 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

PROPUESTA DE ESTÁNDAR DE DOCUMENTACIÓN PARA SOFTWARE DE CARÁCTER CIENTÍFICO

Jessyca Leamsy Restrepo Arroyave

Sandra Patricia Guerra Ortega

Ingeniería de sistemas

Juan Sebastián Botero Valencia

INSTITUTO TECNOLÓGICO METROPOLITANO

2018


 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

El desarrollo de software busca crear aplicaciones apropiadas que den solución a los requerimientos de los sistemas actuales. Para que estos desarrollos estén dentro los acuerdos y solicitudes pactados con el cliente o usuario, es recomendable el uso de la ingeniería, esto permite planificar cada una de las fases, asegurando el cumplimiento de los objetivos sin superar el tiempo y el costo proyectado. Esto parece un estándar y una disciplina fundamental, pues se ha establecido como una parte integral en el campo científico y empresarial, además de ser el escenario propicio para definir el ciclo de vida del software y de facilitar que todo el grupo de desarrollo comprenda de manera clara el orden de los procesos y la funcionalidad de estos.

Este proyecto propone un estándar para la documentación inexistente de los modelos y procedimientos implementados en las aplicaciones desarrolladas por los grupos de investigación de los diferentes laboratorios del Parque i del Instituto Tecnológico Metropolitano (ITM) y que les es solicitada por el Centro de Emprendimiento y Transferencia de Tecnología (CETT) del ITM para su registro como producto de software ante la Dirección Nacional de Derechos de Autor (DNDA), a través de una convocatoria permanente, la cual les exige algunos requisitos para poder hacer parte de ésta. La inscripción es necesaria para participar en la medición de grupos de investigación del Departamento Administrativo de Ciencia, Tecnología e Innovación–Colciencias. Por lo cual, esta documentación se ajusta y respeta las etapas que Colciencias solicita a los proyectos de investigación que generen un producto de software.

Palabras clave: Ingeniería de software, ingeniería inversa, metodología, estándar, software científico.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22


RECONOCIMIENTOS

En este trabajo de grado realizado en el Instituto Tecnológico Metropolitano, queremos agradecer a todas las personas que directa o indirectamente participaron día a día en nuestro desarrollo, principalmente en la orientación de nuestras metas, motivándonos siempre a tener claro el futuro que deseábamos al culminar nuestra carrera, apuntando a un objetivo claro; también a las personas que nos ayudaron en los momentos difíciles y en los llevaderos, los que nos acompañaron con su paciencia, los que siempre estuvieron con sus consejos empujándonos a salir a delante y nunca desfallecer y superar todas las dificultades que la vida nos presenta.

Queremos agradecer a Dios por permitirnos hacer realidad este sueño anhelado, y al Profesor Juan Sebastián Botero por su acompañamiento en este proyecto; por su experiencia, conocimiento, paciencia, constancia y entrega para formarnos en el ser y el saber, por su orientación, porque más que los conocimientos afianzados en ésta nuestra última actividad académica, nos quedan talentos y habilidades para la toma de decisiones personales y laborales.


También agradecemos a los profesores que nos encontramos a lo largo de nuestra formación profesional, todos con su granito de arena aportaron a nuestro crecimiento, por sus consejos, enseñanzas, su visión crítica de muchos aspectos cotidianos de la vida y sobre todo por su rectitud en la profesión como docentes.

Un gracias a todos nuestros compañeros donde se encuentren, por escucharnos, por apoyarnos, por trabajar de nuestra mano, por creer en nosotros, por todos los momentos buenos y por mejorar, por todos los momentos que se presentaron en estos periodos académicos, gracias por las risas que nos dieron, por sus consejos, por enseñarnos a querer

	<p style="text-align: center;">INFORME FINAL DE TRABAJO DE GRADO</p>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

al otro desde la diferencia, y a entender que la vida no se trata de cambiar a las personas sino de aceptarlas tal cual como son.

Todo esto no hubiera sido posible sin el amparo incondicional y el cariño que nos entregaron nuestras familias, que de forma incondicional entendieron nuestras ausencias y malos momentos, gracias por las oportunidades brindadas, por su apoyo económico y emocional, por enseñarnos a ponernos de pie luego de caer.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACRÓNIMOS

CETT Centro de Emprendimiento y Transferencia de Tecnología

DNDA Dirección Nacional de Derechos de Autor

ITM Instituto Tecnológico Metropolitano

wIMUa Wireless Inertial Measurement Unit Acquisition

wIMUc Wireless Inertial Measurement Unit Calibration

IMU Unidad de medición inercial

UML Unified Modeling Language (Lenguaje unificado de modelado)

IDE Integrated Development Environment (Entorno de desarrollo integrado)

PC Computador personal

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE CONTENIDO

1. INTRODUCCIÓN	8
GENERALIDADES.....	8
OBJETIVOS	11
General.....	11
Específicos	11
2. MARCO TEÓRICO	12
METODOLOGÍAS DE INGENIERÍA DEL SOFTWARE	13
METODOLOGÍA EN CASCADA.....	13
MÉTODO DE PROTOTIPOS	15
MODELO INCREMENTAL O ITERATIVO Y CRECIENTE	17
MODELO EN ESPIRAL.....	18
RAD: DESARROLLO RÁPIDO DE APLICACIONES (RAPID APPLICATION DEVELOPMENT)	20
INGENIERÍA INVERSA	22
3. METODOLOGÍA.....	23
ORGANIZACIÓN DE LA TESIS.....	23
4. RESULTADOS Y DISCUSIÓN	25
PREANÁLISIS BÁSICO	26
INTEGRANTES CON ROLES	26
FUNCIONALIDADES DEL SOFTWARE.....	26
ENTREVISTAS	26
UBICACIÓN GENERAL.....	26
ALCANCE DEL SISTEMA.....	27
OBJETIVO DEL SISTEMA	27
FUNCIONALIDADES Y OBJETIVOS	27
DEFINICIÓN DEL SISTEMA.....	28
NIVEL INTERIOR	28
NIVEL EXTERIOR	28

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

NIVEL DE INFORMACIÓN DE ENTRADA AL SISTEMA	28
A NIVEL DE INFORMACIÓN GENERADA POR EL SISTEMA TENEMOS.....	28
ELEMENTOS FUNCIONALES DEL SISTEMA.....	28
ELEMENTOS ADICIONALES DEL SISTEMA	28
DIAGRAMA DE CONTEXTO DEL SISTEMA	29
CRONOGRAMA DE ACTIVIDADES.....	29
ANÁLISIS	31
PROBLEMA A SOLUCIONAR.....	31
DIAGRAMA MODELO DE PROCESO.....	32
MODELO CONCEPTUAL.....	32
NARRATIVA WIMUC	32
DIAGRAMA DE CONCEPTOS.....	33
REQUERIMIENTOS DEL SISTEMA	33
DIAGRAMA DE CLASES	34
DIAGRAMA DE CASOS DE USO DE ALTO NIVEL	34
CASO DE USO: Realizar Calibración	35
CASO DE USO REAL: REALIZAR CALIBRACIÓN	37
CONTRATOS	37
DISEÑO DE PANTALLAS	43
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO	44
REFERENCIAS	46
APÉNDICE	47
Apéndice A: Anexo 1 - Descripción del estándar del manual de usuario	47
Apéndice B: Anexo 2 - Manual de usuario wIMUc	47
Apéndice C: Anexo 3 - Descripción del estándar de Ingeniería de Software	47
Apéndice D: Anexo 4 - Documentación de software de carácter científico.....	47
Apéndice E: Anexo 5 - Ingeniería inversa	48

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. INTRODUCCIÓN

GENERALIDADES

La ingeniería del software es un conjunto de métodos y técnicas utilizados con el fin de garantizar la calidad del software final. El surgimiento e implementación de esta disciplina se dio a conocer después de una conferencia de la OTAN en 1968, luego de identificar los grandes obstáculos presentes en el desarrollo de aplicaciones, que comúnmente no ofrecían a los usuarios lo que requerían y eran más costosos de lo esperado (Wirth, 2008).

En la actualidad, los programadores ya no trabajan en solitario, sino en conjunto con un equipo de especialistas en software, donde cada uno se enfoca en una parte, lo que garantiza poder desarrollar una aplicación más óptima y compleja, dado que el campo de la ingeniería del software tiene como objetivo, desarrollar una base de pruebas para la comprensión científica y la intervención en los procesos implicados en el desarrollo de tecnologías de software, con el fin de apoyar los métodos de la ingeniería y la tecnología empleada en su planificación, diseño, construcción, implementación, validación, procesos de software de mantenimiento y mejora (de Almeida Biolchini, Mian, Natali, Conte, & Travassos, 2007). Hoy es indiscutible que este método es un requisito clave para el éxito del desarrollo de software. Sin embargo, en muchos proyectos de computación científica, las prácticas tradicionales de ingeniería se ignoran. Es importante comprender que es necesario implementar esta metodología, porque ayuda a capturar y gestionar los requisitos para el desarrollo de los proyectos.

El software científico propone un enfoque que se basa en un meta-modelo con el fin de hacer frente a la alta complejidad y el cambio frecuente en su desarrollo. El enfoque es compatible con la ingeniería de requisitos en estos proyectos, proporcionando flexibilidad

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

para su administración, control, rastreo y comunicación a través del límite de disciplinas. Además, ha demostrado sus ventajas en la aplicación y promueve la ingeniería de software (Li et al., 2012).

El “Parque i” del Instituto Tecnológico Metropolitano – ITM es un espacio creado para fomentar desarrollos tecnológicos y de investigación; en éste se comparten conocimientos y recursos de varias instituciones universitarias del municipio de Medellín (“Alcaldía de Medellín”, 2016). El Parque i está compuesto por varios laboratorios en los cuales interactúan los investigadores que se especializan en diferentes áreas. Estos grupos implementan y utilizan distintas aplicaciones que facilitan o mejoran los procesos; dichas aplicaciones son creaciones del personal administrativo, investigadores o docentes y estudiantes en coautoría con alguno de ellos. Las aplicaciones se entregan al Centro de Emprendimiento y Transferencia de Tecnología (CETT) del ITM para su registro como producto de software ante la Dirección Nacional de Derechos de Autor (DNDA), a través de una convocatoria permanente, la cual les exige algunos requisitos para poder hacer parte de ésta, por ejemplo, los poderes para ceder al ITM el derecho a ser representados en dicha entidad, la evaluación del software por parte un par externo a la facultad, un manual de usuario y una descripción del software. Esta inscripción es necesaria para participar en la medición de grupos de investigación del Departamento Administrativo de Ciencia, Tecnología e Innovación – Colciencias, la cual tiene como objetivo general “Fortalecer el Sistema Nacional de Ciencia y Tecnología y a Colciencias para lograr un modelo productivo sustentado en la ciencia, la tecnología y la innovación, para darle valor agregado a los productos y servicios de nuestra economía, y propiciar el desarrollo productivo y una nueva industria nacional.” (Departamento Administrativo de Ciencia, Tecnología e Innovación - Colciencias, 2015).

El desarrollo de software científico o de investigación posee características muy diferentes del desarrollo de software comercial. Normalmente el software científico se realiza en

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ambientes de investigación, donde los mismos integrantes del equipo están interesados en los productos y en general son los usuarios finales de las aplicaciones. Por esta misma razón los requerimientos son definidos por el equipo que desarrolla el proyecto.

Las aplicaciones que se entregan al CETT por parte de los investigadores del Parque i, (como ocurre normalmente) fueron desarrolladas por los integrantes de dichos grupos y aunque cumplen con lo requerido, no cuentan con la documentación, estudio y análisis previo para un buen desarrollo, porque este departamento no exige la ingeniería de software y no tienen un estándar para el manual de usuario que se solicita como requisito para el registro de este, además es común que los integrantes de los grupos de investigación no tengan conocimiento previo sobre esta disciplina y buenas prácticas de documentación, por ende realizan la entrega de dicha documentación muchas veces de manera empírica, sin seguir lineamientos y sin incluir características y detalles importantes, lo que genera grandes dificultades para identificar oportunidades de mejora continua, que permitan hacerlas más optimas, útiles y eficientes, crea una barrera para la comprensión entre los desarrolladores y los demás usuarios, además, resulta difícil idear modificaciones porque en general solo se documenta en el mismo código, pero sin llevar un estándar y la calidad de ésta depende únicamente de quien la creó. Sumado a esto, al no contar con un buen manual de usuario final que incluya todas las características del programa (Gonzalez Reyes, 2016), se experimentará una degradación constante en el uso correcto de sus herramientas y no se aprovecharían tanto como se ideó en el momento de su concepción.

Es claro que los programadores poseen una buena disposición para incorporar mejoras en el proceso de desarrollo de software siempre que sean fáciles y rápidas de usar, pero no manejan las metodologías de la Ingeniería de Software y tampoco tienen la experiencia para implementar sus técnicas (Espina & Neumann, 2012).

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22


OBJETIVOS

General

Desarrollar una propuesta de estándar de documentación para software de carácter científico, que debe ser implementado por los miembros de los grupos de investigación que deseen registrar sus creaciones de software ante la Dirección Nacional de Derechos de Autor (DNDA) a través del Centro de Emprendimiento y Transferencia de Tecnología (CETT) del ITM, con el fin de participar en la medición de grupos de Colciencias.

Específicos

- Verificar el estado actual de las metodologías de ingeniería del software e ingeniería inversa que se utilizan para diseñar los modelos que definen el software y seleccionar las más adecuadas
- Realizar ingeniería inversa al software tomado como caso de estudio, para obtener la documentación que actualmente no se tiene sobre éste
- Definir y ajustar el estándar apropiado para la documentación de ingeniería y del manual de usuario que es solicitado a los investigadores de la institución para el registro de los productos de software
- Implementar un modelo de documentación referente al estándar de ingeniería de software y el manual de usuario que se ajuste a los requerimientos de la Institución y Colciencias

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. MARCO TEÓRICO

El presente trabajo muestra la importancia del análisis previo de cualquier proyecto de software, para ello daremos a conocer algunos conceptos importantes con el fin de comenzar a comprender de qué se trata esta disciplina.

El concepto principal es el de **Software**; los softwares son conjuntos de instrucciones que cuando se ejecutan proporcionan las características y las funciones buscadas. A su vez, la ingeniería de software es la implementación de principios fundamentales con el objetivo de desarrollar de manera económica un proyecto de software.

La ingeniería de software es una disciplina que se desarrolla por capas, es decir, no comienzan el desarrollo de una etapa, sin antes terminar la anterior. Las capas que la conforman son: Proceso, métodos y herramientas. El proceso de ingeniería define la estructura para la administración de los proyectos de software. Los métodos de ingeniería proporcionan experiencia técnica para elaborar software, en esta se describen las actividades y técnicas implementadas. Finalmente, las herramientas de ingeniería proporcionan apoyo normalizado para los procesos y los métodos (Presman & Troja, 1988).

Distinto al concepto anterior, la ingeniería inversa es el proceso de analizar un software, con el fin de construir las definiciones y documentación de este partiendo del código fuente, “es un proceso de recuperación del diseño”. Sobre nuestro caso de estudio, se debe tener en cuenta que la aplicación no cuenta con una interfaz gráfica, que se muestre a base de imágenes o gráficos que interactúen con los usuarios finales, por lo tanto, no es preciso realizar un análisis de esta. Usualmente esta disciplina se aplica a los productos o programas existentes y se hace cuando es requerida como requisito obligatorio por entidades o instituciones donde desee mostrarse; en este caso, Colciencias lo exige para participar en sus concursos. Generalmente las compañías utilizan esta metodología para identificar cómo funcionan los productos o software de la competencia, sin

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

embargo, esta vez es aplicado a un software desarrollado por un grupo de investigadores del Parque i del ITM. La ingeniería inversa debe permitir obtener toda la información requerida solo con su código fuente y la base de datos. Tal y como sucede con la parte grafica inexistente, tampoco se cuenta con una base de datos que permita almacenar, procesar, modificar o eliminar la información generada por la aplicación, es decir, no se obtendría modelo entidad relación y la visualización de sus datos se hará a través de la ventana de comandos de Matlab (Tinetti, F. Li et al., 2008).

Actualmente las metodologías de ingeniería de software que se utilizan para diseñar los modelos que definen el software son las siguientes:

METODOLOGÍAS DE INGENIERÍA DEL SOFTWARE


Las Metodologías de desarrollo de software se enfocan en el entorno en el cual se plantea y estructura un sistema. Existe una gran cantidad de metodologías las cuales se utilizan dependiendo del proyecto que se esté desarrollando, pues no todos son compatibles con las mismas, dado que el ciclo de vida puede ser variable.

En las metodologías de desarrollo de software se usan diversas herramientas, métodos y modelos para el desarrollo. La principal característica de estas metodologías es la documentación, con el fin de que los demás miembros del grupo de desarrollo comprendan el ciclo de vida del software.

Actualmente existen varias metodologías, aunque todas están basadas en características similares, no siguen las mismas etapas. Son las que veremos a continuación:

METODOLOGÍA EN CASCADA

Este modelo es el más antiguo. Su estilo consiste en que no podrás avanzar a la siguiente fase, si la anterior no se encuentra totalmente terminada, dado que no hay vuelta atrás. Las fases de dicho modelo son las siguientes (Presman & Troja, 1988):

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. **Análisis de Requisitos.** Aquí se definen los requerimientos del sistema y los objetivos que el software debe cumplir al terminar el desarrollo. Como se indicó anteriormente, una vez avanzado el paso de análisis, no puede haber vuelta atrás.

2. **Diseño del Sistema.** En esta etapa se elaborará la estructura del sistema y se describirán cada una de sus partes.

3. **Diseño del Programa.** Se realizan los diagramas y algoritmos, teniendo en cuenta que no necesariamente es código.

4. **Codificación.** En esta etapa se desarrolla el código. El tiempo requerido para elaborar esta etapa dependerá del lenguaje que se vaya a utilizar. Algunos permiten utilizar componentes, bibliotecas y algunas funciones para reutilizar código.

5. **Ejecución de Pruebas.** Se utiliza para verificar si el sistema realmente funciona, antes de sacarlos a producción.

6. **Verificación.** Si la realización de la fase anterior se superó con éxito, se procede a la realización de pruebas por parte del usuario final.

7. **Mantenimiento.** Las aplicaciones actuales se están actualizando constantemente. Esta es una de las fases más lentas del modelo, dado que se debe estar dar solución a los comentarios de los usuarios.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

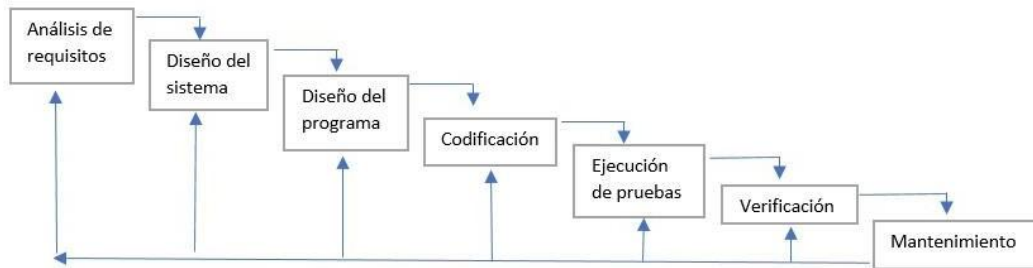


Figura 1: Metodología en cascada


MÉTODO DE PROTOTIPOS

El método de prototipos es uno de los métodos favoritos, por su rapidez en la elaboración de un prototipo sencillo en base al análisis de requerimientos y necesidades, y que permitirá a cualquier programador ir trabajando en él a medida que llega hasta el código final del mismo.

Tal como su nombre lo indica este método se basa en una versión no terminada del producto final, y es usada como plantilla para los clientes y usuarios similares, con el único fin de permitir una ventaja en el desarrollo y adaptar las funciones personalizadas de acuerdo con las necesidades del proyecto (Cataldi, Z. Li et al., 1999).

Las etapas de desarrollo de software de la metodología de prototipos son las siguientes:

1. **Planeación:** A diferencia de otras metodologías esta etapa es muy rápida ya que no se requiere tanta planeación para hacer el prototipo.
2. **Modelado:** Debe ser muy rápida para que no retrase el proyecto. Lo ideal es que se haga un solo prototipo.
3. **Elaboración del Prototipo:** Ahora se debe elaborar el prototipo. Dado que esta es una etapa importante, debe hacerse con calma y sin rapidez.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

4. **Desarrollo:** Se debe comenzar el desarrollo. El tiempo estimado de la tardanza depende del tamaño del proyecto y el lenguaje de programación utilizado.

5. **Entrega y Retroalimentación:** Se realiza la entrega del proyecto y se le da al cliente una retroalimentación de cómo utilizarlo

6. **Comunicación con el Cliente:** El contacto con el cliente es crucial para el proyecto tenga las especificaciones deseadas y los ajustes que se indiquen en el desarrollo de este, con el fin de que garantice si el proyecto es correcto. Esta es la principal diferencia de la metodología cascada, donde luego de terminar una etapa es imposible regresar a la anterior.

7. **Entrega del Producto Final:** Esta metodología tiene como ventaja de que el código es reutilizable y puede ser mejorado si se desea, para que así con el prototipo ya puedas simplemente empezar de nuevo y con una buena base de código que te acelerará el proceso.

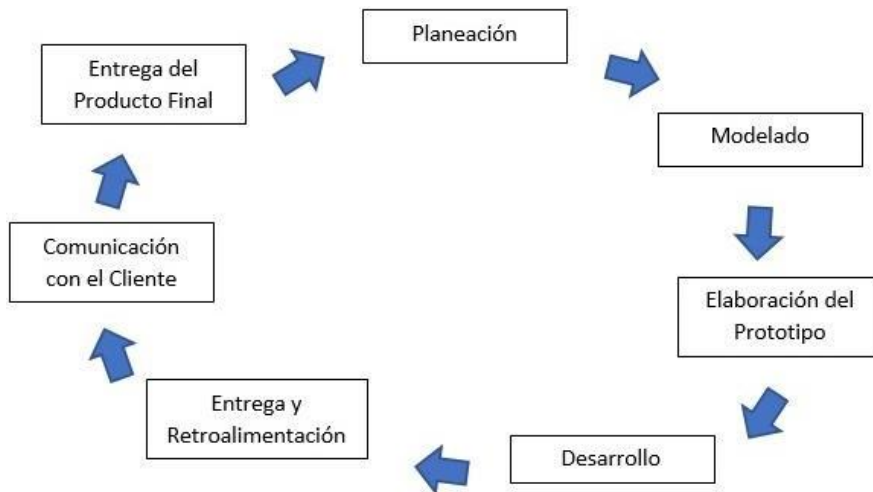


Figura 2: Método de prototipos

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

MODELO INCREMENTAL O ITERATIVO Y CRECIENTE

Este modelo permite que obtener un producto final más completo. Este es la combinación de los modelos lineal e iterativo o bien, modelo de cascada y prototipos. Consiste en realizar el modelo de cascada en varias iteraciones, pero sin completar alguna, esto permite agregar nuevas especificaciones, funcionalidades y opciones que el usuario necesite. Esto quiere decir que el modelo incremental, es el mismo modelo cascada realizado varias veces (Presman & Troja, 1988).

Las fases son las siguientes:

1. **Inicialización:** En esta etapa se define la idea a desarrollar, se definen los requisitos, aunque no es necesario listarlos y definirlos, dado que este método se basa en iteraciones que permitan su avance.
2. **Periodos de Iteración:** Da el inicio a las iteraciones (las cuales son ilimitadas). Luego de la primera iteración se obtiene un segundo prototipo del proyecto, que puede volverse a inicializar para mejorarlo o modificarlo.
3. **Lista de Control:** Como las iteraciones son ilimitadas se debe llevar un control de cada una de ellas identificadas con una versión, con los ajustes de las mismas en una lista. Como si fuera un programa que recibe actualizaciones constantemente.

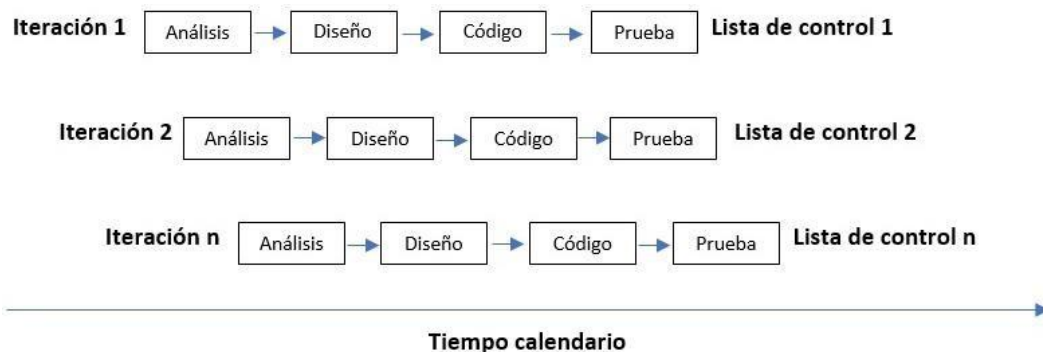


Figura 3: Método incremental o iterativo y creciente

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

MODELO EN ESPIRAL


Este modelo fue creado por Barry Boehm en 1986. Es una combinación entre el modelo de cascada y el modelo iterativo, en el que se incluye la gestión de riesgos, una etapa que no es implementada en los demás modelos

Tal como su nombre lo dice, se desarrolla con fases que se van realizando en modo de espiral, utilizando procesos como lo hace el modelo de cascada, pero acá no son obligatorios y no llevan un orden. Teóricamente podemos decir que es un modelo evolutivo, que irá incrementando, dependiendo del nivel de código fuente desarrollado (Presman & Troja, 1988).

Esta metodología es utilizada en el desarrollo de grandes proyectos como lo son la creación de un sistema operativo.

Las fases del modelo de espiral son:

1. **Determinar Objetivo:** Esta etapa solo debe realizarse una vez, aunque se podrán determinar objetivos durante cada iteración. Esto es posible pues poco a poco se irá incrementando más el tamaño del manual de usuario, los requisitos, las especificaciones y las restricciones.
2. **Análisis de Riesgo:** Una etapa donde incluso una lluvia de ideas podría ayudar, el análisis de riesgos. Aquí deberás tener en cuenta todo aquello que pueda dañar a tu proyecto, ya sea que se trate de ciertas amenazas o de posibles daños que se puedan ocasionar, teniendo además un Plan B por así decirlo, para que en caso de que ocurra algo inesperado, tener a la mano la solución para continuar con el proyecto. En esta fase del modelo espiral, podemos agregar lo que son la creación de prototipos, pues siempre es bueno tener un respaldo de nuestro código, se esta forma en caso de que algo malo suceda, volvemos a la versión anterior. Así que cada

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

vez que vayamos a ingresar a la fase de pruebas e implementación, será necesario contar con un prototipo que nos respalde.

3. **Desarrollar, Validar y Probar:** Con esta fase es con la que se irá desarrollando el proyecto, depende principalmente del análisis de riesgos, dado que el proyecto se enfocará en los riesgos que puedan ser evitados, por ejemplo, si se identifica que la interfaz del usuario debe ajustarse, entonces el prototipo se enfocará en este, lo que no significa que se dejará de lado el resto del proyecto, pero si enfocará un poco más en lo que priorice en ese momento. El modelo de espiral irá avanzando y repitiendo el ciclo hasta que los riesgos sean mínimos o nulos.

4. **Planificación:** No es posible avanzar a esta etapa, si no se resuelven los riesgos de la etapa anterior, lo que significa que sirve para determinar el avance del proyecto y para indicar este hacia donde se dirige.

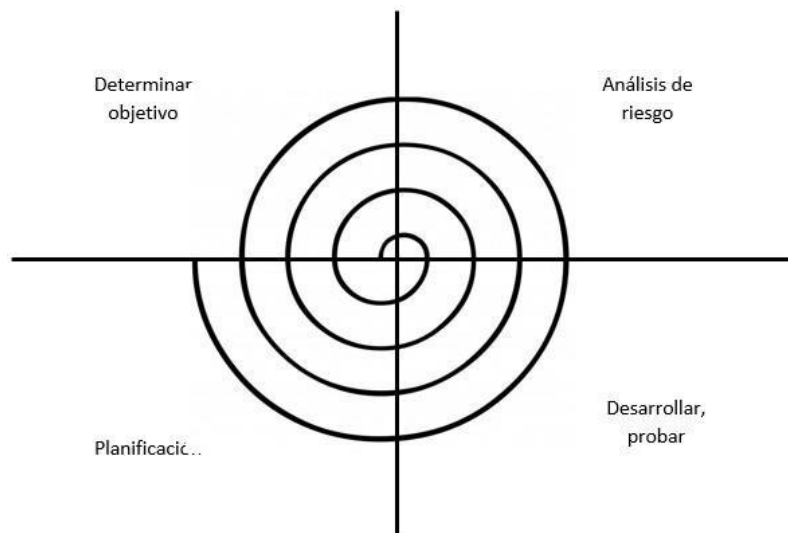


Figura 4: Modelo en espiral

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RAD: DESARROLLO RÁPIDO DE APLICACIONES (RAPID APPLICATION DEVELOPMENT)

La principal diferencia de esta metodología es que no cuenta con una serie de fases o etapas, aunque si se basa en modelo de cascada y la creación de prototipos. Así que vamos a ver los principios del modelo RAD.


La elección del modelo utilizado depende de muchos factores, el principal es el tamaño del proyecto, por ejemplo, el modelo espiral se utiliza en proyectos grandes y el modelo RAD, se enfoca en proyectos pequeño ("Metodología RAD", 2018).

Actualmente Colciencias solicita como requerimiento un modelo en el que se incluyen las siguientes etapas:

1. **Análisis:** (Descripción máxima en 500 palabras). Proceso en el cual se definen los requerimientos del sistema mediante la precisión de sus funciones, su comportamiento, grado de rendimiento, la arquitectura a utilizar y la integración con otros sistemas. (Descripción clara de que producto se va a construir, qué funcionalidades aportará y que comportamiento tendrá).

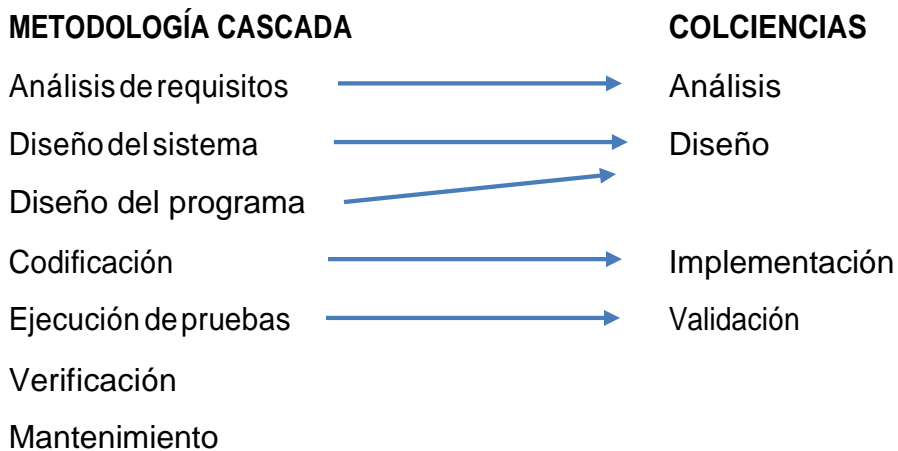
2. **Diseño:** (Descripción máxima en 500 palabras). Proceso en el cual se realiza la definición y descripción del modelo de información, los módulos que conforman la arquitectura, las características de la interfaz del usuario y el detalle procedimental (algoritmos) del software, de acuerdo con las especificaciones definidas en el análisis.

3. **Implementación:** (Descripción máxima en 500 palabras). Proceso en el cual se realiza la traducción del diseño del código fuente y las pruebas para la detección de errores en el código desarrollado.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

4. Validación: (Descripción máxima en 500 palabras). Proceso en el cual se realizan las pruebas para la comprobación del cumplimiento de los requisitos y la aceptación por parte del usuario final.

Luego de analizar las metodologías indicadas en la documentación anterior, se determina que la más adecuada es “Cascada”, pues ambas tienen un desarrollo lineal, en el que se ejecutan una a una las etapas, y siempre dependen de la anterior. Si consideramos que la metodología de Colciencias involucra menos etapas o algunas tienen variaciones en el nombre (Departamento Administrativo de Ciencia, Tecnología e Innovación - Colciencias, 2015)., Se documentarán de la siguiente manera:



Aunque Colciencias no requiere una etapa de Verificación o pruebas de usuario, para este proyecto es importante mostrar su funcionalidad, ya que los usuarios finales son los mismos investigadores encargados de su desarrollo.


Por último, la etapa de mantenimiento no será tomada en cuenta, porque el Software tomado como caso de estudio no requiere ajustes, pues cumple su tarea como parte del proyecto de investigación realizado, en el que como ya se había indicado, participa otra aplicación.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

INGENIERÍA INVERSA

La ingeniería inversa no cuenta con metodologías definidas como sucede con la ingeniería del software, la bibliografía que actualmente se encuentra en la red o en las revistas científicas solo la referencian, pero no profundizan en ella, aunque si se tiene especificado 3 tipos de ingeniería, estos son los siguientes (Presman & Troja, 1988):

1. **Ingeniería inversa para comprender datos:** Permite la obtención de modelos relacionales o el diagrama entidad-relación, a partir de su codificación.
2. **Ingeniería inversa para entender el procesamiento:** Se realiza sobre el código de un programa para identificar como fue creada su lógica o sobre cualquier documento de diseño para obtener documentos de análisis o de requisitos.
3. **Ingeniería inversa de interfaces de usuario:** Es utilizada para obtener los modelos que sirvieron para la construcción de esta, su objetivo es utilizarlas como punto de partida en procesos de ingeniería que permitan modificar dicha interfaz.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3. METODOLOGÍA

Este trabajo de grado está planeado para desarrollar un estándar de documentación basados en la ingeniería del software para aplicaciones de carácter científico, donde debemos considerar toda la información suministrada por parte del docente Juan Sebastián Botero, quien nos guio con conceptos básicos y facilitando el software científico ya desarrollado por parte de grupo de investigación del “Parque i” del Instituto Tecnológico Metropolitano.

Con la información recolectada sobre el software Wireless Inertial Measurement Unit Calibration - wIMUc (en español, Calibración de una unidad de medición inercial inalámbrica) se obtuvo la información requerida y se realizaron los análisis necesarios para concluir que aplica teniendo en cuenta el tipo de software.

ORGANIZACIÓN DE LA TESIS

Para el desarrollo de cada uno de estos objetivos, se entregará toda la información obtenida y construida a lo largo de este proyecto. Esta se puede visualizar en el área “Resultados y discusión”. Inicialmente abordaremos el tema de las metodologías actuales para la ingeniería de software e ingeniería inversa, en vista de que para la segunda no se tienen establecidas una serie de etapas como si pasa con la primera, se darán a conocer algunos conceptos exhibidos por diversos investigadores en sus artículos científicos, se indicará con cual metodología son relacionados; para luego seleccionar las más adecuadas.

Posteriormente entregaremos la documentación recogida y las conclusiones que pudieron sacarse en la realización de la ingeniería inversa, teniendo en cuenta que en este proceso no se realizará ingeniería inversa de datos, porque la aplicación no tiene base de datos en una

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

aplicación como SQL, es decir, no se obtendría modelo entidad relación; esta aplicación maneja los datos y el cálculo de estos en Matlab.

A continuación, se mostrará el manual de usuario creado, que permitirá conocer a fondo la aplicación tomada como caso estudio, desde su instalación hasta la manera de utilizarla. Consideramos que, para una comprensión más clara del estándar de ingeniería de software propuesto, se debe desarrollar cada una de las etapas, implementando algún modelo de los mencionados en la verificación de las metodologías actuales. Por ello, también se entrega la documentación recogida de la ingeniería de software realizada al software indicado.

Finalmente, se describe de manera detallada cada elemento o sección que compone tanto al manual de usuario como al estándar de ingeniería inversa utilizado, teniendo en cuenta que la metodología utilizada debía ajustarse a los requerimientos de Colciencias, fin último de los proyectos de desarrollo de software de los grupos de investigación del ITM.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

4. RESULTADOS Y DISCUSIÓN

El documento “**Anexo 1 - Descripción del estándar del manual de usuario**” se entregade manera adicional a este formato (FDE 089). En este se listan todas las secciones indispensables para la construcción de un manual de usuario final, se indica que debe ir en cada una de ellas y se propone un formato para estos.

El “**Anexo 2 - Manual de usuario wIMUc**”, al igual que el mencionado anteriormente, se entrega de manera adicional a este formato (FDE 089). Este contiene el manual de usuario final del software tomado como caso de estudio, Wireless Inertial Measurement Unit Calibration (wIMUc). Allí se indican las especificaciones técnicas a nivel de hardware y software que facilitarán la ejecución del programa, también incluyen secciones clave como los acrónimos y el glosario, esto con el fin de que cualquier usuario comprenda completamente qué hace el programa, cómo lo hace y por qué lo hace.

El archivo “**Anexo 3 - Descripción del estándar de Ingeniería de Software**” se entregade manera adicional a este formato (FDE 089). En este se listan todas las etapas indispensables para la construcción de la documentación de ingeniería de software indispensable antes de la ejecución de un proyecto. Se describe que debe ir en cada una de ellas y se propone un formato para esta.

Presentamos a continuación la documentación obtenida luego de implementar el estándar propuesto, al software tomado como caso de estudio (esta se entrega de manera adicional en el “**Anexo 4 - Documentación de software de carácter científico**”).

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

PREANÁLISIS BÁSICO

Para el proyecto de investigación “Desarrollo de un sistema de medición Inercial para obtener y registrar variables biomecánicas en deportistas” no existen los softwares necesarios para la recolección de datos necesarios para su ejecución. Para esta necesidad se piensa desarrollar dos softwares: "Wireless Inertial Measurement Unit Acquisition - wIMUa-", el cual realiza la adquisición de un sistema de medición inercial para obtener y registrar variables biomecánicas en deportistas. Y “Wireless Inertial Measurement Unit Calibration - wIMUc -” que realiza la calibración del magnetómetro en una Unidad de Medición inercial. En este análisis nos enfocaremos en el estudio del segundo, wIMUc.

INTEGRANTES CON ROLES

Juan Botero (Investigador)

Luis Montes (Investigador)

David Márquez (Investigador)

FUNCIONALIDADES DEL SOFTWARE

Calibrar el magnetómetro en una Unidad de Medición inercial.

ENTREVISTAS

ROL (Investigador): Para el proyecto de investigación es de suma importancia contar con las aplicaciones necesarias. La recolección de datos el tratamiento que se haga con estos nos ayudara con el desarrollo de este. De manera preliminar se tiene desarrollado el software wIMUa, pero este genera los planos desalineados, por lo cual se requiere la creación de otro software que permita la calibración de dichos planos para que queden en paralelo.

UBICACIÓN GENERAL

El sistema que se va a desarrollarse ubicará en el laboratorio de “” del Parque i del Instituto Tecnológico Metropolitano (ITM), espacio creado para fomentar los proyectos tecnológicos y de investigación.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ALCANCE DEL SISTEMA

El fin del software wIMUc sería calibrar el magnetómetro en una Unidad de Medición inercial (IMU, del inglés inertial measurement unit), el cual es un dispositivo electrónico que mide e informa acerca de la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una base de calibración robotizada de cuatro grados de libertad. Inicialmente se utilizarán los datos obtenidos con el software wIMUa y que son almacenados en una base de datos en Excel, estos datos son: la aceleración en X, Y y Z, los Valores del giroscopio de X, Y y Z (rotación en grados por segundo), Gravedad (en grados por segundos), campo magnético, la temperatura del sensor, y la variable de tiempo en milisegundos. Este archivo sigue el modelo Denavit Hartenberg (DH, permite establecer la ubicación de los sistemas de referencia de los eslabones en los sistemas robóticos articulados).

OBJETIVO DEL SISTEMA

Alinear los planos generados con los datos de un magnetómetro en una Unidad de Medición inercial, para que queden en paralelo.

FUNCIONALIDADES Y OBJETIVOS

Tabla 1: Funcionalidades y Objetivos

FUNCIONALIDAD	OBJETIVO
El wIMUc se encarga de hallar los centros de los planos generados y luego los alinea para que queden en paralelo.	Alinear los planos generados con los datos de un magnetómetro en una Unidad de Medición inercial, para que queden en paralelo.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

DEFINICIÓN DEL SISTEMA

NIVEL INTERIOR

El software wIMUc interactúa con los investigadores Juan Botero, Luis Montes y David Márquez del laboratorio de “” del Parque i del ITM, para el proyecto de investigación “Desarrollo de un sistema de medición inercial para obtener y registrar variables biomecánicas en deportistas” identificado con No 1102-626-38784, en la Convocatoria No. 626 “Deporte, Recreación y Actividad Física” de 2013 - Colciencias.

NIVEL EXTERIOR

Las personas que interactúan en el laboratorio de “” del Parque i del ITM, podrían tener acceso a este software, aunque este no tiene influencia en el desarrollo de sus actividades.

NIVEL DE INFORMACIÓN DE ENTRADA AL SISTEMA

Inicialmente se utilizarán los datos obtenidos con el software wIMUa, que son almacenados en una base de datos en Excel cuyo nombre sería Prueba.xls.

A NIVEL DE INFORMACIÓN GENERADA POR EL SISTEMA TENEMOS

La alineación de los centroides de los planos +Z y -Z, definen la calibración realizada por el wIMUc, la cual es mostrada a través de una vista.

ELEMENTOS FUNCIONALES DEL SISTEMA

Los investigadores del laboratorio de “” del Parque i del ITM, encargados de este proyecto de investigación serán los encargados de obtener la información requerida para la construcción de los elipsoides, a los cuales luego se le calcula los centroides que se requiere alinear.

ELEMENTOS ADICIONALES DEL SISTEMA

El software wIMUc no realiza funciones adicionales a la calibración.

DIAGRAMA DE CONTEXTO DEL SISTEMA

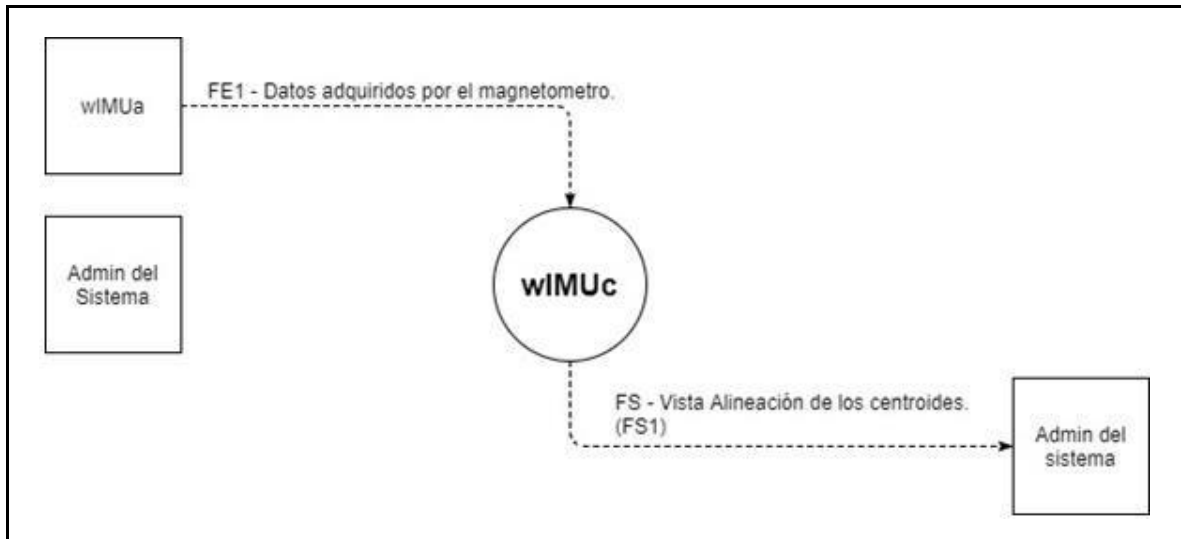


Figura 5: Diagrama de contexto del sistema

CRONOGRAMA DE ACTIVIDADES

Tabla 2: Cronograma de actividades

FECHA DE EJECUCIÓN	HORAS DESTINADAS	OBJETIVO	ACTIVIDAD	RESULTADO
2018/02/20 2018/02/21	4	Definir objetivos	Establecer objetivos a alcanzar	Indicar objetivos en documento de propuesta
2018/02/22 2018/02/23	4	Verificación literaria sobre ingeniería de software	Consultar en internet, revistas científicas y repositorios, información sobre proyectos similares, además identificar implicaciones	Almacenar artículos que más adelante serán utilizados como base bibliográfica
2018/03/05 2018/03/07	5	Verificación literaria sobre ingeniería inversa	Consultar en internet, revistas científicas y repositorios, información sobre esta disciplina	Almacenar artículos que más adelante serán utilizados como base bibliográfica
2018/03/12 2018/03/14	5	Verificación metodologías de creación de	Consultar en internet y repositorios, manuales de	Almacenar manuales que más adelante serán

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

		manuales de usuario final	usuario de aplicaciones similares	utilizados como base bibliográfica
2018/03/19 al 2018/03/23	10	Ingeniería inversa de lógica o de proceso	Realizar la ingeniería inversa al código del programa para examinar su lógica y hacer el levantamiento de los requerimientos	Documentación de requerimientos
			Revisar y clasificar los requisitos en funcionales, no funcionales y reglas de negocio.	Documentación de análisis de la solución
2018/03/26 al 2018/03/30	10	Realizar Ingeniería de Software	Crear la documentación de ingeniería de software	Documentación de ingeniería de software
2018/04/02 al 2018/04/05	8	Crear un manual de usuario final estándar	Construcción del manual de usuario final, donde se describe el paso a paso de los procesos a realizar por el usuario	Manual del aplicativo
2018/04/11	2	Socialización de la propuesta con los grupos de investigación del parque i	Socializar, discutir y ajustar con la participación de los investigadores del parque-I, en particular con aquellos que tiene conocimiento de las técnicas y metodologías de la ingeniería del software	Documentación final
	16			Informe final

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2018/04/16 2018/04/26		Redacción del informe final	Construcción de reporte final, donde se incluyen los resultados obtenidos. Adicionalmente se almacenará esta información en el repositorio indicado	
--------------------------	--	-----------------------------	---	--

ANÁLISIS

PROBLEMA A SOLUCIONAR

Para el proyecto de investigación “Desarrollo de un sistema de medición Inercial para obtener y registrar variables biomecánicas en deportistas” se desea crear un software que permita la posibilidad de alinear los centroides de los elipsoides generados con los datos obtenidos del magnetómetro, esto definiría la calibración requerida.



Figura 6: Actores

wIMUa: Software que permite la adquisición de datos de un sistema de medición inercial para obtener y registrar variables biomecánicas en deportistas.

Investigador: Rol encargado de la creación, administración, manipulación y mantenimiento de los softwares necesarios para llevar a cabo el proyecto de investigación.

wIMUc (Sistema): Software que realiza la calibración del magnetómetro en una Unidad de Medición Inercial.

DIAGRAMA MODELO DE PROCESO

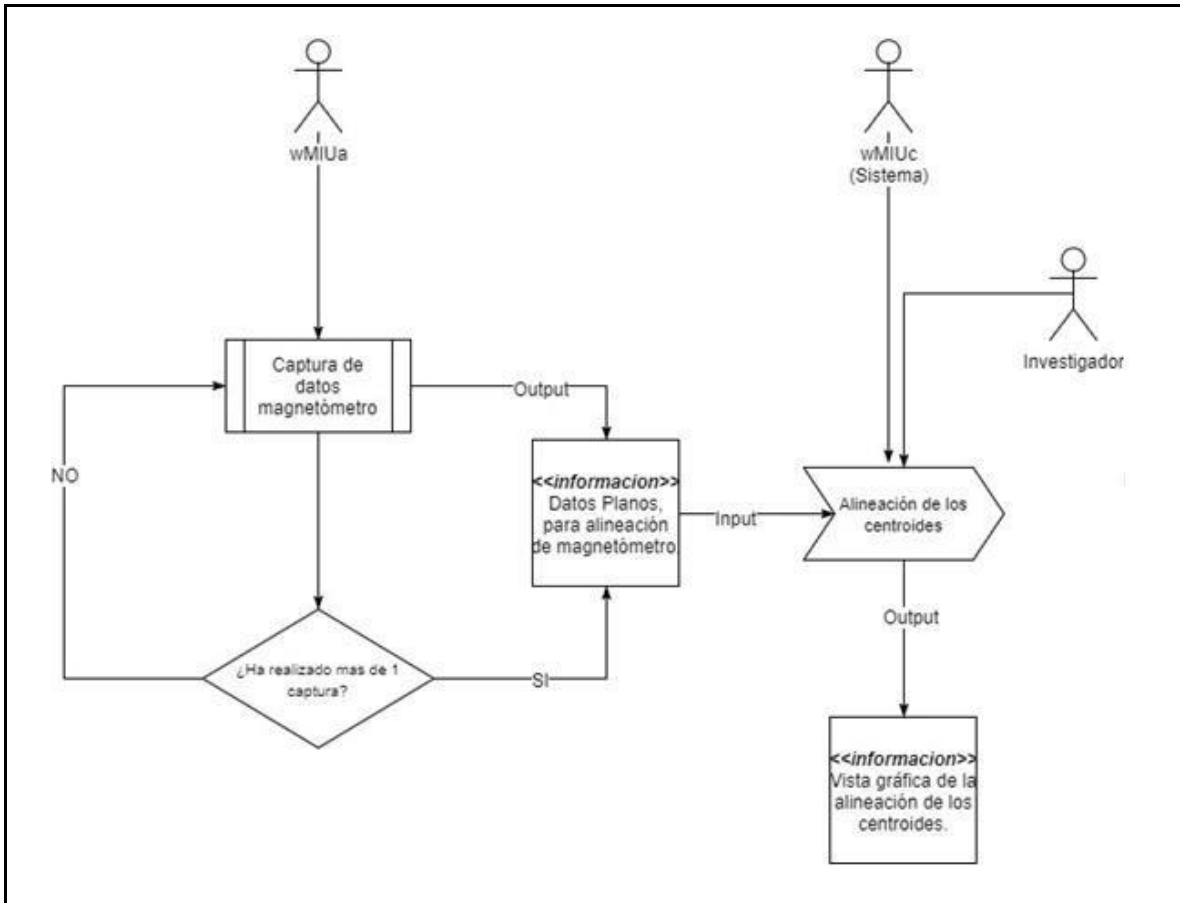



Figura 7: Diagrama modelo de proceso

MODELO CONCEPTUAL

NARRATIVA WIMUC

Se debe crear un software que brinde la posibilidad de calibrar un magnetómetro en una Unidad de Medición Inercial.

Con este sistema interactuarán los investigadores del laboratorio de "" del Parque i del ITM, pues está enfocado explícitamente al trabajo de proyectos con Unidades de Medición Inercial en escenarios de estudio científico. Se podrá tener acceso a este software en este mismo espacio y no deberá reproducirse su contenido sin consentimiento de los investigadores y la institución, dado que este tipo de proyectos se entregan al Centro de Emprendimiento y Transferencia de Tecnología

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

(CETT) del ITM y se ceden sus derechos sobre estos, para ser representados ante la Dirección Nacional de Derechos de Autor (DNDA) para su registro, con el fin de participar de las convocatorias permanentes, Esta inscripción es necesaria para participar en la medición de grupos de investigación del Departamento Administrativo de Ciencia, Tecnología e Innovación – Colciencias.

El usuario deberá ejecutar la aplicación mediante el archivo wIMCU_main, del directorio wIMUc_Matlab, teniendo previamente instalada la aplicación Matlab®. También se debe tener en cuenta que de manera inicial se debe ejecutar el software wIMUa para realizar el almacenamiento de datos en el archivo Prueba.xls. Para cada prueba a realizar se debe generar un nuevo archivo.

DIAGRAMA DE CONCEPTOS

Este software no cuenta con una base de datos en la que se puedan incluir las clases del sistema, sus atributos, los métodos y las relaciones entre los objetos, solo posee una tabla donde almacena sus únicos datos. Por este motivo el diagrama de concepto es representado con una única tabla, la cual por obvias razones no posee relación conceptual con otras y sus datos. Este diagrama **No Aplica** en este proyecto.

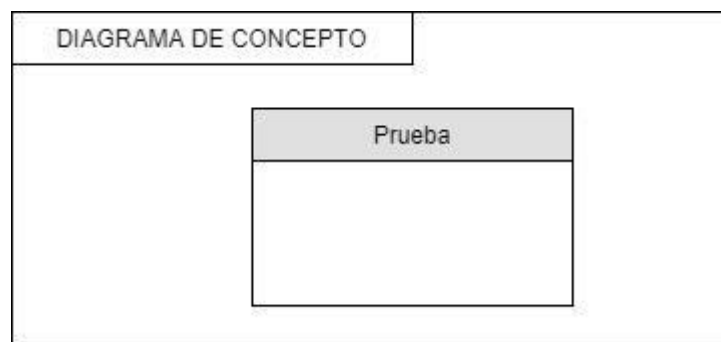



Figura 8: Diagrama de conceptos

REQUERIMIENTOS DEL SISTEMA

Tabla 3: Requerimientos del sistema

REFERENCIA	FUNCIÓN	CATEGORIA	PARTICIPANTES
R1	Ingresar a la aplicación	Evidente	Investigador

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

R2	Ejecutar la aplicación	Evidente	Investigador
R3	Realizar el proceso de calibración	Oculto	Sistema
R4	Generar vista con la calibración realizada	Oculto	Sistema

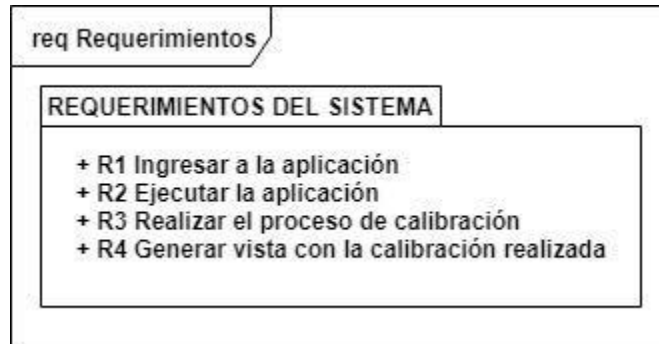


Figura 9: Requerimientos del sistema

DIAGRAMA DE CLASES

Este software no cuenta con una base de datos en la que se puedan incluir las clases del sistema, sus atributos, los métodos y las relaciones entre los objetos. Teniendo en cuenta lo anterior, este diagrama **No Aplica** en este proyecto.

DIAGRAMA DE CASOS DE USO DE ALTO NIVEL



Figura 10: Diagrama de casos de uso de alto nivel

CASO DE USO: Realizar Calibración

DIAGRAMA DE CASO DE USO: REALIZAR CALIBRACIÓN

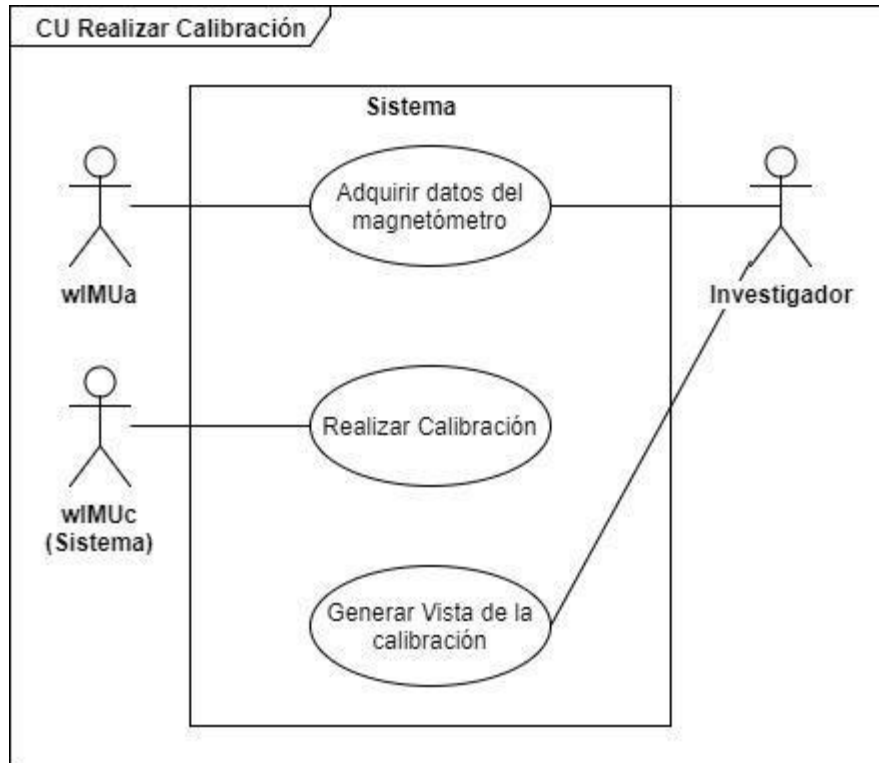



Figura 11: Diagrama de casos de uso: Realizar calibración

DESCRIPCIÓN DE CASO DE USO: REALIZAR CALIBRACIÓN

Tabla 4: Descripción de caso de uso: Realizar calibración

ACTOR PRINCIPAL: wIMUc (Sistema)
ACTOR SECUNDARIO: wIMUa, Investigador
PERSONAL INVOLUCRADO E INTERESES: <ul style="list-style-type: none"> El Investigador necesita que wIMUc (Sistema) realice la calibración de los elipsoides creados con los datos del magnetómetro.
Requerimientos: R3: Realizar proceso de calibración. R4: Generar vista con la calibración realizada.
Precondiciones: Los Investigadores deben haber ingresado a la aplicación.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Postcondiciones: Se debentener los datos para los +Z y -Z para poder realizar la calibración.
Escenario principal de éxito (Flujo Básico) :
1. El wIMUa genera los datos del plano o elipsoide +Z
2. El Investigador gira el magnetómetro con el brazo del robot y esto genera los datos del plano o elipsoide -Z
5. El wIMUc (Sistema) calcula el centroide de los planos, la alineación de estos centroides define la calibración realizada.
6. El wIMUc (Sistema) genera una vista de la calibración realizada.
Frecuencia: Cada vez que realice una prueba o se ejecute la calibración con alguno de los archivos Prueba.xls ya existentes.

DIAGRAMA DE SECUENCIA: REALIZAR CALIBRACIÓN

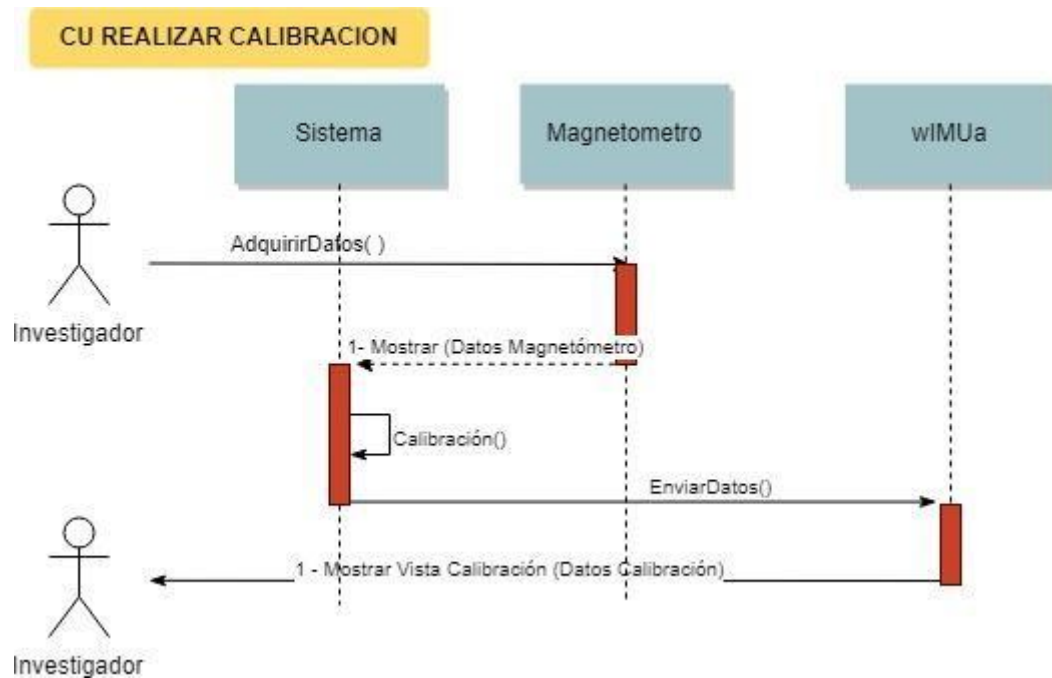


Figura 12: Diagrama de secuencia: Realizar calibración

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

DIAGRAMA DE COLABORACIÓN

Este software no cuenta con una interfaz principal que proporcione algún manejo del proceso, el sistema requiere procesos físicos basados en hardware, por este caso los diagramas de colaboración **No Aplica**, ya que no es posible graficar los niveles de programación basada en capas, ya que el software solo tiene una.

CASO DE USO REAL: REALIZAR CALIBRACIÓN


Tabla 5: Caso de uso real: Realizar Calibración

Caso de uso real: Realizar calibración.
Actor principal: wIMUc
Precondiciones: Los Investigadores deben haber ingresado a la aplicación.
Postcondiciones: Se deben tener los datos para los planos +Z y -Z para poder realizar la calibración.
Formulario: El software utilizado no utiliza una interfaz, por ende sus resultados se muestran en la consola de Matlab, programa sobre el cual se ejecuta el wIMUc . Teniendo en cuenta lo anterior, esta opción No Aplica en este diseño.
Flujo Básico: <ol style="list-style-type: none"> 1. El wIMUa genera los datos del plano o elipsoide +Z 2. El Investigador gira el magnetómetro con el brazo del robot y esto genera los datos del plano o elipsoide -Z 3. El wIMUc (Sistema) calcula el centroide de los planos, la alineación de estos centroides define la calibración realizada. 4. El wIMUc (Sistema) genera una vista de la calibración realizada.
Flujo Alternativo: El sistema no tiene flujos alternos en su proceso.

CONTRATOS

Tabla 6: Contratos

Nombre	Contrato: REALIZAR CALIBRACIÓN
---------------	---------------------------------------

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Responsabilidades	Proceso que permite la calibración del dispositivo con un magnetómetro que permite la realización de la calibración.
Tipo	Contrato de Concepto.
Referencias cruzadas	R1. Ingresar a la aplicación. R2. Ejecutar la aplicación R3. Realizar el proceso de calibración. R4. Generar vista con la calibración realizada.
Personal involucrado e intereses:	Investigador y Personal a cargo del mismo.
Notas:	<pre> %----- data = xlsread('Prueba2_IMU2.xls'); %----- ini1=1; fin1=1300; %----- ini2=5201; fin2=6500; %----- A=data(1:1300,7:9); %Puntos como vectores columna B = data(5201:6500,7:9); %Puntos como vectores columna %----- Am = zeros(3,64); Bm = zeros(3,64); for i=1:64 l1 = 20*i; l2 = l1+20; Am(1,i) = mean(A(1,l1:l2)); Am(2,i) = mean(A(2,l1:l2)); Am(3,i) = mean(A(3,l1:l2)); </pre>

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

Bm(1,i) = mean(B(1,1:l2));
Bm(2,i) = mean(B(2,1:l2));
Bm(3,i) = mean(B(3,1:l2));

end

%-----
C1 = mean(Am,2);           %Calcula la media del plano +Z
C2 = mean(Bm,2);          %Calcula la media del plano -Z
EJ=lin3(C1,C2,3);         %Calcula la recta paramétrica EJE
%-----

Bm = fliplr(Bm);
Bmr = tral(C2)*rotzm(pi+0.4)*tral(-C2)*[Bm;ones(1,64)];
Bm = Bmr(1:3,:);
%-----

figure
pbaspect([2 2 2])
daspect([1 1 1])
hold on
global ColorOrder, ColorOrder='rgb';
plot3(Am(1,:),Am(2,:),Am(3,:),'*r','linewidth',4) %Dibuja puntos plano +Z
plot3(Am(1,1),Am(2,1),Am(3,1),'*y','linewidth',8) %Dibuja punto inicial +Z
plot3(Bm(1,:),Bm(2,:),Bm(3,:),'*k','linewidth',4) %Dibuja puntos plano -Z
plot3(Bm(1,1),Bm(2,1),Bm(3,1),'*m','linewidth',8) %Dibuja punto inicial -Z
for i=1:64
    plot3([Am(1,i) Bm(1,i)],[Am(2,i) Bm(2,i)],[Am(3,i) Bm(3,i)],'b','linewidth',2)
end
%-----

Am(1,:)= 1.1926*Am(1,:);
Am(2,:)= 1.1258*Am(2,:);
Bm(1,:)= 1.1926*Bm(1,:);
Bm(2,:)= 1.1258*Bm(2,:);
%-----

```

```

[mIn posIn] = min(abs(Am(1,:)-C1(1)));
%-----
C1 = mean(Am,2);           %Calcula la media del plano +Z
C2 = mean(Bm,2);           %Calcula la media del plano -Z
EJ=lin3(C1,C2,3);          %Calcula la recta paramétrica EJE
%-----
NAr = tral(-EJ(:,2))*[Am;ones(1,64)];
NBr = tral(-EJ(:,2))*[Bm;ones(1,64)];
NA = NAr(1:3,:);
NB=NBr(1:3,:);
%-----
NC1 = mean(NA,2);          %Calcula la media del plano+Z
NC2 = mean(NB,2);          %Calcula la media del plano -Z
EJ=lin3(NC1,NC2,3);        %Calcula la recta paramétrica EJE
%-----
Anx = asin(NC1(2)/norm(NC1,2));
Any = asin(NC1(1)/norm(NC1,2));
%-----
NAr = rotxm(Anx)*[NA;ones(1,64)];
NBr = rotxm(Anx)*[NB;ones(1,64)];
NA = NAr(1:3,:);
NB=NBr(1:3,:);
%-----
% NAr = rotym(-Any)*[NA;ones(1,64)];
% NBr = rotym(-Any)*[NB;ones(1,64)];
% NA = NAr(1:3,:);
% NB = NBr(1:3,:);
%-----
NC1 = mean(NA,2);          %Calcula la media del plano+Z
NC2 = mean(NB,2);          %Calcula la media del plano -Z
EJ=lin3(NC1,NC2,3);        %Calcula la recta paramétrica EJE

```



```

%-----
if (posIn+21)>64
  Vd1 = cross(NA(:,posIn-43)-NA(:,posIn),NA(:,posIn-21)-NA(:,posIn));
else
  Vd1 = cross(NA(:,posIn+21)-NA(:,posIn),NA(:,posIn-21)-NA(:,posIn));
end
AnInM1x = acos(dot(Vd1,[0 1 0])/(norm(Vd1,2)*norm([0 1 0],2)))-pi/2;
AnInM1y = acos(dot(Vd1,[1 0 0])/(norm(Vd1,2)*norm([1 0 0],2)))-pi/2;
%-----
NAr = tral(NC1)*rotxm(-AnInM1x)*tral(-NC1)*[NA;ones(1,64)];
NBr = tral(NC2)*rotxm(-AnInM1x)*tral(-NC2)*[NB;ones(1,64)];
NA = NAr(1:3,:);
NB=NBr(1:3,:);

% NAr = tral(C1)*rotym(-AnInM1y)*tral(-C1)*[NA;ones(1,64)];
% NBr = tral(C2)*rotym(-AnInM1y)*tral(-C2)*[NB;ones(1,64)];
% NA = NAr(1:3,:);
% NB = NBr(1:3,:);
%-----

% NA=tral(-EJ(:,1))*[NA;ones(1,64)];   %Nuevo plano +Z trasladado
% NB=tral(-EJ(:,3))*[NB;ones(1,64)];   %Nuevo plano -Z trasladado
% NE = tral(-EJ(:,2))*[EJ;ones(1,3)];   %Nuevo EJE trasladado
%-----

% NC1 = mean(NA,2);   %Nuevo centro plano +Z
% NC2 = mean(NB,2);   %Nuevo centro plano -Z
%-----

% Anx = asin(NC1(2)/norm(NC1,2));
% Any = asin(NC1(1)/norm(NC1,2));

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

%-----
plot3(NA(1,:),NA(2,:),NA(3,:),'*r','linewidth',4) %Dibuja puntos plano +Z
trasladado
plot3(NB(1,:),NB(2,:),NB(3,:),'*k','linewidth',4) %Dibuja puntos plano +Z
trasladado
% plot3(NA(1,posIn),NA(2,posIn),NA(3,posIn),'*y','linewidth',8)
% plot3(NA(1,posIn+21),NA(2,posIn+21),NA(3,posIn+21),'*y','linewidth',8)
% plot3(NA(1,posIn-21),NA(2,posIn-21),NA(3,posIn-21),'*y','linewidth',8)
plot3(EJ(1,:),EJ(2,:),EJ(3,:),'m','linewidth',5) %Gráfica el EJE original
%-----
% h1 = arrow3([0 0 0],NC1(1:3),'y',1,3);
% set(h1,'LineWidth',1.5);
h2 = arrow3(zeros(3),diag([400,400,400]),'o',3,9);
set(h2,'LineWidth',3);
text(420,0,0,'X','FontSize',24);
text(0,420,0,'Y','FontSize',24);
text(0,0,420,'Z','VerticalAlignment','bottom','HorizontalAlignment','center','FontSize',24)
axis([-850 850 -850 850 -850 850])
grid, axis square, view(3), title('Magnetometer'), camlight

poM = pos2radMX12W(data(ini1:fin1,15))
mxNN = data(ini1:fin1,7)-mean(data(ini1:fin1,7));
myNN = data(ini1:fin1,8)-mean(data(ini1:fin1,8));
anNN = atan2(myNN,mxNN);

figure
hold on
plot(NA(1,:),NA(2,:))
axis square

```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

	<pre> %plot(NA(2,:)) % figure % hold on % plot(mxNN) % plot(myNN) % % % figure % hold on % plot(poM) % plot(anNN) </pre>
Excepciones:	Es posible que el proceso de calibración falle si no se ha calibrado el magnetómetro.
Salida:	El software grafica las mediciones del magnetómetro como de la calibración.
Precondiciones:	El magnetometromide el campo magnético de la tierra; Al girar el brazo del robot cambian los datos que hay en el magnetómetro (los obtenidos con wIMUa), el valor del campo magnético debería cambiar.
Postcondiciones	El wIMUc toma el centroide de cada plano, la alineación de los centroides de estos planos define la calibración realizada.

DISEÑO DE PANTALLAS

El software utilizado no utiliza una interfaz, por ende, sus resultados se muestran en la consola de Matlab, programa sobre el cual se ejecuta el wIMUc. Teniendo en cuenta lo anterior, este diseño **No Aplica** en este proyecto.


El documento nombrado como “**Anexo 5 - Ingeniería inversa**”, explica dicha metodología, indica en qué consisten los tipos de ingeniería inversa existentes y puntualiza sobre que aplica al software tomado como caso de estudio.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

Después de elaborar el proyecto y de haber realizado un adecuado análisis de la información obtenida, se ponen en consideración algunas conclusiones y recomendaciones.

- Se elabora la propuesta del estándar para la documentación de software de carácter científico, teniendo en cuenta las etapas sugeridas en la metodología seleccionada, la cual cumple con los requisitos exigidos por Colciencias. Se confirma que, debido al tipo de software desarrollado, no aplican todos los diseños correspondientes a la etapa de diseño, por lo cual son omitidos. Adicionalmente, se construye e implementa un estándar para la construcción de un manual de usuario final, que incluye datos relevantes del software desarrollado, su funcionamiento y términos clave para que cualquier usuario con pueda comprenderlo claramente.
- En la actualidad no se cuenta con suficiente documentación relevante que sirva como base y ejemplo fundamental en el desarrollo de la ingeniería de software de carácter científico.
- El desarrollo de la documentación del software científico utilizado como caso de estudio, permite conocer el funcionamiento de este, identificar sus componentes, determinar su alcance y comprender su importancia dentro del proyecto de investigación. Se comprueba que, según cada tipo de software, aplican o no algunos diagramas (representación gráfica que simplifica la definición de algún proceso determinado).

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Quizá el tipo de software más fácil de documentar es el científico, puesto que normalmente estos cumplen un solo objetivo (o tienen una única tarea) y no requieren de grandes desarrollos para cumplir su propósito.
- Para desarrollos futuros, se recomienda la realización de la ingeniería de software en el orden sugerido por esta metodología, sin omitir ninguna etapa y en los tiempos previamente establecidos. La ingeniería inversa se utiliza comúnmente para determinar cómo funciona un software y cuales posibles mejoras puedan ser implementadas, por ello, no es muy útil si solo desea hacerse para generar la documentación inexistente de un software ya desarrollado, más aún si se tiene en cuenta que normalmente, podría realizarse solo al código fuente de la aplicación pues de manera habitual, no cuentan con una base de datos para almacenar datos que luego puedan ser modificados y más aún, casi nunca cuentan con una interfaz gráfica.
- Desde el momento en que se pensó en el desarrollo de una disciplina que permitiera explicar de manera detallada cada una de las etapas por las cuales debía pasar un proyecto de software, se entendió que siempre debía ser utilizada para que dicho proyecto se desarrollara dentro del tiempo y el presupuesto estimados, por ello, es imperiosa su implementación. Teniendo en cuenta que los investigadores que desarrollan sus aplicaciones en los laboratorios del Parque i del ITM, normalmente no utilizan esta metodología, bien sea porque no ven su importancia o porque no tienen los conocimientos propios de esta, se recomienda la divulgación del estándar desarrollado en este proyecto, tanto para la ingeniería de software, como para el manual de usuario, esto servirá para que cualquier persona que acceda a esta información pueda conocer claramente todo lo concerniente a la construcción de la aplicación y su funcionamiento. También se tendría la uniformidad que se propone establecer.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

Alcaldía de Medellín. (2016). Alcaldía de Medellín. Retrieved 9 July 2016, from <https://www.medellin.gov.co/irj/portal/nmedellin?NavigationTarget=navurl://68def41588169162b563a45cb4bf5dee>

Cataldi, Z., Lage, F., Pessacq, R., & García Martínez, R. (1999, August). Ingeniería de software educativo. In Proceedings del V Congreso Internacional de Ingeniería Informática (pp. 185-199).

de Almeida Biolchini, J., Mian, P., Natali, A., Conte, T., & Travassos, G. (2007). Scientific research ontology to support systematic review in software engineering. *Advanced Engineering Informatics*, 21(2), 133-151.

Departamento Administrativo de Ciencia, Tecnología e Innovación - Colciencias,. (2015). Modelo de medición de grupos de investigación, Desarrollo Tecnológico o de Innovación y de reconocimiento de investigadores del sistema nacional de ciencia, Tecnología e Innovación. Bogota.

Espina, H. A., & Neumann, G. C. Desarrollo de Software de Investigación.


GonzalezReyes, A. (2016). Instructivo para la elaboración de un Manual de Usuario (1st ed., p. 7). La Habana. Retrieved from http://cujae.edu.cu/index.php?option=com_k2&Itemid=941&id=76_4cc93c8ac11adcb3093d87bafa85384f&lang=es&task=download&view=item

Metodologia RAD. (2018). Metodologia RAD. Retrieved 1 May 2018, from <http://metodologiarad.weebly.com/>

Pressman, R. S., & Troya, J. M. (1988). Ingeniería del software.

Tinetti, F. G., Cajaraville, P. G., Labraga, J. C., López, M. A., & Olguín, M. G. (2008). Ingeniería inversa aplicada a software numérico: modelos climáticos. In X Workshop de Investigadores en Ciencias de la Computación.

Wirth, N. (2008). A brief history of software engineering. *IEEE Annals of the History of Computing*, 1(3), 32-39.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

APÉNDICE

Se presentan a continuación los anexos que contienen los resultados obtenidos luego del desarrollo del proyecto “Propuesta de estándar de documentación para software de carácter científico”.

Apéndice A: Anexo 1 - Descripción del estándar del manual de usuario

En el documento se listan todas las secciones indispensables para la construcción de un manual de usuario final, se indica que debe ir en cada una de ellas y se propone un formato para estos.


Apéndice B: Anexo 2 - Manual de usuario wIMUc

El anexo 2 contiene el manual de usuario final del software tomado como caso de estudio, Wireless Inertial Measurement Unit Calibration (wIMUc) . En este se indican las especificaciones técnicas a nivel de hardware y software que facilitarán la ejecución del programa, también incluyen secciones clave como los acrónimos y el glosario, esto con el fin de que cualquier usuario comprenda completamente qué hace el programa, cómo lo hace y por qué lo hace.

Apéndice C: Anexo 3 - Descripción del estándar de Ingeniería de Software

En el archivo se listan todas las etapas indispensables para la construcción de la documentación de ingeniería de software indispensable antes de la ejecución de un proyecto. Se describe que debe ir en cada una de ellas y se propone un formato para esta.

Apéndice D: Anexo 4 - Documentación de software de carácter científico

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

El anexo 4 contiene la ingeniería de software realizada al software tomado como caso de estudio, Wireless Inertial Measurement Unit Calibration (wIMUc). En este se desarrollan cada una de las etapas con las que se construyó la documentación del proyecto, etapa que debe realizarse de manera inicial, es decir, antes de la programación, con el fin de que cualquier usuario comprenda que hará el programa, como lo hará y por qué lo hará.

Apéndice E: Anexo 5 - Ingeniería inversa

El documento nombrado como “Anexo 5 - Ingeniería inversa”, explica dicha metodología, indica en qué consisten los tipos de ingeniería inversa existentes y puntualiza sobre que aplica al software tomado como caso de estudio.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES SANDRA GUERRA.

FIRMA ASESOR JUAN SE.

FECHA ENTREGA: _____

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____

RECHAZADO__ ACEPTADO__ ACEPTADO CON MODIFICACIONES__

ACTA NO. _____

FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD _____

ACTA NO. _____

FECHA ENTREGA: _____