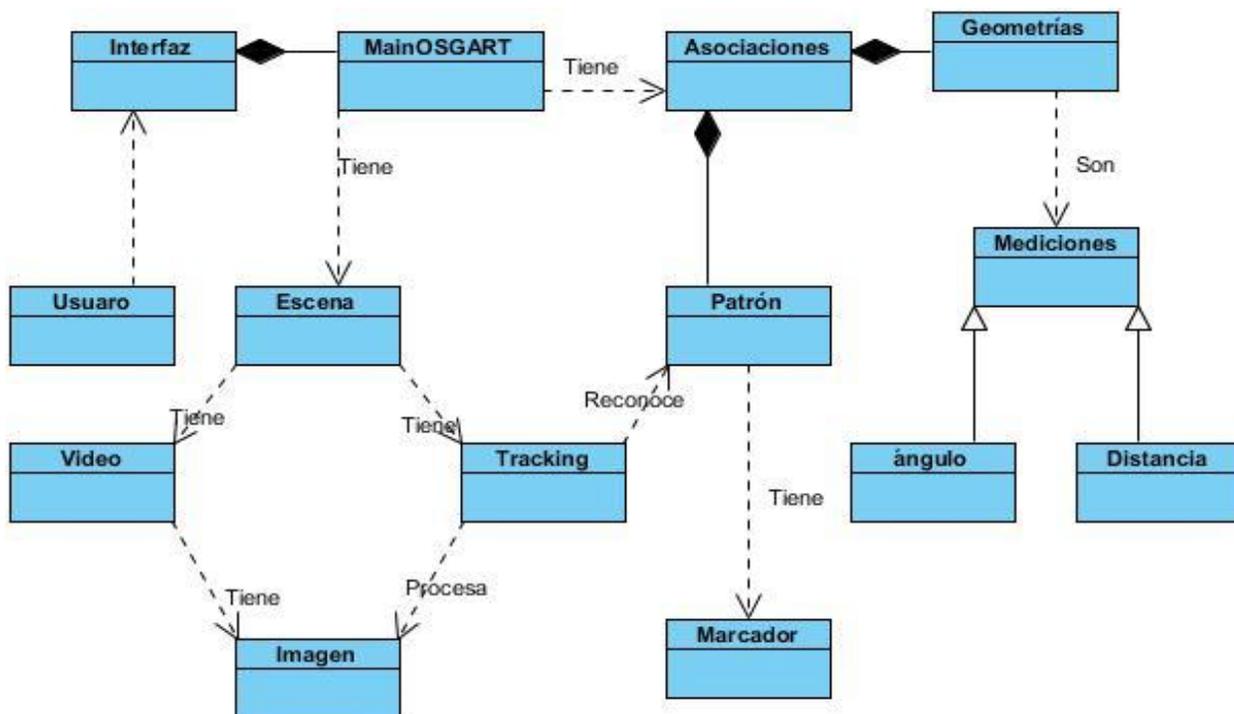


Figura 11. Modelo de dominio.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

2.3.1. Glosario de términos del dominio

Usuario: usuarios que interactúa con el sistema.

Interfaz: procesa la entrada y salida de datos para la aplicación.

MainOSGART: este es un concepto abstracto el cual expresa que es el encargado de manipular otros conceptos. Este tiene escena y asociaciones.

Asociaciones: relaciona los patrones con las mediciones.

Video: está compuesto por un conjunto de imágenes de la escena que se suceden.

Imagen: esta es generada por la cámara.

Tracking: permite añadir y detectar objetos de seguimiento, realiza el reconocimiento y los registros 3D.

Patrón: mapa de bits generado a partir de la región que ocupa un marcador determinado en cada fotograma del video capturado por la cámara.

Marcadores: figura impresa en un papel en blanco y negro.

2.4. Exploración

El ciclo de vida de un proyecto desarrollado con el uso de la metodología XP, se inicia con la fase de exploración, como resultado final, se obtienen las historias de usuario que se utilizarán para el desarrollo de las restantes etapas, las mismas son elaboradas teniendo en cuenta la lista de reserva del producto.

2.4.1. Actores del sistema

Actores	Descripción
Administrador.	Es un usuario con privilegios para la gestión de usuarios y pacientes en el sistema.
Doctor.	Es un usuario con privilegios para la gestión de pacientes y evaluaciones en el sistema.

Tabla 3. Actores del sistema.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

2.4.2. Lista de reserva del producto

Las listas de reservas de productos constituyen funcionalidades que el sistema debe cumplir. A continuación se enumeran cada uno de estos requerimientos, que servirán de apoyo para la creación de las historias de usuario:

1. Autenticar usuario.
2. Gestionar usuario.
 - 2.1. Añadir usuario.
 - 2.2. Eliminar usuario.
 - 2.3. Buscar usuario.
3. Gestionar paciente.
 - 3.1. Adicionar paciente.
 - 3.2. Eliminar paciente.
 - 3.3. Buscar paciente.
4. Gestionar evaluación.
 - 4.1. Adicionar evaluación.
 - 4.2. Eliminar evaluación.
 - 4.3. Buscar evaluación.
5. Emitir reporte.
6. Medir distancia.
7. Medir nivel.
8. Medir ángulo.

2.4.3. Historias de usuarios

Las historias de usuario son la técnica utilizada en XP para describir los requisitos del *software*. Las mismas son escritas por los clientes como las tareas que el módulo debe hacer y su construcción depende

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

principalmente de la habilidad que tenga el cliente para definir las (25). Seguidamente se describen las principales historias de usuario de la solución, las cuales tienen como entrada principal la lista de reserva del producto descrita anteriormente.

Historia de usuario	
Numero: 1	Usuario: Doctor, Administrador.
Nombre de Historia: Autenticar usuario.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 2
Descripción: cada usuario que se autentique tendrá determinado acceso a los datos del sistema según su rol (doctor, administrador).	

Tabla 4. HU Autenticar Usuario.

Historia de usuario	
Numero: 2	Usuario: Administrador.
Nombre de Historia: Gestionar usuario.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 2
Descripción: el administrador podrá insertar, eliminar y buscar todos los usuarios del sistema.	

Tabla 5. HU Gestionar usuario.

Historia de usuario	
Numero: 3	Usuario: Doctor, Administrador.
Nombre de Historia: Gestionar paciente.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 2
Descripción: los administradores y doctores podrán insertar, eliminar y buscar los pacientes existentes en el sistema.	

Tabla 6. HU Gestionar paciente.

Historia de usuario	
Numero: 4	Usuario: Doctor.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

Nombre de Historia: Gestionar evaluación.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 2
Descripción: los doctores podrán insertar, eliminar y buscar las evaluaciones en el sistema.	

Tabla 7. HU Gestionar evaluación.

Historia de usuario	
Numero: 5	Usuario: Doctor.
Nombre de Historia: Generar reporte.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Media.
Puntos estimados: 1	Iteración asignada: 1
Descripción: generar reporte.	

Tabla 8. HU Generar reporte.

Historia de usuario	
Numero: 6	Usuario: Doctor.
Nombre de Historia: Medir distancia.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alta.
Puntos estimados: 1	Iteración asignada: 1
Descripción: se medirá la distancia entre dos puntos.	

Tabla 9. HU Medir distancia.

Historia de usuario	
Numero: 7	Usuario: Doctor.
Nombre de Historia: Medir nivel.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alta.
Puntos estimados: 1	Iteración asignada: 1
Descripción: se medirá nivel entre dos rectas.	

Tabla 10. HU Medir nivel.

Historia de usuario	
Numero: 8	Usuario: Doctor.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

Nombre de Historia: Medir ángulo.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alta.
Puntos estimados: 1	Iteración asignada: 1
Descripción: se medirá el ángulo entre tres rectas.	

Tabla 11. HU Medir ángulo.

2.4.4. Diseño de caso de prueba

La metodología XP propone que después de que se realicen las historias de usuarios definidas por el cliente se deben diseñar los casos de pruebas que cada HU debe vencer en la fase de pruebas. Las pruebas de aceptación son consideradas pruebas de caja negra. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. (25)

Las tablas con los diseños de casos de pruebas se encuentran en el epígrafe 3.5 del capítulo 3 donde además se le incluyen los resultados arribados de la ejecución de las mismas.

2.5. Planificación

Una vez terminada la confección de las historias de usuario, se comienza con la creación del plan de entregas. Este plan tiene como objetivo brindarle a los desarrolladores una comprensión de la estimación de cada una de las historias. En esta fase los clientes establecen la prioridad de las mismas de acuerdo con sus necesidades inmediatas, para luego asignarlas a las iteraciones planificadas.

2.5.1. Plan de entrega

En el plan de entrega se definen las historias de usuario que se entregarán al final de cada iteración. El cronograma de entregas se realiza en función de las estimaciones de tiempos de desarrollo realizadas por los desarrolladores (25). En la Tabla 12 se muestran las versiones finales que serán publicadas.

No.	Historia de usuario	Iteración 1	Iteración 2
1	Autenticar usuario.	No empezado.	1.0
2	Gestionar usuario.	No empezado.	2.0
3	Gestionar paciente	No empezado.	2.0
4	Gestionar evaluación.	No empezado.	2.0
5	Emitir reporte.	1.0	Terminado.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

6	Medir distancia.	1.0	Terminado.
7	Medir nivel.	1.0	Terminado.
8	Medir ángulo.	1.1	Terminado.

Tabla 12. HU Plan de entrega.

2.5.2. Estimaciones del esfuerzo

Las estimaciones del esfuerzo para realizar las historias de usuario, permiten tener una medida de la velocidad del proyecto y brindan una guía a la cual ajustarse. Los resultados estimados se muestran en la Tabla 13.

No.	Historia de usuario	Puntos	Prioridad
1	Autenticar usuario.	1	Media
2	Gestionar usuario.	1	Media
3	Gestionar paciente.	1	Media
4	Gestionar evaluación.	1	Media
5	Emitir reporte.	1	Alta
6	Medir distancia.	1	Alta
7	Medir nivel.	1	Alta
8	Medir ángulo.	1	Alta

Tabla 13. Plan de estimación de esfuerzo por HU.

2.5.3. Plan de iteraciones

El plan de iteraciones tiene como entrada la relación de historias de usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha historia. La duración de la historia de usuario es el valor real que se le asignó en la misma, esta duración se expresa en semanas. Como resultado de la priorización de historias de usuarios se planificaron las siguientes iteraciones:

Primera iteración: en esta iteración se implementan las HU que tienen prioridad “Alta” en el negocio, de esta forma, se crean las funcionalidades principales del sistema. Estas HU son: 5, 6, 7 y 8. Al finalizar dicha iteración se tendrá la primera versión de prueba.

PROPUESTA Y DESCRIPCIÓN DE LA SOLUCIÓN

Segunda Iteración: el objetivo fundamental de la última iteración está dado por la implementación de aquellas HU que tienen una prioridad media dentro del negocio. Con la culminación de la misma se tendrán implementadas las peticiones del cliente descritas en las historias de usuarios 1, 2, 3 y 4. Al término de esta iteración se contará con la versión final de la solución.

2.5.4. Plan de duración de iteraciones

El objetivo fundamental del plan de duración de las iteraciones es mostrar la duración de cada iteración y el orden en que serán implementadas las historias de usuarios en cada una de iteración.

Iteraciones	No.	Historia de usuarios	Duración por HU	Duración por iteraciones
Primera	5	Emitir reporte.	6 días	4 semanas
	6	Medir distancia.	6 días	
	7	Medir nivel.	6 días	
	8	Medir ángulo.	6 días	
Segunda	1	Autenticar usuario.	6 días	4 semanas
	2	Gestionar usuario.	6 días	
	3	Gestionar paciente.	6 días	
	4	Gestionar evaluación.	6 días	

Tabla 14. Plan de duración de iteraciones

2.6. Conclusiones parciales

En este capítulo se detalló la estructura de solución técnica propuesta para cumplir con el objetivo de la investigación. Las características y funcionalidades fundamentales se detallaron a través de las historias de usuarios en la fase de exploración y la obtención del plan de iteraciones en la fase de planificación, constituyen las entradas fundamentales para pasar a la construcción de la solución.

CONSTRUCCIÓN DE LA SOLUCIÓN

CAPÍTULO 3. CONSTRUCCIÓN DE LA SOLUCIÓN

En presente capítulo se describen las tarjetas CRC, se presenta el estándar de codificación y los patrones de diseño utilizado como aspectos fundamentales del diseño. Además se realizan las pruebas a las historias de usuarios.

3.1. Diseño del sistema

La metodología XP no requiere necesariamente la descripción del sistema por medio de diagramas de clase que hacen uso de la notación UML, porque esta se guía por técnicas como las tarjetas Contenido, Responsabilidad y Colaboración (CRC). Esto no implica que no se utilicen los diagramas para obtener una mejor visión y comunicación entre el equipo de trabajo, siempre y cuando su complejidad no sea alta y defina información importante.

3.1.1. Tarjetas CRC

Las tarjetas CRC son una técnica de modelado orientado a objeto que permite identificar clases y sus responsabilidades, se hacen con el objetivo de identificar jerarquías de generalización/especificación, o jerarquías de agregación entre las clases, de manera que ayuda al refinamiento de clases. Las tarjetas CRC definen como finalidad obtener un diseño simple y no incurrir en la implementación de características que no son necesarias.

CMainOSGART	
Responsabilidad	Colaboración
Obtener el nodo raíz de la escena. Obtener cámara. Obtener video. Obtener <i>tracker</i> . Inicializar escena aumentada. Crear la escena. Leer fichero de configuración de la escena.	CVideo. CTracker.

Tabla 15. Tarjeta CRC – CMainOSGART.

CVideo

CONSTRUCCIÓN DE LA SOLUCIÓN

Responsabilidad	Colaboración
Inicializar el video. Abrir video. Iniciar video. Detener video. Cerrar video. Obtener el video de fondo.	

Tabla 16. Tarjeta CRC – CVideo.

CTracker	
Responsabilidad	Colaboración
Inicializar el <i>tracking</i> . Cargar la calibración. Cambiar el video. Leer marcador.	

Tabla 17. Tarjeta CRC – CTracker.

El resto de las tarjetas CRC se encuentran disponibles en el ANEXO 1.

3.1.2. Patrones de diseño

Los patrones son soluciones comunes a problemas de diseño de *software* orientado a objetos y que además poseen ciertas características de efectividad para resolver ese problema. Son reusables, ya que pueden ser aplicados en otros diseños o problemas (26).

En la clase CMainOSGART se hace uso del patrón creacional **Singleton** para garantizar la existencia de una sola instancia de la clase, puesto a que solo se necesita un único punto de acceso global al objeto.

3.1.3. Modelo de datos

El modelo de datos escribe la representación lógica y física de los datos persistentes usados por la aplicación. El modelo de datos utilizado por la solución posee un nivel de complejidad medio, como consecuencia de que está compuesto por 3 tablas, así como las relaciones entre ellas Figura 12.

CONSTRUCCIÓN DE LA SOLUCIÓN

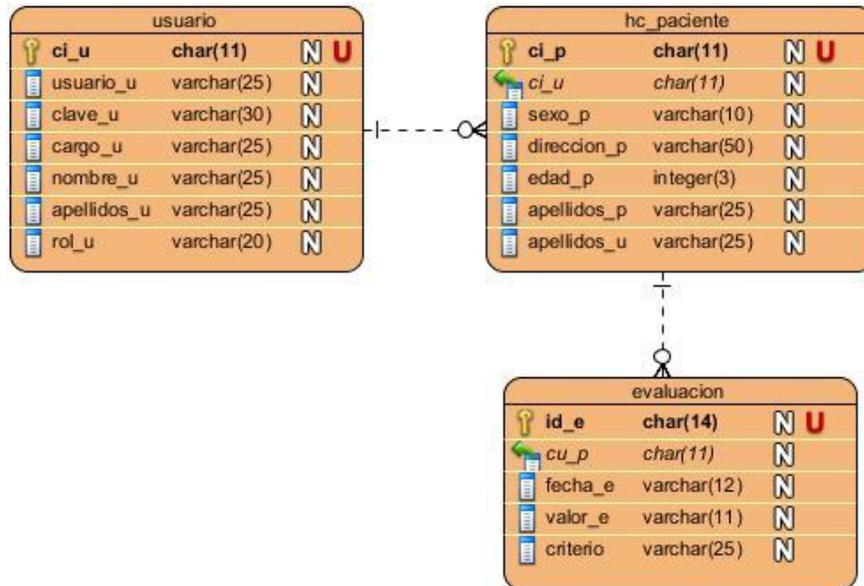


Figura 12. Modelo de datos.

3.2. Estándar de codificación

“Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código” (27). La metodología XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación. Seguidamente se enumeran una serie de convenciones de denominación del código origen OSG.

Clases: en OSG las clases comienzan con una letra mayúscula. Si el nombre de la clase se compone de varias palabras, cada palabra comienza con una letra mayúscula Figura 13.

```
class CVideo
{
```

Figura 13. Declaración de clases

Métodos de las clases: los nombres de métodos dentro de una clase OSG deben comenzar con minúsculas. Si el nombre del método se compone de varias palabras, cada palabra adicional comienza con una letra mayúscula Figura 14.

CONSTRUCCIÓN DE LA SOLUCIÓN

```
void Kinetic::evaluacionHorizontal()  
{
```

Figura 14. Métodos de las clases.

Variables miembros de clase: los nombres de las variables de una clase deben comenzar con minúsculas. Si el nombre de la variable se compone de varias palabras, cada palabra adicional comienza con una letra mayúscula Figura 15.

```
private:  
    QString driver;  
    QString nombreBD;  
    QString ipServidor;  
    QString usuario;  
    QString contrasenna;  
    int puerto;
```

Figura 15. Declaración de variables.

3.3. Desarrollo de Iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una de las iteraciones un entregable funcional que implementa las historias de usuario asignadas a la iteración. Para lograr un análisis y desarrollo más detallado de las HU, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante la fase de desarrollo de iteraciones del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto.

3.4. Implementación

En esta fase se realiza la implementación de las HU que fueron seleccionadas por cada iteración. Primero se lleva a cabo un chequeo del plan de iteraciones por si es necesario realizar modificaciones. Como parte de este plan se crean tareas de programación para ayudar a organizar la implementación exitosa de las HU.

Tareas por cada historia de usuario

CONSTRUCCIÓN DE LA SOLUCIÓN

Con el objetivo de profundizar sobre los artefactos generados en las fases de Exploración y Planificación, se creó un grupo de actividades por cada historia de usuario de la iteración 1 Tabla 18 y la iteración 2 Tabla 19.

Iteración 1:

No.	Historias de usuario	No.	Tareas por historias de usuario
5	Emitir reporte.	5.1	Crear una interfaz para generar reporte.
		5.1	Generar reporte histórico.
6	Medir distancia.	6.1	Crear una interfaz para medir distancia.
		6.2	Medir distancia.
7	Medir nivel.	7.1	Crear una interfaz para medir niveles.
		7.2	Medir nivel.
8	Medir ángulo.	8.1	Crear una interfaz para medir ángulo.
		8.2	Medir ángulo.

Tabla 18. Tareas de ingeniería de la iteración 1.

Iteración 2:

No.	Historia de usuario	No.	Tareas por historia de usuario
1	Autenticar usuario.	1.1	Crear interfaz para el ingreso al <i>software</i> .
		1.2	Comprobar datos para el ingreso al <i>software</i> .
2	Gestionar usuario.	2.1	Insertar usuario en el sistema.
		2.2	Modificar usuario en el sistema.
		2.3	Eliminar usuario del sistema.
		2.4	Buscar usuario en el sistema.
3	Gestionar paciente.	3.1	Insertar paciente en el sistema.
		3.2	Modificar paciente en el sistema.
		3.3	Eliminar paciente del sistema.
		3.4	Buscar paciente en el sistema.
		4.1	Insertar evaluación en el sistema.

CONSTRUCCIÓN DE LA SOLUCIÓN

4	Gestionar evaluación.	4.2	Eliminar evaluación del sistema.
		4.3	Buscar evaluación en el sistema.

Tabla 19. Tareas de ingeniería de la iteración 2.

Descripción de las tareas de ingeniería: una vez que se conozcan las tareas creadas por cada historia de usuario es necesario realizar una descripción general de cada una de las mismas, identificando el número de la tarea, programador encargado y el número de la historia de usuario a la que pertenece.

Las tareas ingenierías de las historias de usuarios correspondientes a la iteración 1 pueden ser localizadas en el ANEXO 2. El resto de las mismas y con prioridad media para el cliente y pertenecientes a la iteración 2 pueden ser localizadas en el ANEXO 3.

3.5. Pruebas de aceptación

Iteración 1: según la planificación de las iteraciones realizadas en el Capítulo 2 epígrafe 2.7, en la primera iteración se deben probar las historias de usuario que tienen una prioridad alta para el negocio. En esta iteración se diseñan y aplican los casos de prueba para las siguientes historias de usuario:

HU5: Emitir reporte.

HU6: Medir distancia.

HU7: Medir nivel.

HU8: Medir ángulo.

Caso de prueba HU5 Emitir Reporte: esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: emitir reporte.

Descripción de la funcionalidad: en esta historia hay que comprobar la emisión de un reporte. Si la selección del paciente y generación del reporte son satisfactorias se mostrará un mensaje indicándolo. Si existen errores se mostrarán mensajes indicando el contrario.

Código: HU5-ER-I1	Historia de usuario: 5. Emitir reporte.
Escenario: hacer reporte.	
Condiciones de ejecución: <ul style="list-style-type: none">• El doctor tiene que estar autenticado en el sistema.	

CONSTRUCCIÓN DE LA SOLUCIÓN

Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Seleccionar del listado de pacientes al que se le desea realizar el reporte y emitir el reporte.	El sistema debe exportar un fichero (*.pdf) con la información del paciente.	Satisfactorio.	
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Emitir reporte sin seleccionar el paciente de la lista.	El sistema debe mostrar un mensaje de error informando: "Seleccione el paciente."	Satisfactorio.	
Evaluación de la prueba: satisfactoria.			

Tabla 20. Prueba de aceptación emitir reporte.

Caso de prueba HU6 Medir Distancia: A continuación se cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: medir distancia.

Descripción de la funcionalidad: En esta historia se debe comprobar la realización de mediciones de distancia. Si la selección del paciente y las evaluaciones son satisfactorias se mostrará un mensaje indicándolo. Si existen errores se mostrarán mensajes indicando lo contrario.

Código: HU6-MD-I1	Historia de usuario: 6 - Medir distancia.		
Escenario: medición horizontal, medición vertical.			
Condiciones de ejecución:			
<ul style="list-style-type: none"> • El usuario doctor tiene que estar autenticado en el sistema. • La captura de video debe estar iniciada. • Se debe activar el tipo de medición a realizar (vertical u horizontal). 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Seleccionar del listado	El sistema debe registrar en la	Satisfactorio.	

CONSTRUCCIÓN DE LA SOLUCIÓN

de pacientes al que se le realizara la medición de distancia e insertar la medición.	base de datos la nueva medición de distancia y muestra un mensaje informando que la evaluación fue realizada "Evaluación correcta".		
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Realizar evaluación sin seleccionar el paciente de la lista.	El sistema debe mostrar un mensaje de error, informando "Seleccione el paciente".	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 21. Prueba de aceptación medir distancia.

Caso de prueba HU7 Medir Nivel: a continuación se cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: medir nivel.

Descripción de la funcionalidad: en dicha historia se debe comprobar la realización de mediciones de nivel. Si la selección del paciente y las evaluaciones son satisfactorias se mostrará un mensaje indicándolo. Si existen errores se mostrarán mensajes indicando lo contrario.

Código: HU7-MN-I1	Historia de usuario: 7 - Medir nivel.		
Escenario: medición horizontal, medición vertical.			
Condiciones de ejecución:			
<ul style="list-style-type: none"> • El usuario doctor tiene que estar autenticado en el sistema. • La captura de video debe estar iniciada. • Se debe activar el tipo de medición a realizar (vertical u horizontal). 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Seleccionar del listado de pacientes al que se le realizara la	El sistema debe registrar en la base de datos la nueva medición de nivel y muestra	Satisfactorio.	

CONSTRUCCIÓN DE LA SOLUCIÓN

medición de nivel e insertar la medición de nivel.	un mensaje informando que la evaluación fue realizada "Evaluación correcta".		
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Realizar evaluación sin seleccionar el paciente de la lista.	El sistema debe mostrar un mensaje de error, informando "Seleccione el paciente".	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 22. Prueba de aceptación medir nivel.

Caso de prueba HU8 Medir Ángulo: a continuación se cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: medir ángulo.

Descripción de la funcionalidad: en esta historia se debe comprobar la realización de mediciones de ángulos. Si la selección del paciente y las evaluaciones son satisfactorias se mostrará un mensaje indicándolo. Si existen errores se mostrarán mensajes indicando lo contrario.

Código: HU8-MA-11	Historia de usuario: 8 - Medir ángulo.		
Escenario: medición angular.			
Condiciones de ejecución:			
<ul style="list-style-type: none"> • El usuario doctor tiene que estar autenticado en el sistema. • La captura de video debe estar iniciada. • Se debe activar la medición angular. 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Seleccionar del listado de pacientes al que se le realizara la medición angular e insertar la medición de angular.	El sistema debe registrar en la base de datos la nueva medición angular y muestra un mensaje informando que la evaluación fue realizada	Satisfactorio.	

CONSTRUCCIÓN DE LA SOLUCIÓN

	"Evaluación correcta".		
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Realizar evaluación sin seleccionar el paciente de la lista.	El sistema debe mostrar un mensaje de error informando: "Seleccione el paciente".	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 23. Prueba de aceptación medir ángulo.

Iteración 2: en esta iteración se diseñan y aplican los casos de prueba para las siguientes historias de usuario:

HU1: Autenticar usuario.

HU2: Gestionar usuario.

HU3: Gestionar paciente.

HU4: Gestionar evaluación.

Los casos de pruebas relacionados con las historias de usuarios a probar en la iteración 2 se encuentran disponibles en el ANEXO 4.

3.6. Pruebas de validación

Luego de desarrollar la solución mediante un prototipo de aplicación se decidió hacer un conjunto pruebas de las mediciones utilizando medidas conocidas para comprobar los resultados. Para validar los resultados de las mediciones de la solución se dispuso de los siguientes materiales: una cámara de video (*webcam*), un marcador y un objeto con dimensiones conocidas (maqueta del cuerpo humano). Tanto el objeto a medir como el marcador se colocaron en el campo de visión de la cámara. Las mediciones de prueba se hicieron a una distancia de 1 y 2 metros respectivamente. A partir de esta configuración de la escena se tomaron varias mediciones sobre las medidas conocidas para comprobar si coincidían los valores respectivamente. Para los ángulos se diseñó una medida similar a la desviación de la cabeza con respecto a la vertical del cuerpo y para las mediciones horizontales y verticales se diseñó una medida de desnivel de hombros y otra de ancho de cuello. El escenario de prueba se muestra en la Figura 16.

CONSTRUCCIÓN DE LA SOLUCIÓN

En la Figura 17 se muestra la interfaz del sistema durante ejecución de las pruebas realizadas al prototipo funcional. Los resultados obtenidos son expuestos en la Tabla 24.

Objeto	Medición real	Software 1 m	Software 2 m
Ancho del cuerpo.	46.00 cm	47.00 cm	49.10 cm
Nivel entre hombros.	3.00 cm	3.00 cm	3.10 cm
Desviación de la cabeza	33 grados	33 grados	33 grados
Ancho del cuello	13.00 cm	13.00 cm	14.00 cm

Tabla 24. Pruebas de validación.

Los resultados arrojados por la solución, muestran que, con los materiales utilizados, para distancias relativas a 1 m los datos variaban poco. Por otro lado al aumentar la distancia entre la cámara y el objeto al que se le realizan las mediciones, el margen de error de las mediciones horizontales y verticales aumenta como resultado de la proyección perspectiva o cónica apreciándose distorsiones en las líneas rectas. Es válido aclarar que las medidas de ángulos no variaron durante el experimento.



Figura 16. Escenario de prueba.

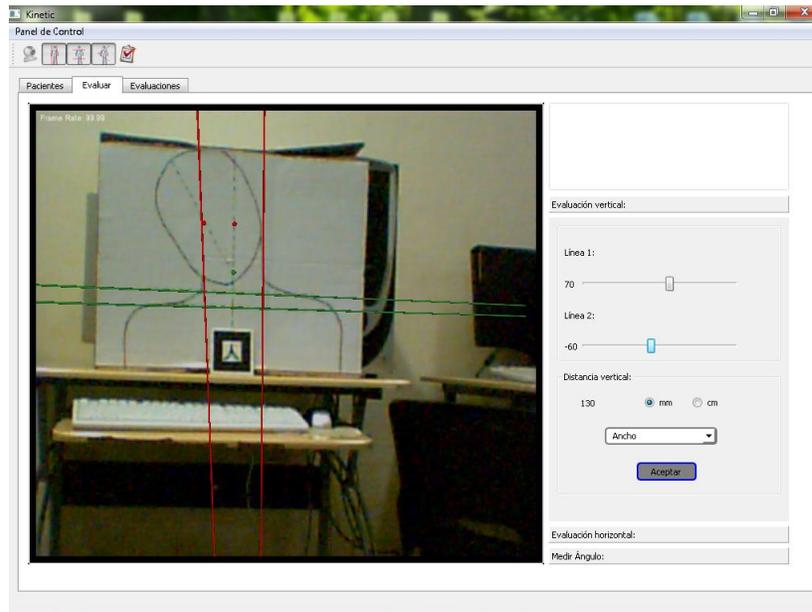


Figura 17: Interfaz en fase de pruebas horizontales.

3.7. Conclusiones parciales del capítulo

En este capítulo se realizó el diseño, la implementación y las pruebas del prototipo funcional. Se obtuvieron las tarjetas CRC, se definieron los patrones de diseños que se utilizan en el desarrollo del sistema y se presentó el estándar de codificación utilizado. Por último se les realizó las pruebas de aceptación al prototipo funcional.

CONCLUSIONES

Al término de la investigación, se implementó un prototipo de *software*, el cual permite establecer mediciones de distancias y ángulos sobre objetos de la realidad utilizando técnicas de RA. Con la realización de pruebas a la solución implementada, se pudo comprobar que los resultados de las mediciones se vieron afectados por la proyección cónica, a medida los objetos a medir se situaban más lejos del dispositivo de captura de la escena. Finalmente, teniendo en cuenta que se utilizó un *webcam* de muy baja resolución, se puede expresar que los resultados fueron aceptables y se pudo demostrar que el procedimiento de evaluación de la postura es viable con la utilización de técnicas de RA.

RECOMENDACIONES

En el análisis de los resultados se pudo comprobar que las mediciones se veían afectadas por la proyección cónica. Por tal motivo se recomienda hacer un estudio para determinar la relación del error en la medición en función de la distancia y la proyección. Utilizar dispositivos de captura de mejor calidad en función de la resolución y la nitidez de la imagen.

REFERENCIAS BIBLIOGRÁFICAS

1. **Rossi, Ph.D. Mark Damian.** Postural evaluation. s.l. : Encyclopedia of Nursing and Allied Health, 02 27, 2010. [Citado el: 08 de 02 de 2012.].
2. *Estudio de validación de un instrumento de evaluación postural (SAM, spinal analysis machine).* **Olaru, Ángela, Farré, Joan Parra and Matas, Ramon Balius.** [ed.] Medicina de l esport. 02 04, 2007, Revistes Catalanes amb Accés Obert (RACO). [Citado el: 05 de 02 de 2012].
3. **Ganong, W.F.** Fisiología Médica. *El Manual Moderno.* México : s.n., 1992.
4. **Nevatia, R.** Machine perception. [Online] 1982. <http://homepages.inf.ed.ac.uk/rbf/BOOKS/NEVATIA/toc.htm>.
5. <http://pilicita.jimdo.com>. [Online] [Cited: 2 19, 2012.] <http://pilicita.jimdo.com/realidad-virtual>.
6. **Azuma, Ronald T.** A Survey of Augmented Reality. In *Presence: Teleoperators and Virtual Environments.* 1997, Vol. 6, 4, pp. 355-385. <http://www.cs.unc.edu/~azuma/ARpresence.pdf>.
7. **Borro, Diego.** *Introducción a la Realidad Aumentada.* Universidad Autónoma de Occidente, UAO. Santiago de Cali, Colombia : s.n., 2005.
8. **INC, BIOTONIX.** 2010. [Citado el: 05 de 02 de 2012]. <http://biotonix.com>.
9. **Zaki, A.** Posture Evaluation Instruments and Anthropometers. [HOSPEQ, Inc]. 2009. [Citado el: 01 de 02 de 2012.].
10. **PostureCo.** *Software Developer for Posture Ray and Posture Screen.* s.l. : Software Developer for Posture Ray and Posture Screen. [Citado el: 20 de 12 de 2011]. http://findarticles.com/p/articles/mi_gGENH/is_20050229/ai_2699003630/.
11. **BTS.** *BTS Digivec.* s.l. : BTSbioengineering. Solución de realidad aumentada para el análisis estático y dinámico de la postura. [Citado el: 10 de 02 de 2012]. http://www.btsbioengineering.com/resources/Company/Media-center/Brochure-Library/BTSMED_DIGIVEC-0611ES_A4.pdf .
12. **Waterworth, John.** Executive summary. [Online] 1999. [Cited: 05 05, 2012.] <http://www8.informatik.umu.se/~jwworth/2Techniques#2.4>.
13. **HITLAB.** ARToolKit Augmented Reality ToolKit. [Online] 2006. [Cited: 1 30, 2012.] <http://www.hitl.washington.edu/artoolkit> .
14. **OpenSceneGraph.** OpenSceneGraph.org. [Online] OpenSceneGraph. [Cited: 2 15, 2012.] <http://www.openscenegraph.org/projects/osg/wiki/About/Introduction>.

REFERENCIAS BIBLIOGRÁFICAS

15. **OSGART**. osgart.org. [Online] [Cited: 2 15, 2012.] http://osgart.org/wiki/index.php/Main_Page.
16. **PIATTINI, M.** Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. 1996.
17. **Canós, José H, Letelier, Patricio and Penadé, Carmen.** Metodologías Ágiles en el Desarrollo de Software. [Online] 2008. [Cited: 03 15, 2012.] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.4553&rep=rep1&type=pdf#page=9>.
18. **Leteliers, Patricio and Penadé, Carmen.** Metodologías ágiles para el desarrollo de software:eXtreme Programming (XP). [Online] Universidad Politécnica de Valencia. [Cited: 04 15, 2012.] www.willydev.net/descargas/masyxp.pdf.
19. **Galves, Ing.Jorge.** Fundamnetos de la Metodología RUP. [Online] Universidad Tecnológica de Pereira, 09 16, 2007. [Cited: 04 16, 2012.] <http://es.scribd.com/doc/297224/RUP>.
20. qt.nokia.com. [Online] nokia. [Cited: 02 02, 2012.] [http://qt.nokia.com/products/..](http://qt.nokia.com/products/)
21. **postgresql.org.** www.postgresql.org.es. [Online] postgresql.org. [Cited: 05 15, 2012.] http://www.postgresql.org.es/sobre_postgresql.
22. Visual Paradigm. [Online] [Cited: 02 02, 2012.] <http://www.visual-paradigm.com/aboutus/10reasons.jsp>.
23. **Gamma, Erich, et al., et al.** *Design Patterns: Elements of Reusable Object-Oriented Software*. 1995.
24. **Martz, Paul** . *OpenSceneGraph Quick Start Guide*. 7 April 2007.
25. **Joskowicz, José.** Reglas y Prácticas en eXtreme Programming. [Online] 2008. [Cited: 04 15, 2012.] <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.
26. **Rojas, Mc. Juan Carlos.** Patrones de Diseño. [Online] 05 2007. [Cited: 03 15, 2012.] <http://antares.itmorelia.edu.mx/~jcolivar/courses/dp07b/patrones.pdf>.
27. **Fernández, Carlos.** Estándar para codificación en lenguaje C++. [Online] 2008. [Cited: 3 15, 2012.] <http://progra.iteso.mx/estandares/estandar%20codificacion%20c%2B%2B/estandardcodificacion.pdf>.
28. **Bruegge, Bernd and Klinker, Gudrun.** DWARF Distributed Wearable Augmented Reality Framework. [Online] [Cited: 02 14, 2012.] <http://campar.in.tum.de/twiki/pub/Chair/DwarfWhitePaper/dwarf.pdf>.
29. **artoolworks.** artoolworks.com. [Online] artoolworks. [Cited: 2 15, 2012.] <http://www.artoolworks.com/community/osgart/docs.html>.

REFERENCIAS BIBLIOGRÁFICAS

30. New York State Physical Fitness Test for Boys and Girls Grades 4-12. *A Manual for Teachers of Physical Education*. 1956.
31. Association, International Chiropractors. *Association, International Chiropractors*. [Online] <http://www.chiropractic.org>.
32. **Moreno Mirabal, Mileydi and de la Cruz Guevara Ramírez, Ernesto**. *Localización de objetos virtuales en el mundo real con técnicas de Realidad Aumentada*. Ciudad de La Habana : s.n., 2008. pp. 30-43.
33. **INC, BIOTONIX**. *FitPrint*. 2008. [Citado el: 15 de 1 de 2012.]. http://biotonix.com/fitness_en.php.
34. —. *BioPrint*. 2008. [Citado el: 15 de 01 de 2012.]. http://biotonix.com/spine_en.php.
35. **Martz, Paul** . *OpenSceneGraph Quick Start Guide*. 2007.

ANEXOS.

ANEXO 1. Tarjetas CRC

CPunto	
Responsabilidad	Colaboración
Calcular distancia. Calcular ángulo.	

Tabla 25. Tarjeta CRC – CPunto.

CBuscar	
Responsabilidad	Colaboración
Buscar nodo.	

Tabla 26. Tarjeta CRC – CBuscar.

CUusuario	
Responsabilidad	Colaboración
Obtener atributos de un usuario Cambiar los atributos de un usuario.	

Tabla 27. Tarjeta CRC – CUusuario.

CPaciente	
Responsabilidad	Colaboración
Obtener atributos de un paciente. Cambiar los atributos de un paciente.	

Tabla 28. Tarjeta CRC – CPaciente.

CConexion	
Responsabilidad	Colaboración

Crear la conexión. Insertar datos. Extraer datos. Actualizar datos. Eliminar datos.	
---	--

Tabla 29. Tarjeta CRC – CConexion.

CReporte	
Responsabilidad	Colaboración
Crear reporte. Obtener reporte. Generar reporte.	

Tabla 30. Tarjeta CRC – CReporte.

CKinetic	
Responsabilidad	Colaboración
Insertar usuario. Eliminar usuario. Buscar usuario. Insertar paciente. Eliminar paciente. Buscar paciente. Insertar evaluación horizontal. Insertar evaluación vertical. Insertar evaluación ángulo. Eliminar evaluación. Realizar medición vertical. Realizar medición horizontal. Realizar evaluación de ángulo. Generar reporte.	CBuscar. CReporte. CPunto. CConexion. CUsuario.

Tabla 31. Tarjeta CRC – CKinetic.

ANEXO 2. Tareas por historia de usuario iteración 1

Tarea de ingeniería	
Numero Tarea: 5.1	Numero Historia: 5
Nombre Tarea: Crear una interfaz para generar reporte.	
Tipo de Tarea: Diseño.	Puntos estimados: 0.5
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para seleccionar al paciente que se le generará el reporte.	

Tabla 32. Tarea de ingeniería 5.1.

Tarea de ingeniería	
Numero Tarea: 5.2	Numero Historia: 5
Nombre Tarea: Generar reporte histórico.	
Tipo de Tarea: Desarrollo.	Puntos estimados: 0.5
Responsable: José Luis Simón Peña.	
Descripción: Se genera un documento PDF con la información del paciente seleccionado.	

Tabla 33. Tarea de ingeniería 5.2.

Tarea de ingeniería	
Numero Tarea: 6.1	Numero Historia: 6
Nombre Tarea: Crear una interfaz para medir distancia.	
Tipo de Tarea: Diseño.	Puntos estimados: 0.5
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para realizar mediciones de distancia.	

Tabla 34. Tarea de ingeniería 6.1.

Tarea de ingeniería	
Numero Tarea: 6.2	Numero Historia: 6

Nombre Tarea: Medir distancia.	
Tipo de Tarea: Desarrollo.	Puntos estimados: 0.5
Responsable: José Luis Simón Peña.	
Descripción: Se medirá la distancia entre dos puntos seleccionados por el usuario.	

Tabla 35. Tarea de ingeniería 6.2.

Tarea de ingeniería	
Numero Tarea: 7.1	Numero Historia: 7
Nombre Tarea: Crear una interfaz para medir niveles.	
Tipo de Tarea: Diseño.	Puntos estimados: 0.5
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para realizar mediciones de nivel.	

Tabla 36. Tarea de ingeniería 7.1.

Tarea de ingeniería	
Numero Tarea: 7.2	Numero Historia: 7
Nombre Tarea: Medir nivel.	
Tipo de Tarea: Desarrollo.	Puntos estimados: 0.5
Responsable: José Luis Simón Peña.	
Descripción: Se medirá el nivel entre dos puntos seleccionados por el usuario.	

Tabla 37. Tarea de ingeniería 7.2.

Tarea de ingeniería	
Numero Tarea: 8.1	Numero Historia: 8
Nombre Tarea: Crear una interfaz para medir ángulo.	
Tipo de Tarea: Diseño.	Puntos estimados: 0.5
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para realizar mediciones de ángulo.	

Tabla 38. Tarea de ingeniería 8.1.

Tarea de ingeniería	
Numero Tarea: 8.2	Numero Historia: 8
Nombre Tarea: Medir ángulo.	
Tipo de Tarea: Desarrollo.	Puntos estimados: 0.5
Responsable: José Luis Simón Peña.	
Descripción: Se medirá el ángulo entre tres puntos seleccionados por el usuario.	

Tabla 39. Tarea de ingeniería 8.2.

ANEXO 3. Tareas por historia de usuario iteración 2

Tarea de ingeniería	
Numero Tarea: 1.1	Numero Historia: 1
Nombre Tarea: Crear interfaz para el ingreso al <i>software</i> .	
Tipo de Tarea: Diseño.	Puntos estimados: 0.5
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para que el usuario se autentique en el sistema.	

Tabla 40. Tarea de ingeniería 1.1.

Tarea de ingeniería	
Numero Tarea: 1.2	Numero Historia: 1
Nombre Tarea: Comprobar datos para el ingreso al <i>software</i> .	
Tipo de Tarea: Desarrollo.	Puntos estimados: 0.5
Responsable: José Luis Simón Peña.	
Descripción: Se comprobará los permisos de acceso al <i>software</i> .	

Tabla 41. Tarea de ingeniería 1.2.

Tarea de ingeniería	
Numero Tarea: 2.1	Numero Historia: 2
Nombre Tarea: Insertar usuario en el sistema.	
Tipo de Tarea: Diseño, Desarrollo	Puntos estimados: 0.25
Responsable: José Luis Simón Peña.	

Descripción: Se diseñará una interfaz para insertar usuario y se insertarán los datos en la base de datos.

Tabla 42. Tarea de ingeniería 2.1.

Tarea de ingeniería	
Numero Tarea: 2.2	Numero Historia: 2
Nombre Tarea: Eliminar usuario del sistema.	
Tipo de Tarea: Diseño, Desarrollo	Puntos estimados: 0.25
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para eliminar el usuario y es eliminado en la base de datos.	

Tabla 43. Tarea de ingeniería 2.2.

Tarea de ingeniería	
Numero Tarea: 2.3	Numero Historia: 2
Nombre Tarea: Buscar usuario en el sistema.	
Tipo de Tarea: Diseño, Desarrollo	Puntos estimados: 0.50
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para buscar un usuario en la base de datos.	

Tabla 44. Tarea de ingeniería 2.3.

Tarea de ingeniería	
Numero Tarea: 3.1	Numero Historia: 3
Nombre Tarea: Insertar paciente en el sistema.	
Tipo de Tarea: Diseño, Desarrollo	Puntos estimados: 0.25
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para insertar un paciente y se insertarán los datos en la base de datos.	

Tabla 45. Tarea de ingeniería 3.1.

Tarea de ingeniería

Numero Tarea: 3.2	Numero Historia: 3
Nombre Tarea: Eliminar paciente del sistema.	
Tipo de Tarea: Diseño, Desarrollo	Puntos estimados: 0.25
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para eliminar un paciente y es eliminado en la base de datos.	

Tabla 46. Tarea de ingeniería 3.2.

Tarea de ingeniería	
Numero Tarea: 3.4	Numero Historia: 3
Nombre Tarea: Buscar paciente en el sistema.	
Tipo de Tarea: Diseño, Desarrollo	Puntos estimados: 0.50
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para buscar un paciente en la base de datos.	

Tabla 47. Tarea de ingeniería 3.4.

Tarea de ingeniería	
Numero Tarea: 4.1	Numero Historia: 4
Nombre Tarea: Insertar evaluación en el sistema.	
Tipo de Tarea: Diseño, Desarrollo	Puntos estimados: 0.25
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para insertar una evaluación y se insertarán los datos en la base de datos.	

Tabla 48. Tarea de ingeniería 4.1.

Tarea de ingeniería	
Numero Tarea: 4.2	Numero Historia: 4
Nombre Tarea: Eliminar evaluación del sistema.	
Tipo de Tarea: Diseño, Desarrollo	Puntos estimados: 0.25
Responsable: José Luis Simón Peña.	

Descripción: Se diseñará una interfaz para eliminar una evaluación y es eliminada en la base de datos.

Tabla 49. Tarea de ingeniería 5.1.

Tarea de ingeniería	
Numero Tarea: 4.3	Numero Historia: 4
Nombre Tarea: Buscar evaluación en el sistema.	
Tipo de Tarea: Diseño, Desarrollo	Puntos estimados: 0.50
Responsable: José Luis Simón Peña.	
Descripción: Se diseñará una interfaz para buscar una evaluación en la base de datos.	

Tabla 50. Tarea de ingeniería 4.3.

ANEXO 4. Pruebas de aceptación iteración 2

Código: HU1-AU-I2	Historia de usuario: 1 - Autenticar usuario.		
Escenario: autenticar.			
Condiciones de ejecución: ninguna.			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Autenticación correcta del usuario en el sistema (usuario y clave).	Tras autenticarse, si el proceso es satisfactorio, Se muestran las funcionalidades en dependencia del rol al que pertenezca el usuario.	Satisfactorio.	
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Entrada incorrecta del usuario o la contraseña.	El sistema debe mostrar un mensaje de error informando: "Compruebe sus datos".	Satisfactorio.	
Campos requeridos vacíos (usuario,	El sistema debe mostrar un mensaje de error informando:	Satisfactorio.	

contraseña).	“Compruebe sus datos”.		
Evaluación de la Prueba: satisfactoria.			

Tabla 51: Prueba de aceptación autenticar usuario.

Código: HU2-IU-I2	Historia de usuario: 2 - Gestionar usuario.		
Escenario: insertar usuario.			
Condiciones de ejecución:			
<ul style="list-style-type: none"> El usuario administrador tiene que estar autenticado en el sistema. 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Entrada correcta del usuario (nombre, apellidos, CI, cargo, usuario, clave, rol).	El sistema debe registrar el nuevo usuario y mostrar un mensaje informado: “Se ha insertado correctamente”.	Satisfactorio.	
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Campos requeridos vacíos (nombre, apellidos, CI, cargo, usuario, clave, rol).	El sistema debe mostrar un mensaje de error informando: “Campos vacíos”.	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 52. Prueba de aceptación - insertar usuario.

Código: HU2-EU-I2	Historia de usuario: 2 - Gestionar usuario.		
Escenario: eliminar usuario.			
Condiciones de ejecución:			
<ul style="list-style-type: none"> El usuario administrador tiene que estar autenticado en el sistema. 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Seleccionar de la lista	El sistema debe eliminar el	Satisfactorio.	

el usuario que desea eliminar y eliminarlo.	usuario y mostrar un mensaje informado: "Usuario eliminado".		
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Eliminar un usuario sin seleccionarlo de la lista.	El sistema debe mostrar un mensaje de error informando: "Seleccione el paciente".	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 53. Prueba de aceptación – eliminar usuario.

Código: HU2-BU-I2	Historia de usuario: 2 - Gestionar usuario.		
Escenario: buscar usuario.			
Condiciones de ejecución:			
<ul style="list-style-type: none"> El usuario administrador tiene que estar autenticado en el sistema. 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Entrada correcta del identificador (CI) del usuario que desea buscar.	El sistema debe mostrar los datos del usuario encontrado.	Satisfactorio.	
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Entrada incorrecta del identificador (CI) del usuario que desea buscar.	El sistema debe mostrar un mensaje error informando: "Datos incorrectos".	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 54. Prueba de aceptación - buscar usuario.

Código: HU3-IHC-I2	Historia de usuario: 3 - Gestionar paciente		
Escenario: insertar paciente.			
Condiciones de ejecución:			
<ul style="list-style-type: none"> El usuario tiene que estar autenticado en el sistema. 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Entrada correcta del paciente (nombre, apellidos, CI, edad, dirección, CI del doctor, sexo).	El sistema debe registrar el nuevo paciente y mostrar un mensaje informado "Se ha insertado correctamente".	Satisfactorio.	
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Campos requeridos vacíos (nombre, apellidos, CI, edad, dirección, CI del doctor, sexo).	El sistema debe mostrar un mensaje de error informando: "Campos vacíos".	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 55. Prueba de aceptación – insertar paciente.

Código: HU3-EHC-I2	Historia de usuario: 3 - Gestionar paciente.		
Escenario: eliminar paciente			
Condiciones de ejecución:			
<ul style="list-style-type: none"> El usuario tiene que estar autenticado en el sistema. 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Seleccionar de la lista el paciente que desea eliminar y eliminarlo.	El sistema debe eliminar el paciente y mostrar un mensaje informado "Paciente	Satisfactorio.	

	eliminado”.		
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Eliminar un paciente sin seleccionarlo de la lista.	El sistema debe mostrar un mensaje de error informando: “Selecione el paciente”.	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 56. Prueba de aceptación - eliminar paciente.

Código: HU3-BHC-I2	Historia de usuario: 3 - Gestionar paciente		
Escenario: buscar paciente.			
Condiciones de ejecución			
<ul style="list-style-type: none"> El usuario tiene que estar autenticado en el sistema. 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Entrada correcta del identificador (CI) del paciente que desea buscar.	El sistema debe mostrar los datos del paciente buscado.	Satisfactorio.	
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Entrada incorrecta del identificador (CI) del paciente que desea buscar.	El sistema debe mostrar un mensaje error informando: “Datos incorrectos”.	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 57. Prueba de aceptación – buscar paciente.

Código: HU4-IE-I2	Historia de usuario: 4 - Gestionar evaluación.		
Escenario: insertar evaluación.			

Condiciones de ejecución:			
<ul style="list-style-type: none"> El doctor tiene que estar autenticado en el sistema. 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Seleccionar de la lista el paciente que desea evaluar y registra la evaluación.	El sistema debe registrar la nueva evaluación y mostrar un mensaje informado “Se ha insertado correctamente”.	Satisfactorio.	
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Evaluar un paciente sin seleccionarlo de la lista.	El sistema debe mostrar un mensaje de error, informando “Selecione el paciente”.	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 58. Prueba de aceptación – inserta evaluación.

Código: HU4-EE-I2	Historia de usuario: 4 - Gestionar evaluación.		
Escenario: eliminar evaluación.			
Condiciones de ejecución:			
<ul style="list-style-type: none"> El usuario doctor tiene que estar autenticado en el sistema. 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Seleccionar de la lista la evaluación que desea eliminar y eliminarla.	El sistema debe eliminar la evaluación y mostrar un mensaje informado: “Evaluación eliminada”.	Satisfactorio.	

Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Eliminar una evaluación sin seleccionarla de la lista.	El sistema debe mostrar un mensaje de error informando: "Selecione la evaluación".	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 59. Prueba de aceptación – eliminar evaluación.

Código: HU4-BE-I2	Historia de usuario: 4 - Gestionar evaluación.		
Escenario: buscar evaluación.			
Condiciones de ejecución:			
<ul style="list-style-type: none"> El usuario doctor tiene que estar autenticado en el sistema. 			
Clases válidas	Resultados esperado	Resultados de la prueba	Observaciones
Entrada correcta del identificador de la evaluación (ID evaluación) que desea buscar.	El sistema debe mostrar los datos de la evaluación encontrada.	Satisfactorio.	
Clases inválidas	Resultados esperado	Resultados de la prueba	Observaciones
Entrada incorrecta del identificador de la evaluación (ID evaluación) que desea buscar.	El sistema debe mostrar un mensaje error informando: "Datos incorrectos".	Satisfactorio.	
Evaluación de la Prueba: satisfactoria.			

Tabla 60. Prueba de aceptación – buscar evaluación.

GLOSARIO DE TÉRMINOS

Biomecánica: la biomecánica es una disciplina científica que tiene por objeto el estudio de las estructuras de carácter mecánico que existen en los seres vivos (fundamentalmente del cuerpo humano).

Estabilométrico: la postura de los pacientes y su evolución en el tiempo.

Framework: en el desarrollo de *software*, un *framework* es una estructura de soporte definida en la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, un *framework* puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros *software* para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

HMD: (*Head Mounted Display*) Dispositivo montado en la cabeza. Como su nombre lo indica este es un dispositivo muy parecido a un casco de realidad virtual, pero el usuario en vez de mirar un entorno totalmente generado por la computadora, observa una parte del mundo real con elementos virtuales insertados en la escena que está visualizando.

OPENGL (Open Graphics Library)(Biblioteca de Gráficos Abierta): OpenGL es una especificación estándar que define una API multi-lenguaje y multi-plataforma para escribir aplicaciones que producen gráficos 3D, desarrollada originalmente por *Silicon Graphics Incorporated* (SGI).

Osteomuscular: relacionado con músculos, los huesos, los tendones, los ligamentos, las articulaciones y los cartílagos.

Propioceptivas: informan sobre la situación del cuerpo en el espacio, la postura del cuerpo, los movimientos, el equilibrio, la actitud entre otros.

stbTracker: biblioteca de la visión por computadora para la detección y representan la estimación de los marcadores de referencia en 2D.