



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DE L TÍTULO DE
INGENIERA EN SISTEMAS COMPUTACIONALES**

TEMA:

**“HERRAMIENTA DE ESTUDIO DE LA METODOLOGÍA MSF MEDIANTE EL
APLICATIVO DE UN SISTEMA INFORMÁTICO DEL PROCESO DE
TITULACIÓN”**

AUTORA: CECILIA LIZBETH CARRILLO YÉPEZ

DIRECTOR: MSC. MAURICIO REA

IBARRA – ECUADOR

2018



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

Certifico que el proyecto de Trabajo de Grado “Herramienta de estudio de la metodología MSF mediante el aplicativo de un sistema informático del proceso de titulación” ha sido realizado en su totalidad por la señorita: Cecilia Lizbeth Carrillo Yépez portadora de la cédula de identidad número: 0923355317.

MSc. Mauricio Rea

DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

A Dios, a mis padres Marco y Guadalupe que junto a mis hermanas y mi novio son el motor que mueve mi vida, así como también la inspiración para superarme a mí misma cada día.

A todos quienes contribuyeron para que este proceso de titulación sea una realidad.

Para ellos, mis sinceros agradecimientos y mi gratitud eterna.

Cecilia Lizbeth Carrillo Yépez



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

A Dios por ser mi aliciente y mi sustento justo en los momentos que quise darme por vencida.

El más afectuoso agradecimiento a mis padres, por ser esos ángeles que Dios puso en mi camino para mostrarme que, aunque exista tempestad y el cielo esté nublado siempre saldrá el sol.

A mis hermanas y mis amigas, por ser la alegría de mi vida y mis fieles compañeras.

A mí enamorado Wilmer Carrera quién es uno de los principales motivadores para superarme profesionalmente y el compañero ideal en cada momento de mi vida.

Un agradecimiento especial al Ing. Mauricio Rea, porque sin su apoyo, paciencia y una increíble e indispensable ayuda hicieron posible la culminación de esta etapa de mi vida dando pie al inicio de nuevas metas y objetivos.

Al Ing. Carpio Pineda e Ing. Fausto Salazar, mi gratitud por que sus conocimientos enriquecieron este trabajo de grado.

Al Ing. Rafael Olmedo por las facilidades y oportunidades de crecimiento profesional dentro de EMELNORTE S.A.

A la Universidad Técnica del Norte que junto con todo el equipo de profesionales hacen de cada uno de los estudiantes que pasamos por las aulas de tan prestigiosa universidad, personas críticas, productivas, ingeniosas, profesionales, pero sobre todo forma personas para ser guerreros en la hermosa batalla que es la vida.

Cecilia Lizbeth Carrillo Yépez

RESUMEN

El presente proyecto tiene como objetivo el diseño de una herramienta de estudio de la metodología ágil Microsoft Solution Framework (MSF) mediante el desarrollo de un aplicativo que sistematice el proceso de titulación. El presente documento consta de tres capítulos que detallan el uso de la metodología a través del diseño y desarrollo del sistema. La introducción contiene el problema, antecedentes, objetivo general, objetivos específicos, justificación, alcance y contexto del proyecto. En el Capítulo 1 se detalla en qué consisten los procesos y metodologías ágiles de una forma clara y sencilla, conceptos o definiciones iniciales, bases, estructura, ventajas y desventajas de la metodología empleada. En el Capítulo 2 se encuentra el desarrollo del sistema acorde a la metodología seleccionada, impactos de la metodología y del sistema informático, así como también los procedimientos del proceso de titulación. El Capítulo 3 contiene la valoración del software y la comprobación de resultados. Al final del documento se pueden revisar las conclusiones, recomendaciones, referencias bibliográficas y anexos. El manual de usuario, manual administrativo y el diagrama de flujo de los procesos se encuentran en los anexos.

SUMMARY

This Project aims to design tool of study if the agile methodology Microsoft Solution Framework (MSF) by means of the development of an application that systematize the titling. This document consists of tree chapters detailing the use of the methodology through the design and development of the system. The Introduction contains the problem, history, general objective, specifics objectives, justification, scope and context of the project. In Chapter 1 details the agile methodologies and processes in a clear and simple way, initial concepts or definitions, bases, structure, advantages and disadvantages of the methodology used. Chapter 2 is the development of the system according to the selected methodology, impacts of the methodology and the computer system, as well as the procedures of the titling process. Chapter 3 contains the evaluation of the software and the verification of results. At the end of the document you can review the conclusions, recommendations, bibliographical references and annexes. User manual, administrative manual and flow diagram of the processes are in the annexes.

ÍNDICE DE CONTENIDO

CERTIFICACIÓN	II
DEDICATORIA	III
AGRADECIMIENTO	IV
RESUMEN	V
SUMMARY	VI
INTRODUCCIÓN	1
Problema	1
Antecedentes	1
Objetivo General	1
Objetivo Específico.....	2
Alcance	2
Justificación	3
Cronograma de Actividades.....	3
Contexto.....	5
CAPITULO I	7
1. MARCO TEÓRICO	7
1.1 Definiciones previas de metodologías y procesos ágiles	7
1.1.1 Definición de proceso de software.....	7
1.1.2 Clasificación de proceso	9
1.1.3 Elemento software o entregable.....	10
1.1.4 Metodología	12
1.2 Metodología Ágil Microsoft Solution Framework (MSF)	18
1.2.1 Principios fundamentales y actitudes en el contexto de MSF.....	19
1.2.2 Etapas y Modelo de gobernanza de MSF.....	23
1.3 Estado del Arte.....	36
Historia según Pantaleo, G., & Rinaudo, L. (2015)	36

Introducción	37
Metodologías Ágiles vs Metodologías Conducidas por Planes	37
Diferencia de aplicabilidad	40
1.4 Impactos de la Metodología.....	42
1.5 Impactos del Sistema Informático	42
CAPITULO II	45
2.1 Visión.....	45
2.1.1 Introducción	45
2.1.2 Posicionamiento	46
2.1.3 Descripción de participantes en el proyecto y usuarios	48
2.1.4 Diagrama de frecuencia del sistema.....	51
2.1.5 Descripción global del producto	51
2.1.6 Matriz de Riesgos.....	53
2.2 Planeación.....	55
Cronograma.....	55
2.3 Compilación.....	56
Proceso de Titulación.....	56
Procedimiento de Seguimiento de Tesis	57
2.3.1 Plan de desarrollo.....	60
2.3.2 Procedimiento de manejo.....	61
2.3.3 Estructura de la base de datos	61
2.4 Estabilización.....	73
2.4.1 Pruebas en la ejecución del Sistema.....	73
2.5 Implementación.	74
2.6 Pruebas	74
CAPITULO III.....	77
3.1 Valoración del Software	77
3.2 Comprobación de Resultados.....	77
Conclusiones y Recomendaciones.....	80
Bibliografía	81
Anexos	82

ÍNDICE DE FIGURAS

FIGURA 1. Arquitectura Tecnológica.....	2
FIGURA 2. Diagrama de la pista de Gobernanza.....	24
FIGURA 3. Relación entre etapas del modelo en cascada y fases de la metodología MSF....	27
FIGURA 4. Representación de una característica del desarrollo incremental en la metodología MSF.....	28
FIGURA 5. Detalle de “entregables” de la metodología MSF de acuerdo al modelo en cascada.	29
FIGURA 6. Diagrama de frecuencia del sistema de seguimiento para la titulación de los estudiantes de la CISIC.	51
FIGURA 7. Matriz para análisis cualitativo de riesgos	54
FIGURA 8. Proceso de Titulación de la Universidad Técnica del Norte	57
FIGURA 9. Líneas de Investigación de las carreras de la Facultad de Ingeniería en Ciencias Aplicadas y la relación que existe entre las líneas de investigación de cada carrera.....	66
FIGURA 10. Modelo Físico de la Base de Datos	72
FIGURA 11. Resultado de encuesta aplicada a 30 jóvenes (porcentajes).....	79

INDICE DE TABLAS

Tabla 1 . Cronograma de actividades para el desarrollo de tesis, aprobado por el Honorable Consejo Directivo	4
Tabla 2. Descripción y diferencias significativas entre las metodologías conducidas por planes y metodologías ágiles.	17
Tabla 3. Guía de uso de los métodos ágiles y conducidos por los planes.....	40
Tabla 4. Comparativo entre metodologías ágiles MSF, XP, SCRUM.....	42
Tabla 5. Sentencia para definición del problema.....	47
Tabla 6. Descripción del sistema para seguimiento al proceso de titulación de la CISIC.....	47
Tabla 7. Descripción y responsabilidades de cada integrante del equipo de desarrollo.	48
Tabla 8. Cuadro Resumen de usuarios del sistema de acuerdo con el cargo UTN.....	48
Tabla 9. Perfil del Director de Tesis	49
Tabla 10. Perfil de Tesista	49
Tabla 11. Perfil de usuario docente.....	49
Tabla 12. Perfil de usuario administrativo.....	50
Tabla 13. Perfil de usuario estudiante.....	50
Tabla 14. Perfil de usuario auxiliar.....	50

Tabla 15. Beneficios del sistema.	52
Tabla 16. Matriz de riesgos.....	53
Tabla 17. Cronograma de desarrollo del sistema.....	55
Tabla 18. Subproceso: Aprobación de anteproyecto	57
Tabla 19. Subproceso: Desarrollo y aprobación de tesis.	58
Tabla 20. TESIS_DOCENTE	62
Tabla 21. TESIS_ESTUDIANTE.....	62
Tabla 22. TESIS_PERIODO_ACAD	62
Tabla 23. TESIS_GRUPO_INT_ACAD	63
Tabla 24. TESIS_INTEGRANTES_ACAD	63
Tabla 25. TESIS_ACADEMICO.....	64
Tabla 26. TESIS_DET_ACAD.....	64
Tabla 27. TESIS_DIRECTOR_PROPUESTO	65
Tabla 28. TESIS_LINEA_INVESTIGACIÓN.....	65
Tabla 29. TESIS_ANTEPROYECTO	67
Tabla 30. Estados el Anteproyecto	67
Tabla 31. TESIS_RESUMEN.....	68
Tabla 32. Tipos de Resumen del Anteproyecto	68
Tabla 33. TESIS_ANT_RESUMEN	68
Tabla 34. TESIS_CORRECCIONES	69
Tabla 35. TESIS_OBJETIVOS	69
Tabla 36. TESIS_CRONOGRAMA.....	70
Tabla 37. TESIS_TEMARIO.....	70
Tabla 38. TESIS_DET_TEMARIO.....	70
Tabla 39. TESIS_DIRECTIVO	71
Tabla 40. TESIS_DEFENSA.....	71
Tabla 41. Planificación de actividades para realizar pruebas con el usuario.....	73
Tabla 42. Tabla para clase de equivalencia de acuerdo con el test de la Caja Blanca.....	75
Tabla 43. Detalle de valoración del aplicativo para dar seguimiento al proceso de titulación de los estudiantes de la CISIC.	77
Tabla 44. Comprobación de resultados obtenidos de acuerdo con el test de la Caja Blanca que se muestra en la tabla 41.	78
Tabla 45. Resultado de encuesta aplicada a 30 jóvenes (número de personas).....	78

INTRODUCCIÓN

Problema

En la Universidad Técnica del Norte se enseñan metodologías de desarrollo tradicionales ocasionando que exista poca información y familiaridad con el uso de metodologías ágiles para el desarrollo de software, especialmente en lo que respecta a la metodología Microsoft Solution Framework (MSF) de Microsoft. Con el objetivo de implementar una solución con la metodología señalada, se ha seleccionado el proceso relacionado con la titulación, de manera que complementa este estudio.

Antecedentes

En las metodologías conducidas a los planes (o tradicionales) los procesos definidos son estrictos y burocráticos. En contraste, las metodologías ágiles como su nombre lo indica, buscan brindar flexibilidad al proceso de desarrollo de software. De esta forma, intentan lograr un punto que sea útil entre las restricciones de los modelos orientados a los planes y la libertad absoluta. (Pantaleo, G. & Rinaudo, L., 2015, p. 64)

En toda institución es importante manejar estadísticas, las universidades no son la excepción. Pero al tener un volumen considerable de información, se vuelve difícil administrar y procesar todos los datos de forma manual o con herramientas de escritorio, en corto tiempo.

La información descentralizada y almacenada en varios lugares físicos genera demoras en el tiempo de atención de requerimientos iniciales y en todas las etapas que son necesarias en el proceso de titulación, haciendo que los procesos se vuelvan tediosos, repetitivos y no se pueda realizar un seguimiento adecuado.

En los últimos años la legislación para las universidades ha estado en constante actualización, y seguirá en cambio, es así como manifiesta el diario El Universo en su edición del 9 de junio del 2017 hace una publicación con el título de “Senecyt plantea 7 reformas a la Ley de Educación Superior”. Con requerimientos en constante cambio es complejo manejar metodologías rígidas, que impiden la generación periódica de nuevas versiones de software, por toda la documentación que este tipo de metodologías exige.

La misión de la Universidad Técnica del Norte es entregar a la sociedad profesionales éticos, críticos, humanistas, líderes y emprendedores con responsabilidad social, cabe mencionar que estos profesionales deben ser legalmente reconocidos por el Senecyt para poder desarrollarse en el ámbito laboral, razón por la cual es importante dar seguimiento a un proceso que está directamente ligado al cumplimiento de la misión institucional.

Objetivo General

Desarrollar una herramienta de estudio de la metodología MSF mediante el aplicativo de un sistema informático del proceso de titulación para que se use metodologías ágiles en el desarrollo de software en aulas de clase.

Objetivo Específico

- Recopilar información para establecer conceptos básicos que permitan entender la metodología MSF.
- Investigar el estado del arte de la metodología MSF.
- Desarrollar el sistema informático para el seguimiento del proceso de tesis de la CISIC con la metodología MSF para demostrar con un caso real la aplicabilidad de la metodología.

Alcance

Para el estudio de metodologías ágiles se tomará como caso de análisis la metodología MSF (Microsoft Solution Framework) y se desarrollará un sistema informático que sirva como herramienta de estudio para la Carrera de Ingeniería en Sistemas Computacionales.

El desarrollo del sistema informático para seguimiento del proceso de titulación de la CISIC es una propuesta que tiene como fin demostrar la aplicabilidad de la metodología MSF

La interface con el usuario será en Oracle Forms y los reportes en Oracle Reports tal cual se puede visualizar en la FIGURA 1, en virtud de que Forms y Reports son productos ORACLE es necesario usar la base de Datos ORACLE junto con la instalación del motor de forms OC4J¹ que requiere la instalación de JDK² y JDBC³.

Para este proyecto se usa herramientas ORACLE, debido a que la Universidad Técnica del Norte tiene ciertas aplicaciones con Oracle Forms y Reports.



FIGURA 1. Arquitectura Tecnológica

Fuente: Propia

¹ **OC4J:** Oracle Container four Java

² **JDK:** Java Development Kit, en español Kit de desarrollo para aplicaciones JAVA

³ **JDBC:** Java Database Connectivity, es una interface de acceso a bases de datos estándar SQL

Justificación

Justificación Social

El proyecto de tesis plantea de la posibilidad de que los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas en especial la Carrera de Desarrollo de Software, se motiven a desarrollar los proyectos de aula con metodologías ágiles desde los primeros niveles, con la finalidad de que los estudiantes adquieran destrezas en desarrollo de software aplicando metodologías en sus proyectos de clase.

Justificación Económica

Las metodologías rígidas demandan de un alto consumo de recursos de oficina y tiempo en la generación de documentación, tal como manifiestan los MSC. Ailin Orjuela y Mauricio Rojas (2008) en un cuadro comparativo que se encuentra en artículo llamado “Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo”, en la tabla 1, de este artículo los autores establecen que las metodologías ágiles requieren de más artefactos de entrega.

Es importante mencionar que el Gobierno Nacional promueve la disminución del consumo de papel de acuerdo con lo estipulado en el Acuerdo Ministerial No. 034 emitido el 14 de marzo del 2014 por la Ex Ministra del Ambiente, Lorena Tapia Núñez, cabe señalar que con una metodología ágil no se suspende el uso de papel, pero si disminuye su consumo, al disminuir el número de documentos de entrega obligatorios en las fases de desarrollo de un sistema.

Justificación Técnica

La Universidad Técnica del Norte usa recursos Oracle, y este proyecto de tesis puede ser adaptado fácilmente al sistema integrado de la universidad debido a que todo el aplicativo está desarrollado con herramientas Oracle.

Oracle es una compañía reconocida por el amplio soporte que ofrece en todos los niveles de desarrollo, base de datos, reportes, entre otros.

Además, Oracle ofrece versiones gratuitas que permite a los desarrolladores adquirir destrezas en el área de desarrollo y en virtud de que el aplicativo está generado como herramienta de estudio, es ideal para este proyecto de tesis.

Cronograma de Actividades

Tabla 1 . Cronograma de actividades para el desarrollo de tesis, aprobado por el Honorable Consejo Directivo

Nro.	Nombre de tarea	MES 1	MES 2	MES 3	MES 4
1	Tema: Estudio de la metodología MSF mediante la implementación de un sistema informático para seguimiento del proceso de tesis de la CISIC.				
2	INTRODUCCIÓN				
3	El Problema				
4	Antecedentes				
5	Objetivo General				
6	Objetivos Específicos				
7	Alcance				
8	Justificación				
9	Contexto.				
10	CAPÍTULO I: MARCO TEÓRICO.				
11	1.1 Definiciones previas de metodologías y procesos ágiles				
12	1.2 Metodología Microsoft Solution Framework (MSF) y sus etapas.				
13	1.3 Estado del Arte de la Metodología MSF				
14	1.4 Impactos del estudio de la Metodología				
15	1.5 Impactos del sistema.				
16	CAPÍTULO II: DESARROLLO				
17	2.1 Visión				
18	2.2 Planificación				
19	2.3 Desarrollo				
20	2.4 Estabilización				
21	2.5 Implementación				
22	2.6 Pruebas				
23	2.7 Manual de usuario				
24	2.8 Manual Administrativo				
25	CAPÍTULO III: RESULTADOS				
26	3.1 Validación				
27	3.2 Comprobación de resultados				
28	CONCLUSIONES				
29	RECOMENDACIONES				
30	ANEXOS				
31	BIBLIOGRAFÍA				

Fuente: Propia

Contexto

Existen proyectos de titulación que se realizaron en diferentes Universidades, como, por ejemplo:

- Escuela Politécnica del Ejército con su proyecto, Aplicación de la Metodología MSF V4.0 a la definición e implementación de arquitecturas orientadas a objetos en Visual Studio .NET2005, caso práctico G Sharing Files, desarrollada por Villarroel González Luis Ramiro y Montalvo Yépez César Alejandro en el año 2008.
- Escuela Politécnica Nacional con su proyecto, Desarrollo de un Sistema para determinar la Ubicación Geográfica de Adolescentes utilizando dispositivos móviles desarrollados por Francisco Xavier Sánchez Álvarez y Ricardo Oswaldo Tituaña Haro en el año 2007.
- Universidad Técnica de Cotopaxi con su proyecto, Análisis de la metodología MSF (Microsoft Solutions Framework), prototipo sistema de administración en la compañía de transporte público urbano popular de pasajeros “Latacunga S.A Citulasa” ubicada en la ciudad de Latacunga provincia de Cotopaxi desarrollada por Pantusin Lema William Fernando y Viracocha Bedoya Edwin Mauricio en el año 2009.
- Universidad Técnica del Norte con su proyecto, Sistema de Facturación y Control de Inventarios para la Unión de Papelerías de la Ciudad de Ibarra desarrollada por Landázuri Ortiz Carlos Julio en el año 2013

A diferencia de los proyectos antes mencionados, esta tesis pretende establecer ventajas y desventajas de la Metodología MSF y dar a conocer su aplicabilidad a través de un sistema realizado con herramientas Oracle.

Es importante mencionar que el sistema se va a desarrollar en una máquina virtual para que sirva como herramienta de estudio para los estudiantes de la UTN.

CAPITULO I

1. MARCO TEÓRICO

1.1 Definiciones previas de metodologías y procesos ágiles

1.1.1 Definición de proceso de software

En el libro “Ingeniería del Software – Un enfoque desde la guía SWEBOK” (2012) se afirma que “Un proceso, entendido de manera general, es una serie de pasos que incluyen actividades, restricciones y recursos que resultan en un producto determinado con ciertas características” (p.33).

En el Glosario IEEE de Términos de Ingeniería del Software (IEEE, 1990) proporciona un concepto más corto en el que manifiesta que:

“Process. (1) A sequence of steps performed for a given purpose; for example, the software development process.”

Del concepto anterior se puede interpretar que un proceso consiste en realizar una secuencia de pasos para conseguir un propósito determinado, cabe resaltar que la IEEE pone como ejemplo de este concepto, el proceso de desarrollo de software.

Según Ian Sommerville (2011) en su libro Ingeniería del Software, novena edición establece como proceso de software a “Una serie de actividades relacionadas que conduce a la elaboración de un producto de software” (p. 28)

Otro concepto un poco más explícito o detallado es el que se encuentra en el libro “Ingeniería del Software – Un enfoque desde la guía SWEBOK” (Sánchez, S. & Sicilia, M. & Rodríguez, D., 2012), que dice que “Un proceso software es un conjunto coherente de políticas, estructuras, organizaciones, tecnologías, procedimientos y artefactos que se necesitan para concebir, desarrollar, implantar y mantener un producto software” (p. 34).

En resumen, un proceso de software consiste en seguir un orden en cada etapa de desarrollo para obtener software de calidad.

Este proyecto de tesis plantea trabajar en base a procesos con el fin de ser una guía para los estudiantes de la Carrera de Ingeniería en Sistemas Computacionales.

1.1.1.1 Aspectos Importantes de Procesos

Ian Sommerville (2011) establece que existen muchos diferentes procesos de software, pero todos deben incluir cuatro actividades que son fundamentales para la ingeniería del software:

- a. *Especificación del Software:* Tienen que definirse tanto la funcionalidad del software como las restricciones de su operación
- b. *Diseño e implementación del software:* Debe desarrollarse el software para cumplir con las especificaciones.

- c. *Validación del software*: Hay que validar el software para asegurarse de que cumple lo que el cliente quiere.
- d. *Evolución del software*: El software tiene que evolucionar para satisfacer las necesidades cambiantes del cliente. (p. 28)

El mismo autor (Ian Sommerville, 2011) afirma que tales actividades forman parte de todos los procesos de software, y que en la práctica estas actividades son complejas en sí mismas e incluyen subactividades tales como la validación de requerimientos, el diseño arquitectónico, la prueba de unidad, etcétera. También existen actividades de soporte al proceso, como la documentación y el manejo de la configuración del software. (p. 28)

Aunque parecería muy complicado trabajar con procesos y muy rígido, no es así, Sánchez, S. & Sicilia, M. & Rodríguez, D. (2012), manifiestan los siguientes aspectos a cerca de los procesos de software:

- No existe un único proceso de software que determine la forma correcta de hacer las cosas independientemente de la organización, el proyecto y las circunstancias. De hecho, hay un considerable grado de diversidad en los procesos, y si hoy se utiliza mayoritariamente en determinado proceso, en el futuro probablemente haya otras variantes, tal vez motivadas por cambios en la tecnología o en la forma de desarrollar. En conclusión, se puede afirmar que hay muchos “tipos de proceso”.
- Hay que diferenciar los procesos extendidos como << las actividades de desarrollo que realmente ocurren en una organización X>> de la discusión general sobre cómo son y qué propiedades tienen los distintos tipos de procesos.
- Los procesos no son sólo para las grandes organizaciones. Los procesos son útiles también para los pequeños desarrollos. Incluso son útiles para equipos formados por una sola persona, los cuales se conoce como procesos personales. (p. 37)

1.1.1.2 Ventajas del proceso de software

En la vida profesional los procesos se usan en todas las áreas de conocimiento y es común ver como al principio existe oposición al cambio, cuando no se tiene la costumbre de trabajar en forma ordenada.

Es como si aún niño se le enseña en el transcurso de su vida como hacer las cosas para que sus acciones tengan consecuencias positivas, al principio el niño quiere hacer todo a su manera, a lo que ve en los demás y se le dificultad trabajar de una forma ordenada, pero cuando ya es un adulto comprende que el orden es importante y muy útil.

Otro ejemplo práctico de trabajo ordenado se puede ver en la preparación de un plato de comida, algo tan simple requiere de un proceso, porque si pongo todos los ingredientes al mismo tiempo o en un orden diferente al correcto, es muy probable que no se tenga el resultado que se espera y quizás tome más tiempo de lo usual.

Lo mismo pasa en empresas u organizaciones es difícil trabajar con procesos, cuando no se ha implementado antes, las excusas empiezan a salir como, por ejemplo, siempre lo he hecho así

y funciona ¿Por qué voy a cambiar?, pero conforme crece la empresa es fácil darse cuenta lo importante que es tener bien definidos los procesos y trabajar con ellos.

Los procesos, le permite saber a quién los ejecuta que es lo que debe hacer, como lo debe hacer y cuando lo debe hacer.

Esto con lleva a ser más ordenado y disminuir el tiempo de ejecución.

Los gerentes pueden saber que está haciendo su personal.

Se mejora la comunicación organizacional.

En el área de desarrollo es mejor sistematizar un proceso definido que tratar de darle forma a un sin número de actividades que aparentemente no tienen relación ni un orden específico.

Durante el desarrollo de un software, al analista le permite ver si está cumpliendo con tiempos e incluso el cliente podrá dar seguimiento al avance de la construcción de su producto.

Como podemos ver los procesos tienen como objetivo mejorar cualquier desarrollo o proyecto.

Sánchez, S. & Sicilia, M. & Rodríguez, D. (2012), mencionan algunas de las desventajas de no tener un proceso de software:

- El software se hace difícil de mantener. Si un desarrollador abandona el proyecto, por ejemplo, otro desarrollador tendrá que completar su trabajo. En el peor de los casos, no habrá ningún tipo de plan ni documentación, ya que todo estaba en la cabeza del autor original. Aun no suponiendo el peor de los escenarios, el impacto de la eventualidad implica siempre riesgo. Pensando en casos menos extremos, la mera ausencia de un proceso definido dificulta la comprensión del software desarrollado.
- Si no hay proceso es difícil establecer medidas de calidad. Si cada desarrollador actúa de una forma diferente, no tendremos forma de comparar sus resultados.
- La ausencia de proceso no permite reutilizar experiencias pasadas. La ausencia de notaciones y técnicas comunes dificulta la comunicación de los desarrolladores, lo cual podría conducir a documentaciones o incluso diseños incompatibles entre sí. (p. 33)

1.1.2 Clasificación de proceso

Los procesos de software pueden clasificarse como dirigidos por un plan (plan-driven) o como procesos ágiles.

Como plantean Boehm y Turner (2003), cada enfoque es adecuado para diferentes tipos de software. Por lo general, es necesario encontrar un equilibrio entre procesos dirigidos por un plan y procesos ágiles.

Procesos dirigidos por un plan (plan-driven):

Son aquellos donde todas las actividades del proceso se planean por anticipado y el avance se mide contra dicho plan.

Procesos ágiles:

En este caso la planeación es incremental y es más fácil modificar el proceso para reflejar los requerimientos cambiantes del cliente.

1.1.2.1 Procesos ágiles

En palabras simples un proceso sería la forma o guía de cómo llegar a tener un producto terminado y si por concepto básico se sabe que agilidad consiste en realizar algo de forma rápida, sencilla y sin causar un mayor grado de esfuerzo, se puede aseverar que un proceso ágil en una definición fácil de entender y sería “la forma de realizar un sistema o aplicación de forma rápida y práctica”.

Sánchez, S. & Sicilia, M. & Rodríguez, D. (2012), plantean que se denominan métodos de desarrollo ágiles a un conjunto de métodos que enfatizan el enfoque iterativo, la adaptabilidad del proceso y a colaboración, Los métodos ágiles se caracterizan además por el hecho de que reducen la documentación y los procedimientos al mínimo. Por ello, se les suele contraponer con otros métodos en los que se produce una gran confusión de documentos y comprobaciones formales, a los que a veces se les aplica la metáfora de métodos de gran ceremonia.

Aunque no todos los procesos ágiles son iguales, la siguiente lista de características generales puede aplicarse a la mayoría de ellos, si bien es posible que no todos los métodos ágiles las cumplan todas:

- La medida del progreso es el software desarrollado y funcional. La clave es que al final de cada iteración el cliente tenga una versión funcional del software. Así, los clientes pueden reevaluarlo y decidir qué les interesa a partir de ese punto.
- La documentación es la documentación del código junto con el diseño de dicho código. Es decir, no se producen documentos de análisis o de especificaciones.
- No hay una estructura organizativa rígida en el equipo de desarrollo. Los equipos suelen ser pequeños e incluyen a un representante del cliente. Este representante se considera crítico para el éxito del proyecto y debe estar disponible para las aclaraciones y decisiones tácticas del día a día.
- El proceso de ser capaz de cambiar de dirección para adaptarse a necesidades o requisitos nuevos que aparecen entre las iteraciones. De hecho, idealmente el cliente decide al principio de cada iteración las funcionalidades a añadir en función del valor que estas tengan para él. (p. 56)

1.1.3 Elemento software o entregable

Los conceptos antes mencionados dan una idea bastante clara de lo que es un proceso de software hay que tener cuenta que un producto o elemento de software es todo aquello no tangible que forma parte de un aspecto tecnológico, es decir que es producto terminado que desde el punto de vista orientado a sistemas informáticos es todo aquello que interviene en el proceso de desarrollo y ciclo de vida de un software.

Se le conoce como entregable a todo documento, software, etc., que le permita al desarrollador visibilizar el avance en el desarrollo del proyecto con el fin de ser presentado al cliente.

Los dos conceptos antes mencionados son muy similares, por lo cual se determina que el elemento de software es un entregable y viceversa.

Como se mencionó, la documentación también se considera como producto entregable en las fases de desarrollo de software

En el libro Auditoría de la Calidad (Lucero, 2000, p. 364) se puede encontrar algunos ejemplos de productos entregables, que son parte de la documentación que se genera durante el proceso de desarrollo o mantenimiento del software, a continuación, se detallan:

- ❖ Documentos de Planificación del proyecto (por ejemplo, planes del desarrollo del software y planes de verificación y validación del software).
- ❖ Especificaciones de requerimientos y diseño del software.
- ❖ Documentación del esfuerzo de las pruebas.
- ❖ Documentación suministrable al cliente.
- ❖ Código fuente de los programas.
- ❖ Representación de las soluciones software implementadas en el firmware.
- ❖ Informes (por ejemplo, revisiones, auditorías y estado del proyecto) y datos (por ejemplo, detección de defectos, pruebas).

En resumen, un producto de software es el resultado de una actividad del proceso de software.

Pantaleo, G. & Rinaudo, L. (2015) afirman que los entregables no deben superar el mes entre ellos y que esto presenta las siguientes ventajas (p. 31)

- Mayor visibilidad del avance del proyecto para la organización lo cual ayuda al grupo a mostrar su trabajo y a la organización a medir el funcionamiento del grupo.
- Gran visibilidad del avance a los clientes lo cual ayuda a medir la dinámica del proyecto y a realizar pruebas conjuntas y obtener realimentación.
- Ayuda a un desarrollo más dinámico y a la posibilidad de rehacer y mejorar a partir de la realimentación obtenida.
- Ayuda al grupo en el aprendizaje del recorrido del camino completo del desarrollo (análisis detallado, diseño detallado, codificación, pruebas e instalación).
- Garantiza la satisfacción y el relajamiento de un objetivo alcanzado para el grupo de desarrollo.
- Provee visibilidad temprana que genera un ambiente de confianza ante la percepción de todos los involucrados del avance del proyecto.
- Los entregables con periodos fijados de tiempo ayudan a medir la eficiencia y velocidad de desarrollo del grupo.

1.1.4 Metodología

1.1.4.1 Definiciones Previas

Proyecto de ingeniería del software

Es un proyecto cuyo objetivo es obtener un producto de software que satisfaga ciertos requisitos, en el plazo previsto y dentro del presupuesto. (Sánchez, S. & Sicilia, M. & Rodríguez, D., 2012, p. 35)

Ciclo de vida de un desarrollo de software

Es el periodo de tiempo que comienza cuando se toma la decisión de desarrollar un producto de software y que concluye cuando se entrega el software. (Sánchez, S. & Sicilia, M. & Rodríguez, D., 2012, p. 35)

Metodología de desarrollo de software

Una metodología es un marco de trabajo que puede ser utilizado como guía de las actividades a llevar a cabo. Por lo tanto, una metodología de desarrollo de software no es más que una forma de trabajo para desarrollar software, donde se especifica las tareas a llevar a cabo, los artefactos a generar y las relaciones entre ambos. Este marco o forma de trabajo permite organizar el proceso de desarrollo de software a través de la definición de pautas a seguir y restricciones a cumplir. (Pantaleo, G. & Rinaudo, L., 2015, p. 54)

Modelos y Metodologías

Según Pantaleo (2016) en su libro Calidad en el desarrollo de software afirma que:

La confusión entre metodologías y modelos es una de las causas frecuentes de fracaso de proyectos de mejora de procesos en organizaciones de desarrollo de software.

Una metodología es una forma de trabajo organizada para un grupo de desarrollo en base a la dinámica de las actividades de los diferentes roles. Establece qué, cuándo y cómo realizan sus tareas y cómo utilizan los demás roles los productos generados por ellos.

Un modelo de referencia establece áreas de proceso con objetivos y prácticas a efectos de calificar y compara grupos de desarrollo. La ambigüedad con que estos modelos definen subprácticas y activos de trabajo. Dicen qué, pero no les interesa cómo. (p. 189)

1.1.4.2 Modelos de proceso de software

Los modelos genéricos no son descripciones definitivas de los procesos de software. Más bien, son abstracciones del proceso que se utilizan para explicar los diferentes enfoques del desarrollo de software. (Sommerville, Ian, 2011, p.29)

Los modelos más frecuentes según el libro Ingeniería de software (Pantaleo, G. & Rinaudo, L., 2015) son:

Cascada: Es un modelo de desarrollo lineal secuencial, El proyecto de software es dividido en fases que deben procederse en forma secuencial (definición del software, análisis de requerimientos, diseño de la arquitectura, codificación, pruebas). Cuando finaliza cada una de las fases mencionadas, se realiza una revisión con el fin de determinar si está en condiciones de avanzar a la siguiente. Las fases no se solapan unas con otras. (p. 55)

Facilidad para encontrar errores, debido a que los requerimientos iniciales se congelan y deben ser definidos de antemano, esto dificulta o impide cualquier tipo de cambio que deba ser incorporado al producto posteriormente. (p. 56-57).

Prototipado: Es un modelo de desarrollo iterativo en el cual se desarrolla una maqueta del producto. Esta maqueta o prototipo desarrollada por el equipo del proyecto y refinada junto al cliente permite idealmente especificar los requerimientos del producto. Así el prototipo evolucionará refinando los requerimientos en cada iteración hasta lograr la aprobación del cliente. (p. 57)

Este modelo requiere la participación del cliente durante el desarrollo del producto, dado a que el cliente se le presentan prototipos funcionando, el puede asumir en un corto plazo que el proyecto está por finalizar. (p. 58)

DRA (Desarrollo Rápido de Aplicaciones): implica un desarrollo lineal secuencial llevado a cabo en paralelo por distintos equipos de trabajo. Cada uno de estos equipos de trabajo será responsable por una parte de las funcionalidades del proyecto. El objetivo es crear en un periodo rápido corto de tiempo un producto final. Asimismo, el modelo se basa fuertemente en la reutilización de componentes y en caso de que esto no sea posible, en desarrollar componentes reutilizables. (p. 59)

Los requerimientos deben ser especificados claramente, para construir productos en poco tiempo, pero requiere de un gran número de recursos simultáneamente. (p. 61)

Incremental: Este modelo permite el aumento iterativo de la funcionalidad del producto. Como resultado de cada iteración se obtendrá un producto operacional y no un prototipo como en el caso del modelo de prototipado. (p. 61).

El modelo iterativo permite adaptarse a la evolución que puede sufrir el producto con el pasar del tiempo. La desventaja es que armar un entregable requiere un trabajo extra por cada iteración. (p. 62)

Espiral: Este modelo combina los modelos de cascada y la esencia iterativa del prototipado descritos previamente. Por lo tanto, es un modelo iterativo incremental. El proyecto de software es dividido en mini-proyectos. Cada uno de los cuales evoluciona el producto y busca minimizar los riesgos.

El modelo controla los riesgos que lleva a cabo. A medida que se avanza en las iteraciones los riesgos del proyecto decrecen, para lograrlo requiere de una administración minuciosa, la que genera que este modelo sea difícil de implementar. (p. 63)

Dichos modelos no son mutuamente excluyentes y con frecuencia se usan en conjunto, sobre todo para el desarrollo de grandes sistemas. Para este tipo de sistemas, tiene sentido combinar algunas de las mejores características de los modelos de desarrollo en cascada e incremental. (Sommerville, Ian, 2011, p. 30)

1.1.4.3 Manifiesto ágil

La información de esta sección se obtuvo de Palacio J. (2015), Scrum Manager I – Las reglas de Scrum, Scrum Manager, Versión 2.5.1., p. 12-14, fue transcrita por ser considerada como base fundamental de las metodologías ágiles.

En marzo de 2001, 17 críticos de los modelos de producción basados en procesos, convocados por Kent Beck, que había publicado un par de años antes el libro en el que explicaba la nueva metodología Extreme Programming (Beck, 2000) se reunieron en Salt Lake City para discutir sobre el desarrollo de software. En la reunión se acuñó el término “Métodos Ágiles” para definir a aquellos que estaban surgiendo como alternativa a las metodologías formales: CMM-SW, (precursor de CMMI) PMI, SPICE (proyecto inicial de ISO 15504), a las que consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas, previas al desarrollo.

Los integrantes de la reunión resumieron en cuatro postulados lo que ha quedado denominado como “Manifiesto Ágil”, que son los valores sobre los que se asientan estos métodos.

Hasta 2005, entre los defensores de los modelos de procesos y los de modelos ágiles fueron frecuentes las posturas radicales, más ocupadas en descalificar al otro, que en estudiar sus métodos y conocerlos para mejorar los propios.

El Manifiesto Ágil dice: Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar:

- *A los Individuos y su interacción, por encima de los procesos y las herramientas.*
- *El software que funciona, por encima de la documentación exhaustiva.*
- *La Colaboración con el cliente, por encima de la negociación contractual*
- *En respuesta al cambio, por encima del seguimiento de un plan*

Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda.

Valoramos más a los individuos y su interacción que a los procesos y las herramientas.

Este es el valor más importante del manifiesto.

Por supuesto que los procesos ayudan al trabajo. Son una guía de operación. Las herramientas mejoran la eficiencia, pero hay tareas que requieren talento y necesitan personas que lo aporten y trabajen con una actitud adecuada.

La producción basada en procesos persigue que la calidad del resultado sea consecuencia del know-how “explicitado” en los procesos, más que en el conocimiento aportado por las personas que los ejecutan.

Sin embargo, en desarrollo ágil los procesos son una ayuda. Un soporte para guiar el trabajo. La defensa a ultranza de los procesos lleva a afirmar que con ellos se pueden conseguir resultados extraordinarios con personas mediocres, y lo cierto es que este principio no es cierto cuando se necesita creatividad e innovación.

Valoramos más el software que funciona que la documentación exhaustiva.

Poder anticipar cómo será el funcionamiento del producto final, observando prototipos previos, o partes ya elaboradas ofrece un "feedback" estimulante y enriquecedor, que genera ideas imposibles de concebir en un primer momento, y difícilmente se podrían incluir al redactar un documento de requisitos detallado en el comienzo del proyecto.

El manifiesto ágil no da por inútil la documentación, sólo la de la documentación innecesaria. Los documentos son soporte de hechos, permiten la transferencia del conocimiento, registran información histórica, y en muchas cuestiones legales o normativas son obligatorios, pero su relevancia debe ser mucho menor que el producto final.

La comunicación a través de documentos no ofrece la riqueza y generación de valor que logra la comunicación directa entre las personas, y a través de la interacción con prototipos del producto.

Por eso, siempre que sea posible debe preferirse reducir al mínimo indispensable el uso de documentación, que requiere trabajo sin aportar un valor directo al producto.

Si la organización y los equipos se comunican a través de documentos, además de ocultar la riqueza de la interacción con el producto, forman barreras de burocracia entre departamentos o entre personas.

Valoramos más la colaboración con el cliente que la negociación contractual.

Las prácticas ágiles están indicadas para productos cuyo detalle resulta difícil prever al principio del proyecto; y si se detallara al comenzar, el resultado final tendría menos valor que si se mejoran y precisan con retroinformación continua durante el.

También son apropiadas cuando se prevén requisitos inestables por la velocidad de cambio en el entorno de negocio del cliente.

El objetivo de un proyecto ágil no es controlar la ejecución conforme a procesos y cumplimiento de planes, sino proporcionar el mayor valor posible al producto.

Resulta por tanto más adecuada una relación de implicación y colaboración continua con el cliente, más que una contractual de delimitación de responsabilidades.

Valoramos más la respuesta al cambio que el seguimiento de un plan.

Para desarrollar productos de requisitos inestables, que tienen como factor inherente el cambio y la evolución rápida y continua, resulta mucho más valiosa la capacidad de respuesta que la de seguimiento y aseguramiento de planes. Los principales valores de la gestión ágil son la anticipación y la adaptación, diferentes a los de la gestión de proyectos ortodoxa: planificación y control que evite desviaciones del plan.

El manifiesto ágil, tras los postulados de estos cuatro valores en los que se fundamenta, establece estos 12 principios.

1. Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
2. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se doblan al cambio como ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
4. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
5. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la principal medida del progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad como arte de maximizar la cantidad de trabajo que se hace es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto organizan.
12. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta.

1.1.4.4 Clasificación de metodologías.

De acuerdo con las definiciones revisadas en este documento, se puede decir que una metodología es la forma en la que se va a desarrollar un proceso.

Como ya se mencionó, existen dos tipos procesos por lo tanto existe un tipo de metodología para cada tipo de proceso, metodologías que se resumen en la Tabla 2.

Tabla 2. Descripción y diferencias significativas entre las metodologías conducidas por planes y metodologías ágiles.

Metodologías conducidas por planes	Metodologías ágiles
Conducidas por planes	Orientada al código (dado que la parte importante de la documentación es el código)
Planificación predictiva	Planificación adaptable
Resistencia a los cambios	Los cambios son bienvenidos
Orientada a los procesos	Orientadas a las personas
Negociación de contratos	Colaboración con el cliente
Se basa en los modelos genéricos de cascada, prototipado, DRA, Incremental, Espiral	El modelo de desarrollo es evolutivo, iterativo, y trabaja con timeboxed (las iteraciones poseen tiempo fijo) (p. 64)
Entregables fijos	Entregables evolutivos

Fuente: Propia basada en la tabla 4.1 de Pantaleo, G. & Rinaudo, L. (2015), Ingeniería de software, p. 55 y 64

1.1.4.5 Metodología ágil

En un mundo tan cambiante sujeto a muchas variables, es importante que la tecnología también sea cambiante, por lo que la metodología ágil propone una forma de desarrollo sin que la persona que desarrolle un sistema se frustre cuando le propongan un cambio o el cliente se decepcione cuando el sistema o aplicación deje de funcionar con el pasar del tiempo.

En la actualidad todos queremos optimizar tiempo y al disminuir la cantidad de documentación es evidente que el factor tiempo también es una ventaja que se debe considerar para el uso de metodologías ágiles.

Además, algo que hay que resaltar es que esta metodología promueve a principios básicos del Buen Vivir, demostrando una vez más que las personas somos más importantes que las máquinas, dando a entender que la relación entre desarrollador y cliente no debe ser hostil sino al contrario, entre mejor diálogo exista entre el desarrollador y el cliente cada versión del sistema o aplicación será funcional y adaptable.

Una acotación importante es la que se encuentra en Poppendieck, M., & Poppendieck, T. (2003). Lean Software Development: An Agile Toolkit

The development of Agile software requires innovation and responsiveness, based on generating and sharing knowledge within a development team and with the customer.

Agile software developers draw on the strengths of customers, users, and developers, finding just enough process to balance quality and agility.

Que traducido quiere decir:

El desarrollo de software ágil requiere innovación y capacidad de respuesta, sobre la base de generar y compartir conocimientos dentro de un equipo de desarrollo y con el cliente.

Los desarrolladores de software ágiles se basan en las fortalezas de los clientes, usuarios y desarrolladores, la búsqueda del proceso para equilibrar la calidad y agilidad.

1.2 Metodología Ágil Microsoft Solution Framework (MSF)

Una vez revisados los conceptos anteriores se llegó a la conclusión que la metodología óptima para el desarrollo del proyecto de tesis es la metodología ágil MSF, misma que será detallada a continuación junto con sus ventajas y desventajas.

Sin bien es cierto no existe una gran cantidad de documentación referente a la metodología MSF, en este documento, se muestra una recopilación de varios autores y aportes personales, para que la aplicación de metodologías ágiles sea un tema sencillo de comprender.

Microsoft en su página oficial [https://msdn.microsoft.com/es-es/library/jj161047\(v=vs.120\).aspx](https://msdn.microsoft.com/es-es/library/jj161047(v=vs.120).aspx) (Microsoft, 2013) establece que:

Microsoft Solutions Framework (MSF) es un enfoque personalizable para entregar con éxito soluciones tecnológicas de manera más rápida, con menos recursos humanos y menos riesgos, pero con resultados de más calidad. MSF ayuda a los equipos a enfrentarse directamente a las causas más habituales de fracaso de los proyectos tecnológicos y mejorar así las tasas de éxito, la calidad de las soluciones y el impacto comercial.

MSF se centra en (Microsoft, 2013):

- Alinear los objetivos de negocio y de tecnología
- Establecer de manera clara los objetivos, los roles y las responsabilidades
- Implementar un proceso iterativo controlado por hitos o puntos de control
- Controlar los riesgos de manera proactiva
- Responder con eficacia ante los cambios

La metodología Microsoft Solution Framework o como se la conoce comúnmente MSF, es considerada una metodología ágil por las siguientes razones:

- Simplifica la documentación, al generar documentación básica sin muchos requisitos y sin una estructura rigurosa.
- Promueve la comunicación constante entre el desarrollador y el cliente.
- Se adapta a los requerimientos cambiantes del cliente.

La metodología MSF es un ejemplo de que se puede combinar dos modelos de procesos de software, por esta razón se detalla algunas características que adopta esta metodología de los modelos en cascada e incremental.

Características del Modelo en Cascada

- Pasa de una fase a otra en cascada, es decir que cada fase no debe comenzar sino hasta que termine la fase previa.
- Proceso dirigido por un plan, es decir que se debe planear y programar todas las actividades del proceso, antes de comenzar a trabajar con ellas.

- El resultado de cada fase consiste en uno o más documentos que se autorizaron (firmaron).

Características del modelo incremental.

- A pesar de que no comienza una fase antes de terminar la anterior esta metodología permite la rápida retroalimentación a través de las actividades.
- El desarrollo de software incremental es una parte fundamental de los enfoques ágiles.
- Se desarrolla diversas versiones hasta producir un sistema adecuado.

Ventajas de usar los dos métodos

- El estado del proyecto se hace visible.
- Los entregables pueden asociarse a las fases, facilitando así la gestión.

En resumen la metodología MSF es del tipo de metodologías ágiles, flexible e interrelacionada a una serie de conceptos, modelos y mejores prácticas de uso que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos es decir que está enfocada a dirigir proyectos o soluciones de innovación, en ella no se detalla ni se hace énfasis de la organización ni el tamaño del equipo de desarrollo, está más bien centrada en la gestión y administración del proyecto para lograr el impacto deseado dejando en segundo plano las elecciones tecnológicas.

Involucra indudablemente la calidad ya que se quiere liberar una solución y para evitar que ésta tenga fallos o desperfectos, se propone seleccionar un grupo de prueba piloto el cual es una VERSION BETA y cumplido un tiempo de prueba ya es liberada la versión formal o VERSION ALFA en la cual está garantizada la calidad.

1.2.1 Principios fundamentales y actitudes en el contexto de MSF

Microsoft Solutions Framework (MSF) se asienta sobre una base de principios y actitudes que representan años de experiencia. Estos principios y actitudes, destilados en conceptos que se sostienen en los distintos modelos, procesos y disciplinas de MSF, son la base de MSF. Aunque son conceptos de sentido común, comprenderlos e implementarlos correctamente puede llegar a ser complicado. No obstante, una vez que se comprendan, el equipo podrá crear productos de calidad. (Microsoft, 2013)

Principios fundamentales

Los siguientes principios y conceptos de MSF sirven de guía al equipo de proyecto para entregar una solución de calidad. Cada miembro del equipo deberá comprender y aplicar estos principios en sus interacciones con otros miembros del equipo, con la organización y con las partes interesadas. MSF se basa en nueve principios fundamentales (Microsoft, 2013):

1. Fomentar una comunicación abierta. Para que el equipo sea eficaz y eficiente, tanto usted como su equipo deben compartir niveles de información apropiados entre los miembros del equipo y en toda la empresa. El equipo debe comprender la naturaleza de lo que se debe hacer y el modo en que se comunican los miembros del equipo y los contactos externos. Lo

difícil es determinar un nivel apropiado para cada relación y qué información se debe compartir.

2. Intentar lograr una visión compartida. El hecho de tener una visión compartida empodera a los miembros del equipo y les permite actuar con agilidad para poder tomar decisiones rápidas, pero bien fundadas con el objetivo de lograr una visión. Al tener una visión compartida, los miembros del equipo pueden ir satisfaciendo los requisitos a medida que se vayan detectando.
3. Empoderar a los miembros del equipo. Empoderar a los miembros del equipo no solo es una de las muchas maneras de sobrevivir en un entorno en constante cambio, sino que los miembros del equipo también aprenden a encontrar modos de alcanzar el éxito de manera creativa y a ayudarse unos a otros. Si no se permite a los miembros del equipo dar lo mejor de sí mismos, no solo disminuye su creatividad, sino que también pueden sufrir de baja moral y ser incapaces de contribuir a crear un equipo de alto rendimiento.
4. Establecer responsabilidades claras y compartidas. A menudo, los miembros del equipo empoderados se sienten más responsables de sus decisiones y están dispuestos a ser corresponsables de un proyecto. A mayor responsabilidad de los miembros del equipo, mayor calidad. Por ejemplo, si un miembro del equipo afirma que ha completado una tarea, pero se detecta que no tiene el nivel de calidad adecuado, ese miembro del equipo es responsable de resolver este problema de manera que la tarea completada tenga los niveles de calidad indicados. Si se fomenta el crecimiento positivo y la responsabilidad en lugar de castigar tales deslices, el miembro del equipo comparte la responsabilidad de la solución general y sus entregas. Esto fomenta la motivación entre los miembros más sólidos del equipo para ayudarse mutuamente a dar lo mejor de sí mismos.
 - a. Ofrecer valor incremental. Este principio tiene dos facetas:
 - b. Asegurarse de que lo que se entrega tiene un valor óptimo para las partes interesadas.
5. Determinar los incrementos óptimos en los que se entregará valor (o la "frecuencia de entrega").
6. Mantenerse ágil, esperar cambios y adaptarse a ellos. Como los cambios pueden darse a menudo y en el peor momento posible, disponer de una manera ágil de manejarlos ayuda a minimizar los trastornos habituales que provocan. Mantenerse ágil significa que una organización está preparada para los cambios y puede adaptarse y ajustarse sin contratiempos.
7. Invertir en la calidad. Muchas organizaciones adoptan el principio de calidad, a menudo con una definición bastante difusa, pero no saben cómo cuantificarla. La calidad es algo que se debe incorporar de manera proactiva al ciclo de vida de entrega de la solución y no es algo que aparezca de la nada.
8. Aprender de todas las experiencias. Si todos los niveles de una organización no aprenden de lo que funcionó y lo que no funcionó anteriormente, ¿cómo se puede esperar que mejoren la próxima vez? Los miembros del equipo deben comprender y darse cuenta de que el aprendizaje se da en todos los niveles:
 - a. A nivel de proyecto, como, por ejemplo, al perfeccionar un proceso válido para todo el proyecto

- b. A nivel individual, como, por ejemplo, al buscar la manera de interactuar mejor con otros miembros del equipo.
 - c. A nivel de la organización, como, por ejemplo, al ajustar las métricas de calidad que se recopilan para cada proyecto.
9. Colaborar con clientes internos y externos. Las probabilidades de éxito del proyecto aumentan cuando el cliente trabaja con el equipo del proyecto. Eso no quiere decir que los clientes tengan que hacer el trabajo de un equipo. Sin embargo, cuando los clientes colaboran estrechamente y de manera incremental con un equipo de entrega, la solución satisface mejor sus expectativas. Colaborar con los clientes es ventajoso para ambas partes, ya que ayuda a reducir la incertidumbre, reduce el tiempo necesario para resolver temas de requisitos y aumenta la comprensión por parte del equipo de las propuestas de valor de la solución por medio del contacto periódico.

Actitudes

Mientras que los principios fundamentales explicados más arriba sirven de guía para orientar a un equipo para lograr el éxito al máximo, las actitudes permiten orientar a los miembros del equipo como individuos para lograr el mayor éxito. Cada actitud ayuda a los miembros del equipo a hacer realidad la entrega de su solución concreta. Idealmente, los miembros del equipo acaban estando tan cómodos con estas actitudes que las usan tanto en el trabajo como en la vida personal. A continuación, se incluyen unas cuantas actitudes que debe internalizar cada miembro del equipo:

- Fomentar un equipo de iguales. Si su organización es capaz de encarnar los principios fundamentales de MSF, en especial el empoderamiento y la responsabilidad, ¿tiene sentido ejecutar un proyecto con una estructura jerárquica? Si todo el mundo comprende tanto la misión como sus objetivos para tener una visión compartida y sus roles y responsabilidades en la entrega de una solución, todos actúan como iguales y se les puede tratar por igual. Con esto no se propone ningún tipo de anarquía ni comité, sino que todos compartan la responsabilidad de entregar correctamente una solución. A nivel individual, cada rol es responsable de los aspectos que le corresponden en un proyecto y, en conjunto, es responsable del proyecto como un todo. Como verá, sigue habiendo un rol de Administrador de programas, pero ese rol se encarga de entregar un proyecto dentro del plazo establecido, pero no de administrar los miembros del equipo.
- Centrarse en el valor empresarial. El éxito se mide en términos de entrega de valor empresarial. Esto no solo significa entregar algo que los clientes necesitan, sino también entregar lo que los clientes desean y valoran. Para entregar valor, todos los integrantes de un equipo deben comprender qué es lo que los clientes consideran valioso. Si no se proporciona a los clientes valor empresarial, el proyecto peligra: corre el riesgo de desviarse de sus objetivos, de invertir mal el tiempo, el esfuerzo y el dinero, e incluso corre el riesgo de terminar cancelándose.
- Mantener la perspectiva de la solución. Debido al tamaño y la complejidad de la mayoría de los proyectos, cuando se visualiza una solución examinando sus partes procesables, a veces los miembros del equipo examinan con demasiado detenimiento pequeños detalles y olvidan tener en mente la solución final. Por eso se da tanta importancia al principio de tener una visión compartida. Cuando los miembros del equipo entregan sus partes, deben

tener en cuenta la misión, los objetivos y la visión generales de una solución. Con demasiada frecuencia, un equipo secundario optimiza su parte creyendo que actúa por el bien común, pero descubre que tiene que rehacer aspectos importantes para que vuelva a estar en sintonía con una solución: se enreda en los detalles y pierde de vista la solución.

- Enorgullecerse por el trabajo realizado. Un equipo no solo debe invertir en calidad, sino que sus miembros también deben darse cuenta de que la calidad es tan responsabilidad suya como de los otros miembros del equipo y no debe delegarse o pasarse a otra persona. La calidad no es sino responsabilidad de todos durante todo el ciclo de vida de entrega de la solución. Esta actitud no solo abarca el aumento de la calidad en las entregas de un miembro individual del equipo, sino que también abarca el aumento de la calidad de la ejecución de procesos y la gobernanza del proyecto. Esta actitud mental también anima a cada miembro del equipo a ampliar sus conocimientos sobre las habilidades necesarias para cumplir su misión general. Si observan su propia calidad y entregan el mejor trabajo posible, los miembros del equipo pueden facilitar mejoras continuas en el producto final.
- Aprender continuamente. A veces, no basta con enorgullecerse de sus habilidades y de las de su equipo para alcanzar el objetivo final. Los miembros del equipo deben aprender nuevas habilidades para ser mejores integrantes del equipo. Dado que la mayoría de los proyectos, equipos y entornos son únicos, cada proyecto presenta oportunidades de aprendizaje, experimentación e innovación de habilidades, procesos y procedimientos. Para sacar provecho de estas oportunidades, el aprendizaje y la adaptación deben ser continuos en todos los niveles de una organización y no limitarse únicamente a los miembros del equipo.
- Internalizar la Calidad de servicio. La Calidad de servicio (QoS) define las características operativas de una solución, como su nivel de disponibilidad esperada. Es fundamental que las partes interesadas y los miembros del equipo, no solo los arquitectos, comprendan la QoS y cómo el hecho de cumplirla puede afectar a las entregas. De lo contrario, es probable que las partes interesadas y los miembros del equipo hagan suposiciones implícitas sobre el comportamiento esperado de la solución. Como estas suposiciones raramente son las mismas, cada miembro del equipo debe tomar decisiones explícitas sobre el diseño desde el principio para garantizar que se satisface la QoS. De ese modo, las suposiciones implícitas se convierten en requisitos de QoS explícitos. Así, la QoS se diseña de manera explícita en la solución desde el principio y no se trata como algo a posteriori.
- Ser cívico. Desde el punto de vista del desarrollo de software, ser cívico significa ser digno de confianza, honrado, responsable y respetuoso en todos los aspectos del trabajo. Esto incluye, entre otros aspectos:
 - El modo en que interactúa con los otros miembros del equipo, con una organización y con las partes interesadas.
- La manera en que participa en un proyecto y contribuye a entregar una solución, incluido el hecho de ser el encargado de confianza de los recursos corporativos, los del proyecto y los recursos informáticos. Esto incluye compartir recursos, información y conocimientos de manera abierta y voluntaria. Las personas que son cívicas actúan por y para el bien común.
- Cumplir los compromisos. A pesar de los numerosos controles y contrapesos integrados, MSF se alimenta de la confianza y el empoderamiento que los miembros del equipo se

ganan, en parte, a base de cumplir sus compromisos. MSF establece un entorno en el que los miembros del equipo y las partes interesadas pueden confiar en que los otros miembros del equipo cumplirán lo que afirman que van a hacer. Como un proyecto es una colección de actividades interdependientes, cuando un miembro del equipo incumple sus compromisos, desequilibra y pone en peligro todo el proyecto.

1.2.2 Etapas y Modelo de gobernanza de MSF

Modelo de Gobernanza MSF

El modelo de gobernanza (llamado anteriormente modelo de procesos) está diseñado para proporcionar una guía adecuada a las personas apropiadas en el momento oportuno. Está estructurado para permitir a un equipo entregar partes fundamentales de una solución más rápido de lo que sería posible si se centrara primero en las características de mayor prioridad y pasara las menos importantes a versiones posteriores. El modelo está estructurado para ayudar a un equipo a llegar rápidamente a un consenso compartido sobre cómo entregar los distintos aspectos de una solución. El modelo de gobernanza es un componente flexible de MSF que se ha usado con éxito para mejorar el control del proyecto, minimizar los riesgos, mejorar la calidad de la solución y aumentar la velocidad de desarrollo. Como MSF es totalmente personalizable, cabe esperar que una organización adapte el modelo de gobernanza para que se ajuste a sus procesos de negocio y metodologías de entrega de soluciones existentes. (Microsoft, 2013)

El modelo de gobernanza de MSF combina la gobernanza del proyecto con la ejecución de procesos. La gobernanza del proyecto se centra en optimizar el proceso de entrega de la solución y el uso eficiente y eficaz de los recursos del proyecto. La ejecución de procesos se centra en la definición, la compilación y la implementación de una solución que satisfaga las necesidades y las expectativas de las partes interesadas. (Microsoft, 2013)

Son aspectos fundamentales del modelo de gobernanza de MSF la superposición de pistas de actividad, la sincronización de puntos de control y un enfoque incremental en la entrega de valor al cliente. (Microsoft, 2013)

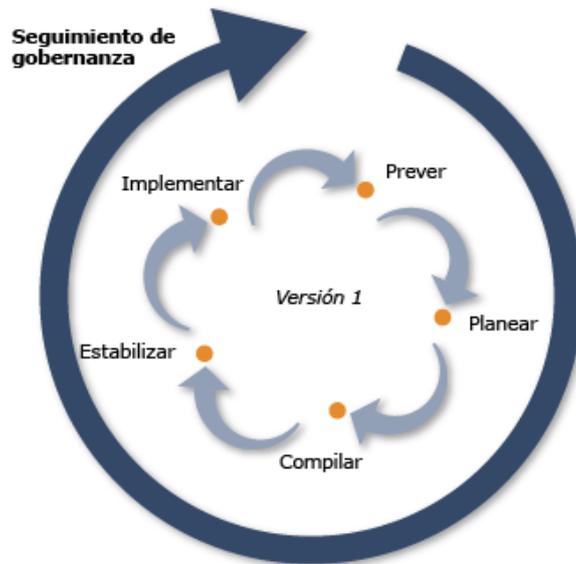


FIGURA 2. Diagrama de la pista de Gobernanza

Fuente: Página oficial de Microsoft, 2013

Pistas (Página Oficial de Microsoft, 2013)

El modelo de gobernanza de MSF facilita la gobernanza del proyecto y la ejecución de procesos por medio de pistas de actividad superpuestas. En un nivel, las pistas son agrupaciones superpuestas y coordinadas de ciertas actividades cuya finalidad es generar entregas relevantes de cada pista. No obstante, las pistas de MSF son más que eso; cada una tiene una misión distinta y representa un cambio en el ritmo y el foco de un proyecto. En las pistas se usan revisiones y puntos de sincronización denominados puntos de control (hablaremos de ellos a continuación) para poder determinar si se cumplen sus objetivos. Además, se usan puntos de control principales para cerrar cada pista, lo cual permite trasladar las responsabilidades de muchas actividades y anima al equipo a adoptar una nueva perspectiva más apropiada para los objetivos de la siguiente pista.

El modelo de gobernanza de MSF consta de cinco pistas de ejecución superpuestas y una pista de gobernanza persistente que abarca todas las pistas de ejecución.

Pista de gobernanza (Página oficial de Microsoft, 2013)

La pista de gobernanza se centra en equilibrar el uso eficiente y eficaz de los recursos del proyecto y en la entrega de una solución respetando un conjunto de restricciones del proyecto que pueden cambiar. Además, la pista de gobernanza propugna la mejora continua de los procesos.

Una buena gobernanza del proyecto proporciona supervisión, procesos, instrucciones y rigor suficientes para usar de manera eficiente y eficaz los recursos del proyecto, entregar una solución y gestionar decisiones de compensación, todo ello equilibrando el respeto de un conjunto de restricciones del proyecto que pueden cambiar. La pista de gobernanza de MSF se

esfuerzo por proporcionar y mejorar continuamente la buena gobernanza del proyecto. Consta de actividades discretas y persistentes a lo largo de un proyecto.

Los objetivos de la pista de gobernanza son:

- Guiar las actividades de ejecución para entregar una solución con resultados repetibles y confiables.
- Optimizar y mejorar continuamente el rendimiento y la capacidad del equipo, la calidad de las soluciones y la mejora de procesos
- Obtener la aprobación de:
 - Usuarios (la solución satisface sus necesidades y se puede usar con suficiente facilidad)
 - Operaciones (la solución está preparada para su implementación)
 - Cliente (el proyecto se ha completado)

Pistas de ejecución (Página oficial de Microsoft, 2013)

La ejecución de procesos es la secuencia detallada de pasos que sirven para definir, compilar e implementar una solución. Básicamente, las pistas de ejecución ayudan a un equipo a llegar a un acuerdo de alto nivel sobre lo que se imagina de cara al futuro y crear opciones de enfoque para hacer realidad esa visión (pista de visión); evaluar esas opciones y planear la opción seleccionada (pista de planeación); compilar la solución (pista de compilación); asegurarse de que la solución se entrega según lo esperado (pista de estabilización); y, por último, implementar la solución (pista de implementación).

Los objetivos de cada pista de ejecución son:

Visión

Desarrollar un entendimiento claro sobre lo que se necesita dentro del contexto de las restricciones del proyecto.

Reunir al equipo necesario para concebir soluciones con las opciones y los enfoques más adecuados para esas necesidades y que también satisfagan de manera óptima esas restricciones.

Planeación

Hacer evolucionar la solución conceptual hasta llegar a diseños y planes tangibles para que se pueda compilar en una pista de compilación.

Compilación

Compilar los aspectos de la solución de acuerdo con las entregas de la pista de planeación, como diseños, planes, programaciones y requisitos.

Estabilización

Mejorar la calidad de la solución para satisfacer los criterios de lanzamiento para la implementación en producción.

Validar que la solución satisfaga las necesidades y expectativas de las partes interesadas.

Validar la facilidad de uso de la solución desde la perspectiva de los usuarios.

Maximizar el éxito y minimizar los riesgos asociados con la implementación y las operaciones de la solución en los entornos de destino de la solución.

Implementación

Integrar una solución correctamente en producción dentro de los entornos designados.

Transferir la responsabilidad de la entrega restante de la solución de un equipo de proyecto a equipos de operaciones y soporte técnico tan pronto y de manera tan fluida como sea posible.

Puntos de control (Página oficial de Microsoft, 2013)

Los puntos de control, que son un tema central en MSF, se usan para planear y supervisar el progreso del proyecto e informar sobre la finalización de las entregas y actividades. Los puntos de control se usan para proporcionar oportunidades explícitas para que un equipo y los clientes reconfirmen el ámbito del proyecto o para ajustar el ámbito del proyecto a fin de reflejar los cambios en los requisitos de los clientes o el negocio o a fin de dar cabida a los riesgos y problemas que se podrían materializar en el transcurso de un proyecto. Los puntos de control se usan por muchos motivos, como:

Ayudar a sincronizar los elementos de trabajo.

Proporcionar visibilidad externa del progreso y la calidad.

Permitir correcciones a mediados del proyecto.

Centrar las revisiones en los objetivos y las entregas.

Proporcionar puntos de aprobación del trabajo antes de avanzar.

MSF distingue entre dos tipos de puntos de control: puntos de control principales y puntos de control provisionales. Los puntos de control principales marcan la finalización de actividades y entregas principales, incluido el final de las actividades planeadas para una pista. El equipo define los puntos de control provisionales para indicar el progreso en una pista y segmentar actividades de gran envergadura en piezas viables.

Enfoque iterativo (Página oficial de Microsoft, 2013)

Una solución no proporciona valor empresarial hasta que se implementa en producción y se usa de manera eficaz. Por este motivo, el ciclo de vida del modelo de gobernanza de MSF incluye el desarrollo y la implementación incrementales de una solución en producción, lo que garantiza que se haga realidad el valor empresarial, además de la visión y los objetivos estratégicos globales de un equipo. Para garantizar que los proyectos cumplen lo que promete la tecnología, MSF combina una representación de negocio multidimensional en un equipo con el foco explícito en el impacto en el negocio a lo largo de todo el proceso.

La práctica del desarrollo iterativo es un tema constante en MSF. Los documentos, diseños, planes y otras entregas se desarrollan de manera iterativa. Como cabe esperar, el modelo de gobierno de MSF tiene un enfoque iterativo.

Etapas de la Metodología Microsoft Solution Framework (MSF)

Las etapas de la metodología MSF son (Microsoft, 2013):

- Visión
- Planeación
- Compilación
- Estabilización
- Implementación

Las fases de la metodología Microsoft Solution Framework tienen una amplia relación con las principales etapas del modelo en cascada, lo cual se puede ver reflejado en las actividades fundamentales del desarrollo que se detalla en la Figura 3.

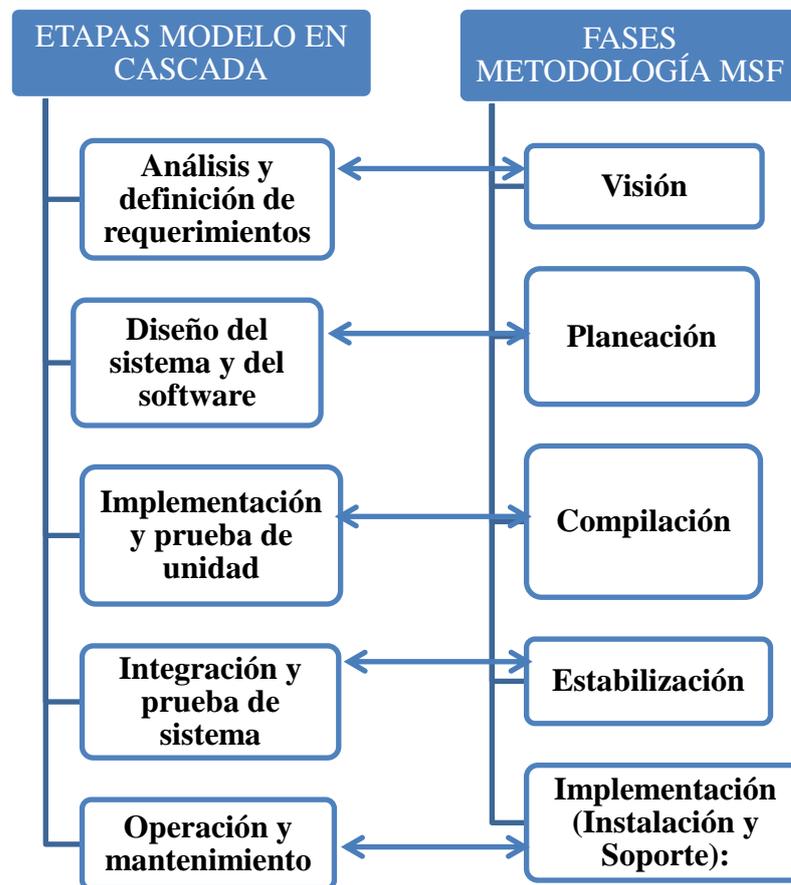


FIGURA 3. Relación entre etapas del modelo en cascada y fases de la metodología MSF

Fuente: Propia

Las etapas del modelo en cascada se combinan con el modelo incremental y dan como resultado que este ciclo se puede llevar a cabo de forma iterativa, de manera que cuando liberamos una solución podemos iniciar nuevamente la metodología para darle más funcionalidad.

Es importante indicar que una de las ventajas de la metodología MSF es que, en todas las etapas de la metodología, el usuario o cliente puede constatar los progresos o avances en el desarrollo del software mediante documentos, archivos, etc., que se los conoce como “ENTREGABLES” de cada fase.

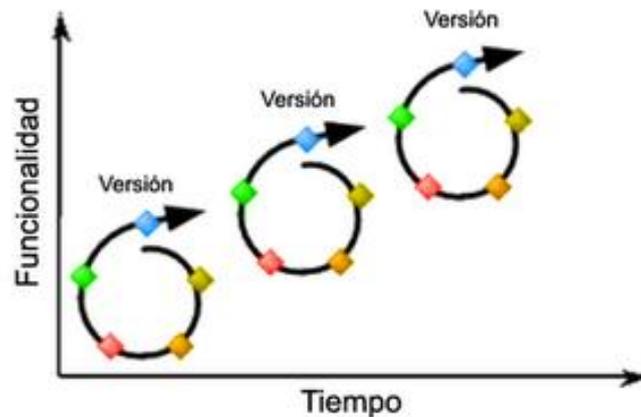


FIGURA 4. Representación de una característica del desarrollo incremental en la metodología MSF.

Fuente: Imágenes Google

Los entregables son fundamentales en esta metodología, con el fin de que el cliente pueda revisar constantemente el avance en el desarrollo de su requerimiento y pueda realizar las modificaciones, actualizaciones o cambios durante cualquier etapa de esta metodología. Es decir que, si el desarrollador opta por esta metodología, debe tener la posibilidad de estar en contacto permanentemente con el cliente.

En la figura 5, se detalla algunos ejemplos de entregables de cada fase de acuerdo con el modelo en cascada:

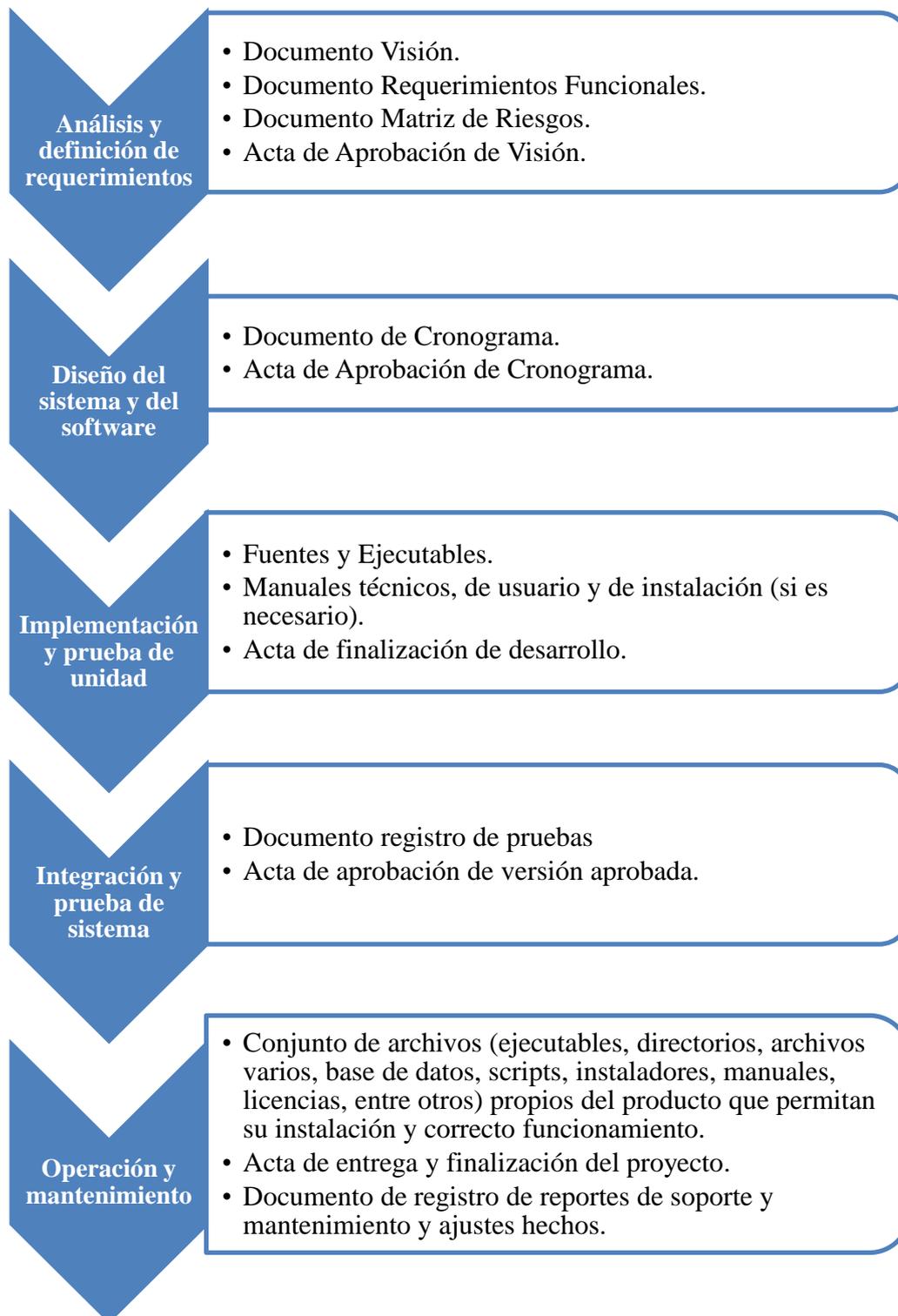


FIGURA 5. Detalle de “entregables” de la metodología MSF de acuerdo al modelo en cascada.

Fuente: Propia

A continuación, se hace una breve explicación y se da pautas para desarrollar cada etapa de la Metodología MSF de una forma clara y sencilla.

Es importante mencionar que esta sección contiene algunas acotaciones de acuerdo con el conocimiento adquirido en la aplicación de la metodología para el desarrollo de este tema de tesis

➤ **Visión:** Según Pantaleo, G., & Rinaudo, L. 2015, p. 309 -312

Es muy probable que la mayor parte de los desarrolladores nunca hayan preguntado a los involucrados por parte de la empresa cliente de un proyecto de desarrollo ¿por qué trabajaremos en este proyecto? ¿Por qué construiremos este sistema? ¿Para qué lo construiremos?

Sin embargo, estas son las respuestas que se necesitan a la hora de establecer una visión para el proyecto. Si un proyecto de desarrollo se avanza sin responder estas preguntas, si se asume que la idea del sistema a construir nació por que sí, por generación espontánea, seguramente este tendrá graves problemas en su evolución.

Estas preguntas ayudan a definir una visión de la cual derivarán objetivos y prioridades y será esta visión a la que se recurrirá cuando se pierda el rumbo en cualquier momento del ciclo de vida del proyecto.

A efectos de construir esta visión se debe analizar y documentar los aspectos siguientes:

- Requerimientos del negocio.
- Estado Actual.
- Oportunidades de negocio.
- Objetivos y criterios de éxito.
- Problemas y necesidades de los usuarios.
- Riesgos.
- Características salientes de la solución.
- Supuestos y dependencias.
- Alcance inicial y posterior.
- Involucrados.
- Prioridades del proyecto.
- Ambiente operacional.

Esta visión compartida volcada en un documento de visión debe ser acordada por todos los involucrados al momento de acordar el desarrollo del proyecto.

Requerimientos y prioridades.

Entre los criterios utilizados para este fin se analizaron los siguientes: (p. 311)

- Valoración Financiera
- Deseo de los miembros de la empresa cliente
- Valor agregado al negocio
- Aporte de conocimientos del negocio
- Riesgo de desarrollo.

Estimaciones de alcance y esfuerzo

Independientemente del método que se utilice para hacer estimaciones, se deben tener en cuenta los siguientes aspectos.

- Las estimaciones no son ni objetivos ni compromisos de negocio.
- Las estimaciones se hacen con información reducida.
- Las estimaciones como tal tienen límites de incertidumbre.
- Las estimaciones deben ser presentadas como estimaciones.
- Las estimaciones deben ser construidas por todos los involucrados.
- Las estimaciones teóricas deben ser ajustadas a la realidad del proyecto.
- Las estimaciones de organizaciones que conservan sus grupos de desarrollo formados y aprenden de sus errores son cada vez más precisas.
- Las estimaciones cerradas de proyectos vendidos con alcance fijado, costo fijado y planificación (tiempo) fijada conducen a proyectos fallidos.
- Las estimaciones realizadas para un proyecto trabajado en forma iterativa (alcance variable de acuerdo con las prioridades, costo fijo y planificación fija) deben ser comunicadas y explicadas a la organización cliente como tal, de lo contrario se presentarán conflictos cuyo desenlace puede ser la cancelación del proyecto.

Nota: Una mayor explicación de los efectos de construir una visión, requerimientos – prioridades y de las estimaciones de alcance y esfuerzo, se encuentra en el libro Ingeniería de Software de Pantaleo, G., & Rinaudo, L., 2015, p. 309-312. Este libro también cuenta con una guía más completa de Relevamiento preliminar que se encuentra en las páginas desde 323 hasta 327

En resumen, la etapa visión consiste en tener una idea clara del proyecto que se va a realizar, esto se logra con la comunicación con el cliente que es una de las actividades que promueve las metodologías ágiles.

La clave es definir requerimientos y objetivos iniciales que sean claros tanto como para el cliente como para el desarrollador, no fijar requerimientos demasiado extensos y que estos deben tener la capacidad de adaptarse en el tiempo.

- **Planeación:** Según Pantaleo, G., & Rinaudo, L. 2015, lo primero que se debe hacer es planificar los siguientes aspectos: (p.103)
 - Ambiente de desarrollo.
 - Recursos humanos.
 - Asignación de roles a los miembros del grupo del proyecto.
 - Alcance y prioridades de los requerimientos.
 - Arquitectura del sistema en desarrollo.
 - Entregables del proyecto.

Si bien es una problemática conocida, cuando se perciben retrasos en las planificaciones se sigue sugiriendo sumar desarrolladores a los proyectos. Una planificación no puede reducirse más allá de un 75% a partir del agregado de miembros al grupo de desarrollo. Existe una

llamada “zona imposible” de evolución de los proyectos de desarrollo cuyo límite está dado por el número mencionado. (Pantaleo, G., & Rinaudo, L. 2015, p. 312)

El ser humano es un ente que trata de planificar toda actividad, pero existen ciertos factores externos e internos que impiden el cumplimiento en 100% de la planificación, como desarrolladores no debe haber frustración cuando una planificación falla o se retrasa, la comunicación en este tipo de metodologías con el cliente es fundamental, tal como afirma Pantaleo, G. (2016), en las claves para trabajar con metodologías iterativas como MSF, las cuales se detallan a continuación:

- Tecnología que permita hacer crecer el sistema en desarrollo con cada iteración de forma que el producto construido esté clausurado ante cambios.
- Acuerdo en la priorización de las características del producto a construir entre la organización desarrolladora y el cliente.
- Participación del cliente en los proyectos de desarrollo (p. 108).

Como estimar tiempos y planificar, pues por experiencia estimar tiempos es uno de los retos que como desarrolladores debemos enfrentar, cuando la experiencia en desarrollo es muy corta, las planificaciones iniciales no suelen ser muy precisas, pero conforme se aprende de los errores se adquiere destreza en planificar y estimar tiempos.

MSF al ser una metodología que tiene como características ser una metodología incremental es decir que se puede sacar una versión mejora al producto inicial tiene algunas ventajas sin embargo es importante tener en cuenta los siguientes factores que se deben considerar mientras se planifica una N versión del producto inicial.

Según Pantaleo, G. en su libro Calidad en el Desarrollo de Software (2016, p. 59-21) afirma que: Los planes son importantes, la planificación continua es imprescindible (p. 106) y por esto establece algunos factores críticos y de riesgo, así como también factores generales de éxito que pueden ser considerados en la fase planeación mismos que a continuación se describen:

Factores críticos y de riesgo:

Algunos factores críticos a tener en cuenta en la planificación (planeación) de un proceso de mejoras son (p. 59-60):

1. Historia de procesos previos de mejora.
2. Recursos financieros disponibles.
3. Recurso humano dedicados al proceso.
4. Capacidad de ingeniería de software.
5. Disponibilidad de soporte técnico.
6. Obligaciones contractuales.
7. Alcance de la intervención.
8. Cultura de la organización.
9. Estándares de la industria y la organización.
10. Comprensión y apoyo de todos los niveles de gerencia y desarrollo.
11. Presión política de la corporación.

12. Objetivos de negocio.

13. Visión.

Son importante porque:

1. Procesos previos fracasados demandarán mayor esfuerzo.
2. Recursos no suficientes demandarán mayor esfuerzo de los miembros o la factibilidad del proyecto.
3. Sin la dedicación de sector de recursos humanos, que también son miembros de la empresa, no es posible implementar las mejoras.
4. Cuanto menor sea el conocimiento de los miembros, mayor será la capacitación necesaria.
5. Sin soporte técnico para las tareas del proyecto no es posible su desarrollo.
6. Condicionan la factibilidad de realización de pilotos en los proyectos.
7. Debe ser claramente establecido el alcance del proyecto para no generar confusión.
8. La cultura debe ser tenida en cuenta ya que condiciona el proceso de cambio.
9. Los estándares deben ser tenidos en cuenta y pueden condicionar el proyecto.
10. Sin el apoyo gerencial se aconseja no comenzar el proyecto.
11. La presión de la corporación hay que tenerla en cuenta y contrarrestarla con planificaciones realistas, nunca ceder ante ella.
12. Los objetivos del negocio deben ser tenidos en cuenta siempre, se derivan de la visión.
13. En el grupo de partida, respecto de la visión, debe responderse la pregunta: ¿Por qué? Para descubrirla y darle forma.

Factores Generales de Éxito

Entre los factores que la mayor parte de las veces garantizan el éxito se pueden enumerar (p. 60-61):

1. Establecer expectativas realistas.
2. Asegurar el apoyo de los cuadros gerenciales.
3. Identificar las necesidades gerenciales, sus expectativas y la comprensión del problema.
4. Crear un grupo de Ingeniería de Procesos de Software (con un "campeón" del cambio).
5. Comenzar mejoras poco después de una evaluación.
6. Capacitar a todos los usuarios de los nuevos procesos.
7. Gestión de la dimensión humana del proceso de mejoras.
8. Selección cuidadosa de proyectos piloto.
9. Hacer auditorías de proceso durante la implementación.
10. Hacer encuestas de efectividad de los equipos de trabajo.
11. Comenzar el proceso de mejoras con los ajustes gerenciales más altos (dar ejemplo).
12. Tener acompañamiento de expertos en cambio organizacional.
13. Mantener relación cercana entre mejoras y objetivos de negocios.
14. Adoptar un vocabulario compartido.

Porque:

1. Evita planificaciones fuera del alcance de la organización.

2. Dota al proyecto de compromiso y soporte.
3. Facilita el desarrollo y evita conflictos por afectación de intereses.
4. En este ámbito es donde se generan los cambios.
5. Evita la generación de procesos virtuales, como se describió en la sección anterior.
6. Es esencial como parte del proceso de comunicación.
7. Evita el rechazo generado por los cambios.
8. Para no afectar proyectos críticos y convertir a los pilotos en responsables de los retrasos y fallas de los proyectos.
9. Para corregir desviaciones de la planificación y mantenerlo vivo.
10. Para generar realimentación y conocer la opinión de todos los involucrados.
11. Es parte de la generación de un ambiente propicio de confianza en el que se apoyarán los mentores a la hora de acompañar los miembros.
12. Las empresas en general no cuentan con recursos especializados en los temas necesarios por una cuestión de costos.
13. Para lograr también la aprobación y conformidad de la gerencia de marketing y comercial al ver cumplidos sus objetivos.
14. Para generar un ambiente en el cual el proceso de cambio tenga identidad.

Como acotación adicional, si bien es cierto, los factores de riesgo y factores generales de éxito antes mencionados se aplican en su mayoría a empresas medianas y grandes, existen algunos factores que se pueden aplicar para emprendimientos o desarrollos individuales.

➤ **Compilación:**

Pantaleo, G., & Rinaudo, L (2015), afirma que:

Cuando el proyecto ya fue acordado se comienza con el desarrollo de las tareas que fueron planificadas. A partir de este momento del ciclo de vida debe realizarse un seguimiento que consiste en monitorear cada una de las actividades y controlar su evolución tomando las acciones que se decidan necesarias. (p. 330)

En estas revisiones se comparan los resultados obtenidos con los esperados para: (p. 332).

- Los requerimientos implementados.
- Los tiempos invertidos.
- Las pruebas de aceptación.
- La evolución de los riesgos.

En esta fase se empieza la ejecución de lo planificado, razón por la cual, planificar es muy importante.

La recomendación es que si no se tiene levantado los procesos que se van a transformar en un sistema, primero se realice el levantamiento de procesos y procedimientos antes de empezar a desarrollar. Una vez que el cliente esté de acuerdo con los procesos, se continúe con la estructuración y desarrollo del proyecto.

➤ **Estabilización:**

Para Pantaleo, G., & Rinaudo, L (2015), las pruebas de software son uno de los ingredientes que ayudarán a garantizar la calidad de los productos generados. (p. 286)

El autor menciona algunos aspectos que deben ser considerados en la fase de pruebas, que se enlistan a continuación:

- Los recursos humanos necesarios para la realización de las tareas planificadas
- La infraestructura de hardware y software; es decir que debe incluir la definición del ambiente en el cual se realizarán las pruebas.
- La estrategia de diseño y ejecución de las pruebas.
- Prioridades.
- Objetivos.
- Acciones a tomar ante las fallas detectadas.
- Modelo de informe de resultados.
- Criterios de comienzo y finalización.
- Coordinación de tiempos con el resto de las tareas del proyecto.
- Métricas que recolectar.
- La asignación de roles a efectos de determinar quiénes son los responsables del diseño, construcción, ejecución y análisis de los resultados de las pruebas.

Es importante resaltar lo que afirma Pantaleo, G. (2016) acerca de realizar pruebas:

Un documento terminado no implica ningún logro en el proyecto ni evidencia ningún grado de avance si no está acordado y aprobado por todos los involucrados. (p. 116)

Es decir que un proyecto finalizó cuando el sistema se encuentre implementado, funcione sin errores de acuerdo a los requerimientos iniciales y el cliente este de acuerdo con el producto final.

Para probar el resultado del promedio de notas se usará el método de la Caja Blanca

Caja blanca se denomina cuando el método de prueba se diseña a partir de la estructura y lógica del software bajo prueba (p. 267)

➤ **Implementación:**

Según Pantaleo, G., & Rinaudo, L (2015) afirman que:

El esfuerzo de los desarrolladores por implementar determinada funcionalidad no se acaba con una bella interfaz de usuario ni con un flujo de control bien logrado, ni con la persistencia precisa de los datos manipulados, ni con la aislación de las transacciones asociadas; sino con todo esto terminado, compilado, probado, por un conjunto de test y subido al repositorio del proyecto. (p. 286)

Esto se puede interpretar como que el proyecto finaliza cuando ha pasado todas las pruebas y se encuentran funcionando de acuerdo con las especificaciones solicitadas.

1.3 Estado del Arte

Con el fin de realizar un análisis desde los fundamentos y principios de las metodologías se va a realizar un comparativo entre las Metodologías Ágiles y Metodologías Conducidas por Planes.

La mayoría del contenido de esta sección se obtuvo del libro Ingeniería del Software de Pantaleo, G., & Rinaudo (2015), por considerar importante y de gran relevancia la información que contiene.

Historia según Pantaleo, G., & Rinaudo, L. (2015)

Nota: En este libro se toma como ejemplo de metodologías conducidas por planes al Proceso Unificado de Desarrollo de Software.

En la década de 1970 existían en la industria del desarrollo de software dos estándares instalados, uno era el lenguaje “C” de programación y el otro el Desarrollo Estructurado de Sistemas, como paradigma de las metodologías llamadas *trabajo en Cascada*. Estas metodologías, consistían en el desarrollo del proceso en una cascada de tareas. Las dificultades que presentaban estas metodologías radicaban en: (p. 72)

1. Falta de una relación estrecha con los requerimientos en todo el ciclo de vida del proyecto de desarrollo.
2. Falta de una especificación estable de la arquitectura del software.
3. Una dinámica de trabajo con poca posibilidad de realimentación entre las distintas etapas del proceso de desarrollo.

Tanto las metodologías en Cascada como su transición al Proceso Unificado de Desarrollo de Software son un claro ejemplo de la estrecha relación entre los conceptos técnicos y los relacionados con la forma de trabajo. Afirmamos esto debido a que las falencias mencionadas fueron resueltas en parte por la introducción de un nuevo paradigma basado en la programación orientada a objetos y los lenguajes que la implementaron. (p. 72)

En los años de 1995 a 2003 existían en la industria del desarrollo de software dos estándares instalados, uno eran los lenguajes “C++/Java” de programación y el otro era el Proceso Unificado de Desarrollo de Software como paradigma de las metodologías orientadas a objetos. Estas metodologías, consistían en una evolución en forma iterativa e incremental del proceso, soportado por la arquitectura y conducido por los casos de uso. (p. 92)

Las dificultades que presentaron la utilización de estas metodologías radicaron en: (p. 92)

1. Carencia de un criterio para la adaptación de la forma de trabajo a los diferentes tipos de proyectos.
2. Generación excesiva de la documentación propuesta para los modelos de desarrollo.
3. Inconsistencias en la gestión de los proyectos con requerimientos, tiempo y presupuesto fijados.

Hacia el año 2001 se fundó el movimiento Ágil de metodologías de desarrollo a partir de la definición de un conjunto de principios comunes expresados en el famoso manifiesto (p. 92), citado en este capítulo.

De todos los referentes que contribuyeron a esta fundación, hubo algunos que aportaron algunas de las metodologías conocidas de la época.

Introducción

Con el tiempo, este aumento en la predictibilidad de los proyectos los hizo más estables. Pero esta estabilidad no era la dinámica apropiada cuando surgió el internet como plataforma de una comunicación, oferta y consumo de servicios. Los tiempos de salida al mercado de los productos se acortaron y la estabilidad mencionada era percibida como un obstáculo; fue así como aparecieron las metodologías ágiles. (Pantaleo, G., & Rinauldo, 2015, p.9)

Fue necesario ponderar la creatividad y velocidad de adaptación a los nuevos escenarios por sobre el control y la previsibilidad en los proyectos. Este fenómeno de estabilidad de procesos versus creatividad ya se había percibido décadas antes entre el software Factory de Japón y la industria del software norteamericana. La calidad de excelencia de la producción oriental se logró a expensas de la ausencia de nuevos productos los cuales eran ofrecidos por las empresas de EE.UU. que trabajan en un ambiente mucho menos controlado. (Pantaleo, G., & Rinauldo, 2015, p.9)

Como muchas veces a lo largo de la historia de esta industria se instaló hace algunos años una idea de moda. Esta vez consiste en pensar que cuanto menos estructuradas las empresas sean, mejor estarán posicionadas para enfrentar cambios y ganar nuevos negocios. (Pantaleo, G., & Rinauldo, 2015, p.9)

La razón de esto sigue siendo explicada por W. Deming y su frase acerca de cuál es el objetivo de las empresas, perdurar en el tiempo y generar empleo. (Pantaleo, G., & Rinauldo, 2015, p.9)

El desafío de hoy día con respecto a garantizar la calidad en las organizaciones dedicadas al desarrollo de software es organizarse para que sus proyectos sean ordenados y predecibles; aunque deben a su vez tener la capacidad de dejar de lados estos procesos rápidamente para reorganizarse y adaptarse a los cambios. Por eso es muy importante lograr madurez en estos procesos de manera de seguirlos cómodamente para que no se constituyan en una carga que aumente los costos e impida contar con agilidad para cambiar. (Pantaleo, G., & Rinauldo, 2015, p. 10)

Metodologías Ágiles vs Metodologías Conducidas por Planes

En esta sección se encuentra una recopilación de las principales ideas de Pantaleo, G. & Rinauldo, L., (2015), junto con la de otros autores que faciliten establecer comparaciones entre los dos tipos de metodologías.

A continuación, se enlistan algunos aspectos de las metodologías que según Pantaleo, G. & Rinaudo, L., (2015) no se ocupan, pero que las metodologías ágiles, en su mayoría las han incluido en sus definiciones (p. 29):

- Las personas se comunican hablando cara a cara y a partir de diálogos con preguntas y respuestas.
- Las personas no son consistentes en su forma de pensar continuamente en el tiempo.
- Las personas son variables, día a día y de lugar en lugar.
- Las personas generalmente desean sentirse integrados a las empresas, son curiosos, tomando iniciativas y están dispuestos a realizar esfuerzos para que los proyectos tengan éxito.
- Las personas necesitan tiempo para pensar y oportunidad para comunicarse.
- Las personas trabajan mejor a partir de ejemplos.
- Las personas prefieren fallar en forma conservativa que tener éxito asumiendo riesgos.
- La gente tiene rechazo a los cambios.
- Personalidades individuales dominan los grupos.
- Carácter personal que afecta la habilidad para desarrollar tareas específicas.

Pantaleo, G. & Rinaudo, L., (2015) también afirma que en general las descripciones de las distintas metodologías son:

- Largas y complejas.
- Mezclan descripción de procesos con productos generados.
- No tratan pequeñas reglas que producen grandes efectos.
- No tratan de las personas y su ambiente de trabajo, sino de roles. (p. 29)

Sin estas condiciones ninguna metodología es asumida como la mejor por todos, ya que están ausentes las bases que aseguran al ser humano desarrollarse como profesional. (Pantaleo, G. & Rinaudo, L., 2015, p. 29)

Según este autor, manifiesta que las metodologías basadas en planes generaban retrasos en los proyectos y sobre presupuesto, debido a que se enfatizó en la realización de las diferentes actividades en lugar de trabajar para lograr objetivos. En resumen, las metodologías ágiles antepusieron a las personas antes que cumplir con la extensa documentación. Es decir, “Software funcionando sobre documentación extensa” (p. 101 y 102).

Otras de las ventajas de las metodologías ágiles es que no requiere que se realicen obligatoriamente diagramas o casos de uso (p. 115).

La dinámica de los procesos en cascada es más lenta que los iterativos, por lo tanto, es posible una planificación y un seguimiento con más información y en forma más abierta. El plan es único y la gestión, aún de los niveles gerenciales, posee información detallada de la evolución de los proyectos. (Pantaleo, G. & Rinaudo, L., 2015, p. 29)

En los procesos iterativos, en cada iteración se decide la continuidad del proyecto. La dinámica es más veloz, se realizan todas las tareas para generar un entregable. Esta dinámica es posible si se organizan los grupos de desarrollo, se les otorga independencia a partir de una gestión con

formación relevante no detallada y se deja al líder del proyecto la conducción con decisión de primera línea. (p. 30).

Parte de la independencia del grupo de desarrollo es la libertad para fijar las políticas de su funcionamiento. Esto es en gran medida garantizado a partir de facilitar y generar las buenas prácticas que se listan y se describen a continuación (p. 30):

Cultura organizacional: debe ser consistente la forma iterativa de desarrollo con el modo en que los proyectos son vendidos.

Grupo de desarrollo: de parte del grupo de desarrollo se espera actitud y compromiso para con el proyecto.

Organización cliente: dispuesta a establecer acuerdos centrados en la confianza que genera el conocimiento de la forma de trabajo y el trabajo compartido más que en conflicto de intereses y multas por incumplimientos.

Si bien son factibles diferentes formas de trabajo con distintas dinámicas, el desarrollo de software ha obtenido una madurez que indica a la iterativa como la forma más adecuada. Los ingredientes anteriores son los necesarios para que esta forma de trabajo se convierta en una propiedad deseable. (Pantaleo, G. & Rinaudo, L., 2015, p. 30)

La necesidad de la incorporación del cliente al proyecto tiene dos razones importantes. Una es el rol de usuario experto que define y aclara aspectos del sistema en desarrollo, la otra tiene que ver con la visibilidad de la evolución y las decisiones que se toman con el fin de que hay en todo momento un conocimiento detallado del estado del proyecto. Esta propiedad busca facilitar la relación con los involucrados por parte del cliente, aspecto de fundamental importancia. (Pantaleo, G. & Rinaudo, L., 2015, p. 32)

Cuando se habla de que una metodología es ágil lo que se está expresando es que la disciplina de los miembros del grupo se enfoca en un conjunto de prácticas definidas lo que determina la disciplina del proyecto. Con relación a su objetivo se dice que consiste en maximizar la cantidad de requerimientos implementados en un tiempo y presupuesto fijados. (Pantaleo, G. & Rinaudo, L., 2015, p. 92)

La agilidad y adaptabilidad de estas metodologías requieren que el cliente pueda comprender y aceptar lo que esto implica. Es decir que es necesario un tipo de relación distinta con el cliente. Los contratos fijos que generalmente exigen los clientes no encajan en este modelo. Esto no significa que no se pueda fijar un presupuesto para el producto por adelantado. Lo que queremos decir es que no se puede fijar tiempo, precio y alcance. La manera ágil usual es fijar tiempo y precio, y permitir que el alcance varíe de manera controlada. (Pantaleo, G. & Rinaudo, L., 2015, p. 64)

La modificación del alcance le brinda la posibilidad al cliente de definir durante la vida del proyecto sus requerimientos, entonces puede apreciar a lo largo del proceso que elementos le dan mayor valor a su negocio. Generalmente estos rasgos son difíciles de apreciar al principio del proyecto. (Pantaleo, G. & Rinaudo, L., 2015, p. 64)

Metodologías Conducidas por los Planes (Centra en los costos) que planifican los proyectos a partir de estimaciones de alcance y esfuerzo. A partir de estas se estima el tiempo de duración del proyecto y el presupuesto. La falencia de esto radica en que el alcance es el componente de mayor incertidumbre y todo este proceso se basa en él. (Pantaleo, G. & Rinaudo, L., 2015, p. 67)

Metodologías Ágiles (Centrada en la Producción) que planifica sus proyectos en base a estimaciones preliminares de la misma forma que las anteriores por no se atan sus componentes a estas. A diferencia de las anteriores se fija un tiempo de duración del proyecto y un presupuesto, y se comprometen a implementar la mayor cantidad de requerimientos a partir de prioridades establecidas por el negocio. La clave del cambio está en subordinar el elemento de mayor incertidumbre a los de menor. (Pantaleo, G. & Rinaudo, L., 2015, p.32)

Diferencia de aplicabilidad

Es necesario examinar la aplicabilidad de los métodos ágiles y considerar el espectro de dominios de problemas en los que podrían ser aplicables y aquellos en los que otros métodos serían mejores. (Pantaleo, G. & Rinaudo, L., 2015, p. 68)

La evidencia de los datos relevados estadísticamente indica que deben tenerse en cuenta las siguientes variables y ayuda a sacar algunas conclusiones las cuales fueron condensada en la tabla 3 que sigue: (Pantaleo, G. & Rinaudo, L., 2015, p. 68)

Tabla 3. Guía de uso de los métodos ágiles y conducidos por los planes

<i>Según sea</i>		Ágil	Conducida por Planes
Tipo de proyecto	Maduro		√
	Inmaduro	√	
Tipo de sistema	Monolítico		√
	Divisible	√	
Madurez del proceso de desarrollo	Maduro		√
	En inducción	√	
Madurez del modelo de desarrollo	Definido		√
	Indefinido	√	
Capacidad / experiencia de desarrolladores	Autónomos	√	
	Dependientes		√

Fuente: (Pantaleo, G. & Rinaudo, L., 2015, p. 68)

Se observa que las metodologías ágiles son más adecuadas cuando los dominios son inmaduros o desconocidos ya que permiten una mayor adaptación a los cambios, cuando es factible el desarrollo incremental y se cuenta con desarrolladores con buena formación y experiencia ya que esto permite organizar el grupo de proyecto en un trabajo autónomo, Cuando se cuenta con procesos y modelos de desarrollo definidos y maduros se está en condiciones de afrontar los proyectos utilizando otras metodologías con el previo análisis del resto de las variables. (Pantaleo, G. & Rinaudo, L., 2015, p. 68).

Conclusiones (Panteleo, G. & Rinaudo, L., 2015, p. 69):

Los contextos en los que sería más apropiado utilizar metodologías conducidas a los planes:

- Un equipo de trabajo de gran magnitud.
- Un alcance o contrato fijo.

La previsibilidad es una característica deseada, pero no siempre puede darse en los proyectos de desarrollo de software. Creer que tenemos previsibilidad y luego esto no se cumple puede provocar importantes desvíos sobre el proyecto. En contraposición a un contexto de previsibilidad mencionamos una serie de situaciones en las cuales sería conveniente aplicar metodologías ágiles.

- Requerimientos inciertos o volátiles.
- Desarrolladores responsables y motivados.
- Cliente comprometido, es decir que entienda y se involucre.

Es conveniente adaptar los modelos en función de la realidad de la empresa o equipo de trabajo que la aplicación de una metodología aplicada al pie de la letra o, peor aún, la usencia de una forma de trabajo.

A las conclusiones anteriores agregamos las siguientes:

1. Las metodologías ágiles no presuponen la no existencia de planes, por el contrario, promueven el ejercicio de planificación continua.
2. Las metodologías ágiles no se ocupan del modelo de desarrollo de software en los aspectos técnicos (modelo de requerimientos, modelo de dominio, modelo de diseño); los grupos que elijan una metodología ágil deben definir su propio modelo de desarrollo.
3. La forma en que se venden y comparan los proyectos de desarrollo de software son determinantes al momento de utilizar una metodología ágil o una conducida por los planes.

Como conclusiones personales se puede agregar que las metodologías ágiles están diseñadas para trabajar de una forma ordenada y sistemática, pero de una forma sencilla.

En un mundo moderno el tiempo es uno de los recursos más valiosos que tiene el ser humano, todo es rápido, comida rápida, consulta a cuentas rápida, acceso a dinero rápido, comunicación rápida, por que insistir en algo que requiere tiempo.

Si bien es cierto la metodología tradicional busca tener documentado la mayor cantidad del desarrollo del proyecto para corregir posibles errores que se generen en el transcurso de la vida útil de la aplicación, pero se debe ser realistas, en la actualidad, los sistemas no son eternos, nuevas versiones de software con más bondades se desarrollan constantemente, entonces surge la pregunta: ¿Es necesario documentar tanto algo que voy a reemplazar?

Las empresas de hoy en día buscan soluciones rápidas, no necesariamente bien documentadas, es por lo que las metodologías ágiles cada vez tienen mayor aceptación en el mundo de la industria para desarrollo de software.

Luego de haber realizado un comparativo entre metodologías tradicionales o basadas en planes y metodologías ágiles, también se revisó algunas de las características de las principales metodologías ágiles como XP y SCRUM, con el fin de establecer diferencias y similitudes entre las metodologías antes mencionadas y la metodología MSF, esta información se encuentra en el siguiente cuadro resumen:

Tabla 4. Comparativo entre metodologías ágiles MSF, XP, SCRUM

	Microsoft Solution Framework (MSF)	XP	SCRUM
Descripción General	Metodología flexible e Interrelacionada enfocada a desarrollo de proyectos o soluciones de innovación	Conjunto de prácticas que configuran aspectos de desarrollo	Marco de gestión de los proyectos de desarrollo
Número de desarrolladores	Individual o en Equipos	Programación por pares	Desarrollo de proyectos en Equipo (entre 3 y 7 personas)
Diseño	Basado en el modelo incremental y en cascada	Incremental y conducido por pruebas	Iterativo
Entregables	Diversos entregables en todo el desarrollo del proyecto	Entregables frecuentes	Al inicio y al final de cada Sprint
Fundamentos	Modelo de Gobernanza (llamado Modelo de procesos)	Valores, prácticas y principios	Componentes de SCRUM
Pruebas	Realiza pruebas a las versiones del proyecto	Testeo continuo	Al final del Sprint
Relación con el cliente	Comunicación continua con el cliente	Parte del Equipo	El cliente es parte del equipo (Product Owner)
Planificación	Cronograma que tenga definido responsables	Planificación semanal y cuatrimestres	Planificación del Proyecto Y Planificación de una iteración (sprint)
Reuniones	Cuando sea estrictamente necesario	Semanal y cuatrimestral	Planeación, revisión, retrospectiva y diaria

Fuente: Propia, referencia Pantaleo y Rinaudo (2015)

1.4 Impactos de la Metodología

Impacto Ambiental

Es una metodología amigable con el medio ambiente porque de forma indirecta contribuye a disminuir el impacto ambiental que genera un proyecto con otras metodologías que exigen una cantidad exagerada de documentación que muy probablemente no será útil en su totalidad, al disminuir la cantidad de documentación se tiene una relación directamente proporcional a la disminución de papel y útiles de oficina.

1.5 Impactos del Sistema Informático

En el caso de que el sistema se ha implementado como parte del sistema integrado de la Universidad Técnica del Norte los impactos son los siguientes:

Impacto Ambiental

Una de las razones para que se implementen sistemas informáticos en la actualidad es el poder disminuir el uso de papel en las instituciones, esto se debe a una política gubernamental de “Cero Papeles”.

Es importante destacar que el sistema de seguimiento de tesis contribuye al cumplimiento de este objetivo, esto se debe a que muchos de los procedimientos que intervienen en el proceso de titulación se realizan de forma impresa y gracias a la sistematización del proceso ya no será necesario.

Aunque la aplicación desarrollada está orientada a la disminución del uso de papeles, cabe destacar que el sistema también genera reportes que le permite al usuario certificar y validar la información ingresada en el sistema.

Como es de conocimiento general la disminución del consumo de papel, contribuye a disminuir la tala de árboles y como consecuencia logra que el hábitat de muchas especies no desaparezca.

Impacto Institucional.

La optimización de tiempo y recursos logra que se optimice el cumplimiento de las actividades de todos los usuarios involucrados en el proceso de titulación, lo cual le permita al usuario emplear el tiempo de mejor manera y disminuir actividades rutinarias.

Otros de los beneficios de una aplicación es tener acceso a la información histórica, esta información es necesaria para procesos de acreditación, realizar estadísticas y otros tipos de análisis que son de vital importancia para la mejora continua de las instituciones educativas.

Gracias al sistema las autoridades de la facultad FICA pueden revisar y dar seguimiento al proceso de Titulación de los estudiantes de la facultad, para la toma de decisiones y realizar plan de mejoras.

Los docentes podrán tener un mejor control de los estados de las tesis que están supervisando, esto conlleva a un mejor seguimiento del proceso.

Impacto Estudiantil

Cuando se habla de optimizar tiempos y recursos dentro de una institución educativa, los principales beneficiados son los estudiantes.

El tener acceso a la información le otorga al estudiante, tener una gran herramienta, que le permita dar seguimiento al cumplimiento de tiempos establecidos dentro del proceso de titulación.

Uno de los principales objetivos de la aplicación es lograr que el estudiante tenga una mejor comunicación e interacción con los docentes y personal administrativo y viceversa, logrando que el proceso de titulación sea un trámite transparente y óptimo para la institución.

CAPITULO II

2.1 Visión

En esta etapa se van a detallar los requerimientos del cliente, es importante que la información sea revisada y tenga la aceptación del cliente, debido a que este es el punto de partida para el desarrollo y la implementación del sistema.

Con el fin de que este documento sea más didáctico el documento visión se dividió en 5 partes: Introducción, Posicionamiento, Descripción de participantes en el proyecto y usuarios, Descripción Global del Producto y Matriz de Riesgos.

2.1.1 Introducción

Propósito

El propósito de este documento es recoger, analizar y definir las necesidades de alto nivel y las especificaciones del sistema informático para el proceso de titulación de la FICA. El documento se centra en la funcionalidad requerida por los usuarios finales.

Esta funcionalidad se basa principalmente en la optimización de tiempo y recursos de todos los usuarios involucrados en el proceso de titulación de los estudiantes de la FICA.

Situación actual

El área de concesión de la Universidad Técnica del Norte está en constante crecimiento, cada vez es más importante dar prioridad a mejorar la agilidad y tiempo de respuesta en los procesos, para lograr que el usuario final, es decir el estudiante aproveche el tiempo al máximo en cualquier trámite o fase del proceso de titulación.

Como en cualquier institución la toma de decisiones es fundamental para continuar o realizar cambios en los procesos, pero el requisito inicial es tener acceso a la información, lo cual resulta complicado si existen fallas de comunicación entre procesos que estén relacionados entre sí.

La Universidad Técnica del Norte no ha escatimado recursos para que el conocimiento y la tecnología estén estrechamente ligadas, pero aún existen procesos que no han sido optimizados.

La Facultad de Ingeniería en Ciencias Aplicadas por tener carreras tecnológicas, es donde se genera la materia prima para el desarrollo tecnológico, es una razón primordial para que los procesos estén automatizados en su totalidad.

Alcance

El sistema permitirá a los usuarios dar seguimiento a las actividades en las que intervengan dentro del proceso de titulación.

Los usuarios podrán modificar, ingresar y consultar información de acuerdo con el tipo de usuario y de acuerdo con los requerimientos del proceso

Definiciones, Acrónimos, y Abreviaciones

MSF: Son las siglas de Microsoft Solution Framework, metodología aplicada para el desarrollo de este proyecto.

FICA: Son las siglas de Facultad de Ingeniería en Ciencias Aplicadas. Es la facultad que sirve como prototipo para la implementación del sistema de titulación.

UTN: Siglas de Universidad Técnica del Norte.

OC4J: Oracle Container four Java

JDK: Java Development Kit, en español Kit de desarrollo para aplicaciones JAVA

JDBC: Java Database Connectivity, es una interface de acceso a bases de datos estándar SQL

CA: Consejo Académico

CD: Consejo Directivo

2.1.2 Posicionamiento

Oportunidad de Negocio

En caso de ser implementado, este sistema puede permitir a la facultad sistematizar el control de todas las etapas del proceso de titulación (seguimiento a temas de anteproyectos, seguimiento de cumplimiento de tiempos en desarrollo de tesis, número de tesis por docente, número de tesis por periodo académico, número de graduados, etc.), lo cual mejorará índices de eficiencia y calidad, otorgándole a la FICA una ventaja competitiva frente a otras universidades.

El promover el cumplimiento de índices de evaluación, lleva a la FICA a la excelencia académica y por ende a un proceso de acreditación óptimo y con menor riesgos.

Los estudiantes que se encuentren en el desarrollo de tesis enfrentarían menos trabas administrativas para la obtención del título, y en tal virtud la FICA, tendrá un mayor número de profesionales graduados.

Definición del problema

Tabla 5. Sentencia para definición del problema

El problema de	Controlar las etapas del proceso de titulación. Gestionar el número de tesis por docente. Optimizar tiempos de revisión de tesis. Controlar el tiempo de cumplimiento para la ejecución de tesis.
Afecta a	Decano Subdecano Directores de Carrera Secretario Abogado FICA Docentes Estudiantes Personal Administrativo
El impacto asociado es	Almacenar toda la información referente a los anteproyectos y tesis que se desarrollan en la FICA.
Una solución adecuada sería	Sistematizar el proceso de titulación y la mayoría de sus etapas, usando la base de datos, red local y la página de la UTN y generar una aplicación amigable para el usuario final, para que cada usuario tenga acceso a la información necesaria con el fin de dar cumplimiento a las actividades asignadas.

Fuente: Propia

Descripción del Producto

Tabla 6. Descripción del sistema para seguimiento al proceso de titulación de la CISIC.

Para	Decano, Subdecano, Directores de Carrera, Secretario Abogado FICA, Docentes, Estudiantes, Personal Administrativo
Quienes	Controlan cada una de las etapas del proceso de titulación y/o registran información relacionada con el proceso.
El nombre del producto	Es una aplicación o herramienta software.
Que	Almacena la información necesaria para gestionar el proceso de titulación de la FICA
No como	En la actualidad que se usa de forma parcial un sistema y de forma impresa.
Nuestro producto	Permite administrar y gestionar las distintas etapas del proceso de titulación de la FICA mediante una interfaz gráfica sencilla y amigable. Además, proporciona un acceso rápido y actualizado a la información desde cualquier punto que tenga acceso a la base de datos.

Fuente: Propia

2.1.3 Descripción de participantes en el proyecto y usuarios

Para proveer de un sistema efectivo que cumpla con los requisitos iniciales del usuario, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los usuarios del sistema y asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que éstos perciben para enfocar la solución propuesta hacia ellos.

Resumen de participantes en el proyecto

Tabla 7. Descripción y responsabilidades de cada integrante del equipo de desarrollo.

Nombre	Descripción	Responsabilidades
Ing. Mauricio Rea	Tutor del Proyecto de Tesis y supervisor	Supervisar y revisar la aplicación desarrollada con el de que cumpla con los requisitos solicitados por el cliente.
Srta. Cecilia Carrillo	Tesista CISIC y desarrollador	Desarrollar la aplicación que cumpla con los requisitos de los diferentes tipos de usuario.

Fuente: Propia

Nota: En esta sección se describe todos los participantes que intervienen en el desarrollo del sistema, por ejemplo, Analistas de Requerimientos, Desarrolladores, etc., Esta información es importante debido a que se establecen las responsabilidades de cada miembro del equipo.

Resumen de Usuarios

Tabla 8. Cuadro Resumen de usuarios del sistema de acuerdo con el cargo UTN.

Nombre	Descripción	Cargos UTN
Administrativo	Responsables de dar seguimiento y realizar estadísticas tanto de los estudiantes como de los docentes que se encuentran involucrados en el proceso de titulación, para toma de decisiones	Decano, subdecano y coordinadores de carrera, secretario abogado CISIC
Auxiliar	Encargado de entregar reportes e información relacionada al proceso de titulación. También registra información del consejo académico	Secretaria de la carrera
Docente	Revisa avances del desarrollo de la tesis, emite correcciones, evalúa y da seguimiento al cumplimiento de tiempos de cada una de las etapas del proceso de titulación	Tutores de tesis y/o docentes
Estudiante	Registra y revisa información del anteproyecto y tesis en desarrollo, realiza correcciones.	Estudiante

Fuente: Propia

Entorno de usuario

Por ser una herramienta de estudio la aplicación se encuentra en una Máquina Virtual con el fin de que los estudiantes puedan revisar el código fuente.

Perfil de Participantes en el proyecto

Tabla 9. Perfil del Director de Tesis

Representante	Ing. Mauricio Rea
Descripción	Director de Tesis
Tipo	Tutor asignado e Ingeniero en Sistemas
Responsabilidades	Encargado del asesoramiento técnico y teórico en el proceso de desarrollo del sistema informático
Grado de participación	Revisión, seguimiento y aprobación del desarrollo del trabajo de grado.

Fuente: Propia

Tabla 10. Perfil de Tesista

Representante	Srta. Cecilia Carrillo
Descripción	Tesista
Tipo	Estudiante
Responsabilidades	Recopilación los requerimientos del cliente, realizar procesos y procedimientos, desarrollar e implementar el sistema. Documentación del trabajo de grado
Grado de participación	Desarrollo del sistema

Fuente: Propia

Nota: Esta sección del documento visión sirve como información para el cliente y personal administrativo, con el fin de que el cliente tenga una idea del equipo que está desarrollando el aplicativo.

Perfiles de usuario y responsabilidades.

Docente

Tabla 11. Perfil de usuario docente

Representante	Docentes
Descripción	Docentes UTN que tengan asignada la materia TESIS I o sean Tutores de Tesis
Tipo	Ingeniero en Sistemas
Responsabilidades	Responsable de orientar al estudiante en el proceso de titulación y elaboración de tesis Sugerir correcciones. Revisar y aprobar anteproyectos o tesis antes de ser aprobados por el Consejo Académico y Directivo.
Grado de participación	Revisión, seguimiento y aprobación de anteproyecto/Tesis.

Fuente: Propia

Administrativo

Tabla 12. Perfil de usuario administrativo

Representante	Coordinadores, Decano y Subdecano
Descripción	Autoridades de la Facultad y Coordinadores de Carrera
Tipo	Personal Administrativo
Responsabilidades	Acceder a reportes para toma de decisiones
Grado de participación	Revisión de reportes

Fuente: Propia

Estudiante

Tabla 13. Perfil de usuario estudiante

Representante	Estudiante
Descripción	Estudiantes que tengan aprobado más del 75% de la malla curricular, estén matriculados en las materias TESIS I, TESIS II o estén en proceso de desarrollo de tesis.
Tipo	Estudiante
Responsabilidades	Realizar y defender el anteproyecto. Escoger Tutor de tesis. Desarrollar la Tesis. Realizar correcciones sugeridas por el tutor de tesis y docente de TESIS I Defender la Tesis.
Grado de participación	Desarrollo total del trabajo de grado.

Fuente: Propia

Auxiliar

Tabla 14. Perfil de usuario auxiliar

Representante	Secretarías de Carrera & Secretario Abogado de la FacultadSTK1 Tuto
Descripción	Personal administrativo que realiza funciones de secretariado en las unidades académicas & Persona Jurídica delegada para velar por el cumplimiento de la normativa legal en cada Facultad
Tipo	Personal Administrativo
Responsabilidades	Revisión de carpetas o expedientes del estudiante. Ser la secretaria de actas del Consejo Académico. & Revisar cumplimiento de tiempos y normativa legal referente al proceso de titulación. Registrar resoluciones del Consejo Directivo. Realiza Actas de Grado Asesora en el proceso de legalización de los Títulos de Graduación.
Grado de participación	Registro de Información

Fuente: Propia

2.1.4 Diagrama de frecuencia del sistema

En la figura 6, se muestra un diagrama de frecuencia de la navegabilidad entre pantallas del aplicativo para realizar seguimiento al proceso de titulación de los estudiantes de la CISIC.

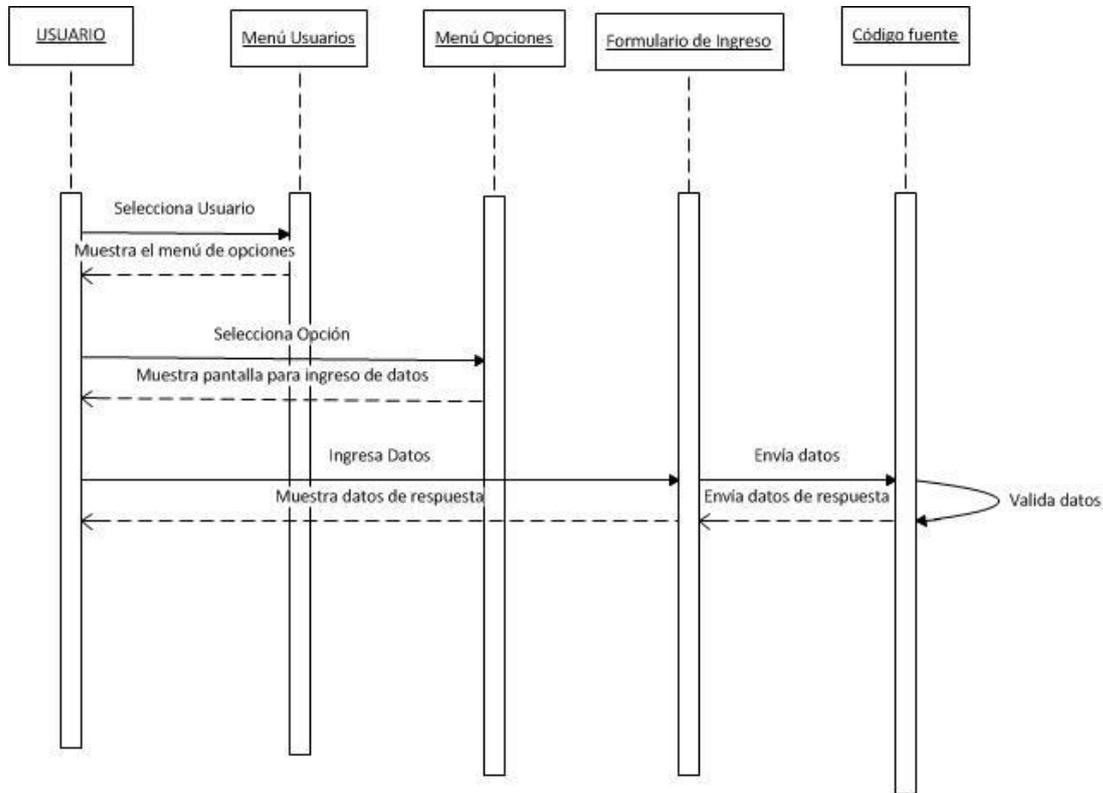


FIGURA 6. Diagrama de frecuencia del sistema de seguimiento para la titulación de los estudiantes de la CISIC.

Fuente: Propia

2.1.5 Descripción global del producto

Perspectiva del producto

El producto que se va a desarrollar es un sistema para la Facultad de Ingeniería en Ciencias Aplicadas (FICA), con el fin de generar una herramienta de carácter educativo que le permita al estudiante tener una guía para aplicar metodologías ágiles en el desarrollo de un producto. Este sistema tiene la característica de agilizar y mejorar el seguimiento de los temas de tesis que aún no estén ejecutados en su totalidad o que estén en la etapa de anteproyecto. Las áreas que se van a tratar por el sistema son las unidades académicas de la FICA.

Resumen de características

En el caso de que este sistema se integre a los servidores de la Universidad Técnica del Norte puede brindar los siguientes beneficios:

Tabla 15. Beneficios del sistema.

Beneficio del cliente	Características del sistema que aportan al beneficio
Mayor agilidad en el control de seguimiento de tesis	Aplicación desarrollada en Oracle Forms
Facilidad en el acceso a la información en tiempo real	Base de datos centralizada con la información de todo el personal.
Mejor distribución de las tesis asignadas a cada docente.	Sistema automatizado para la asignación de docentes para cada tesis.
Posibilidad de registrar la aprobación, anulación y las correcciones de los anteproyectos	Sistema automatizado para el registro de resoluciones del consejo académico.
Automatización del cálculo de los promedios involucrados en el proceso de titulación.	Base de datos con la información histórica del estudiante.
Agilidad en la entrega de certificaciones y reportes.	Base de datos con la información histórica del estudiante.
Acceso a la información para la toma de decisiones.	Reportes de la aplicación.

Fuente: Propia

Costo y precio del desarrollo de la tesis

A continuación, se realiza un cálculo aproximado del proyecto de tesis:

Costos de Hardware

DESCRIPCIÓN	COSTO \$
Computador	1500.00
Impresora y Copiadora HP	250.00
Total de Hardware	\$1750,00

Costos de Software

DESCRIPCIÓN	COSTO \$
Internet	19.90*4=79.60
Servidor de Base de Datos	0.00
Servidor de Aplicaciones	0.00
Total de Software	\$79.6

Materiales de oficina

DESCRIPCIÓN	COSTO \$
Copias (documentos, libros)	100.00
DVD's, esferos	20.00
Anillados	300.00
Total de Materiales de oficina	\$420,00

Costo salarial del desarrollador

\$2813.36

Total Inversión

\$ 5062.96

2.1.6 Matriz de Riesgos

El aplicativo al ser una herramienta de estudio se encuentra en una máquina virtual, razón por la cual no presenta riesgos reales.

Con el fin de que los estudiantes tengan noción de como armar una matriz de riesgos, se va a tomar como caso práctico, que el sistema se pone en producción como parte del sistema integrado de la Universidad Técnica del Norte.

Es importante mencionar que los sistemas informáticos están desarrollados para mejorar el estilo de vida y el cumplimiento de las funciones asignadas a los usuarios, pero esto no quiere decir que no existe un riesgo de que el sistema o la aplicación fallen.

El objetivo de diseñar una matriz de riesgos es determinar las acciones correctivas y preventivas

Para tener presentes los riesgos externos o internos que puede tener un aplicativo se muestra el siguiente ejemplo:

Tabla 16. Matriz de riesgos

Riesgo	Probabilidad	Consecuencias
Caída o daño en la red	Possible	Moderadas
Daño en el sistema integrado UTN	Poco Probable	Menores
Modificación o destrucción de la información confidencial	Poco Probable	Catastróficas
Uso inadecuado o daño de los equipos del personal administrativo	Muy Probable	Despreciable
Daño de los servidores o base de datos	Raro	Mayores
Ataques de virus informáticos a los servidores.	Poco Probable	Mayores
Falta de disponibilidad de la aplicación	Poco Probable	Menores
Mal uso de la aplicación	Possible	Moderadas
Acceso de personal no autorizado	Raro	Catastrófico

Fuente: Propia

		PROBABILIDAD				
		Raro	Poco probable	Posible	Muy probable	Casi seguro
CONSECUENCIAS	Despreciable	Bajo	Bajo	Bajo	Medio	Medio
	Menores	Bajo	Bajo	Medio	Medio	Medio
	Moderadas	Medio	Medio	Medio	Alto	Alto
	Mayores	Medio	Medio	Alto	Alto	Muy alto
	Catastróficas	Medio	Alto	Alto	Muy alto	Muy alto

FIGURA 7. Matriz para análisis cualitativo de riesgos

Fuente: Carmona M. A. (2013), Blog Corporativo, España, IAT Innovación y Tecnología, <http://www.iat.es/2013/10/gestion-riesgo-clave-excelencia-empresarial-iv/>

De acuerdo con la Figura 7 y a la matriz de riesgos (tabla 16), se puede determinar que los riesgos más sensibles son:

- ✚ Modificación o destrucción de la información confidencial.
- ✚ Daño de los servidores o base de datos.
- ✚ Ataques de virus informáticos.
- ✚ Acceso de personal no autorizado

Acciones correctivas y preventivas para la disminución de riesgos.

De acuerdo con el caso práctico, los riesgos a disminuir de mayor prioridad e impacto tienen relación con la seguridad, por este motivo se debe hacer énfasis en que la aplicación sea más segura para todos los usuarios.

La parte de respaldos e información histórica tiene un riesgo alto, si ocurre un daño en el servidor o en la base de datos, las consecuencias pueden ser significativas, este riesgo se debe al uso de equipos que son susceptibles a daño como cualquier equipo, pero cabe destacar que la Universidad Técnica del Norte dispone de equipos de alta tecnología, lo cual disminuye este riesgo.

La mejor acción preventiva de todo sistema que administre información sensible es el correcto uso de contraseñas o claves de acceso por parte de los usuarios y administradores y tener respaldo en tiempo real para evitar que la información histórica pueda ser dañada, modificada o eliminada.

2.2 Planeación

Todo proyecto debe ser planificado para que la ejecución de este sea exitosa.

Parte de planificar es establecer tiempos, en esta etapa es preciso tener la confianza de comunicar al cliente que debe haber un margen de error en el cumplimiento de dichos tiempos, esto se debe a que puede haber situaciones externas que contribuyan al retraso o agilización de la ejecución del proyecto.

Este margen de error no debe ser mayor a una prolongación de tiempo mayor al 10 por ciento del tiempo total establecido inicialmente, para evitar inconformidad o reclamos por parte del cliente.

El cumplimiento de tiempos o ejecutar el proyecto en menor tiempo de lo establecido puede demostrar que hubo una buena planificación y organización, sin embargo, la metodología MSF, determina que lo importante es que el cliente esté satisfecho con el producto terminado.

Cronograma

Este proyecto tiene un solo desarrollador, de existir más de una persona en el desarrollo de un producto se puede agregar la columna responsable, con el fin de que quién este como coordinador del proyecto pueda dar seguimientos a si se está cumpliendo con los tiempos previstos y quién está a cargo de cada actividad.

Tabla 17. Cronograma de desarrollo del sistema

#	Objetivos	Actividad	Nro. Semanas	Mes			
				1	2	3	4
1	Recopilar y seleccionar la información que se vaya a registrar en el sistema mediante la interacción con los futuros usuarios.	1.1 Entrevistar a los posibles usuarios del sistema.	1	X			
		1.2 Clasificar y organizar la información recolectada.	1	X			
		1.3 Definir los requerimientos iniciales del sistema.	2	X			
2	Definir procesos que permitan optimizar el proceso de titulación de la CISIC.	2.1 Hacer un previo análisis de los elementos que intervienen en la fase de seguimiento de tesis.	1		X		
		2.2 Definir los procesos partiendo de la información recopilada.	1		X		
		2.3 Ordenar y agrupar los procesos definidos para su posterior sistematización.	2		X		

#	Objetivos	Actividad	Nro. Semanas	Mes			
3	Diseñar e implementar el sistema con los requerimientos seleccionados y clasificados.	3.1 Seleccionar la base de datos que se va a utilizar para almacenar la información del sistema.	0,5			X	
		3.2 Diseñar la estructura de la base de datos.	0,5			X	
		3.3 Escoger las herramientas tecnológicas que permitan la comunicación del usuario con la base de datos y que mejoren la presentación general del sistema.	0,5			X	
		3.4 Diseñar e implementar el sistema en su fase de mejoras.	1			X	
		3.5 Definir los requerimientos finales del sistema.	0,5			X	
		3.6 Presentar el sistema informático a los usuarios con los requerimientos acordados.	1			X	
4	Probar e implementar el sistema.	4.1 Pruebas en el sistema realizadas por el usuario.	1				X
		4.2 Recolectar comentarios y posibles correcciones al sistema, emitidas por el usuario.	1				X
		4.3 Realizar las correcciones correspondientes y hacer pruebas finales.	1				X
		4.4 Entregar el sistema funcionando de la manera óptima.	1				X

Fuente: Propia

2.3 Compilación

Antes de empezar el desarrollo del sistema es importante el levantamiento de procesos y procedimientos si es que no existe un levantamiento previo por parte del cliente.

Para esta tesis se realizó el levantamiento de los procesos que intervienen en la titulación de un estudiante de la CISIC.

Proceso de Titulación

El diagrama de flujo del proceso de “Investigación – Titulación Profesional” se encuentra en el ANEXO 1, en virtud de que el diagrama se puede visualizar de forma más clara.

Procedimiento de Seguimiento de Tesis

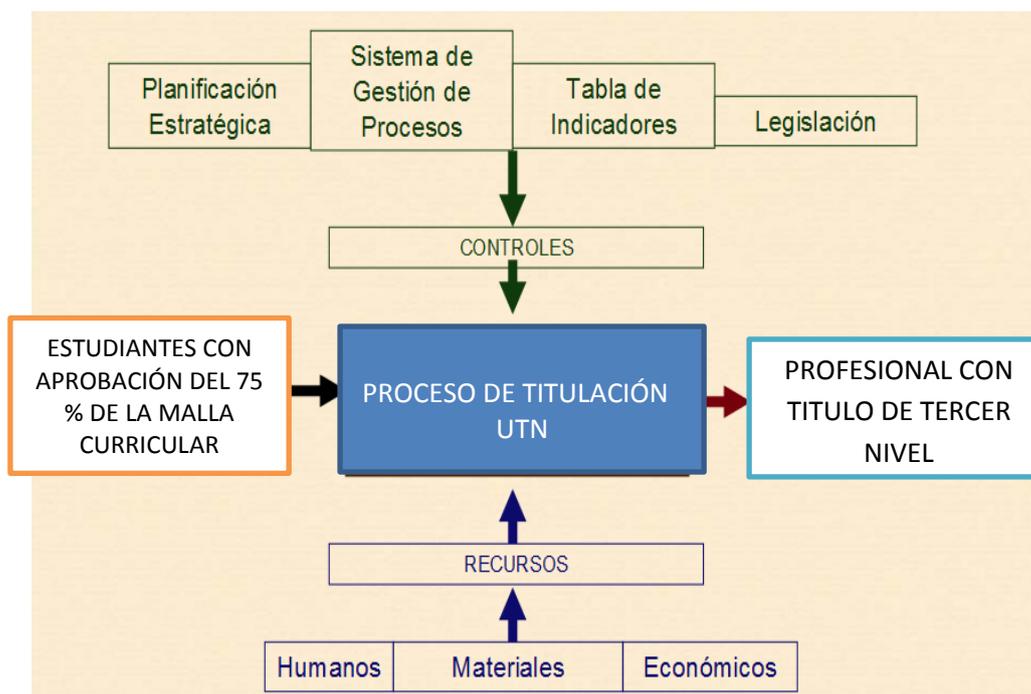


FIGURA 8. Proceso de Titulación de la Universidad Técnica del Norte

Fuente: Propia

A continuación, se detalla los subprocesos que intervienen en el proceso de titulación de la CISIC:

Tabla 18. Subproceso: Aprobación de anteproyecto

PROCESO: TITULACIÓN DE LA CISIC			
SUBPROCESO: Aprobación de Anteproyecto			
Objetivo: Aprobar la materia de Tesis I y aprobación del anteproyecto			
Nro.	Actividades	Responsables	Duración
1	Matricularse en la materia de Tesis I	Estudiante	Periodo de Matrículas
2	Definición del tema de tesis	Estudiante / Docente de Tesis I	3 a 4 semanas
3	Desarrollo del anteproyecto y revisiones periódicas	Estudiante / Docente de Tesis I	3 meses
4	Selección y aceptación del director o tutor de tesis	Estudiante	1 semana
5	Revisión, corrección y aprobación del último borrador de anteproyecto de tesis	Docente de Tesis I /Director de Tesis	1 semana
6	Entrega de 5 copias y el original del anteproyecto en Decanato	Estudiante	1 día
7	Enviar copias al Consejo Académico	Decanato	2 a 3 días

PROCESO: TITULACIÓN DE LA CISIC			
SUBPROCESO: Aprobación de Anteproyecto			
Objetivo: Aprobar la materia de Tesis I y aprobación del anteproyecto			
Nro.	Actividades	Responsables	Duración
8	Aprobación Preliminar de Anteproyecto	Consejo Académico	Acorde planificación de actividades y sesiones del Consejo Académico
9	Fijar fecha para defensa de anteproyecto	Consejo Académico	1 a 2 semanas posterior a aprobación preliminar
10	Defensa de Anteproyecto	Estudiante /Consejo Académico	1 día
11	Sugerir la aprobación de Anteproyecto	Consejo Académico	2 a 3 días
12	Entregar una copia del anteproyecto en CD	Estudiante	1 día
13	Aprobación Final de Anteproyecto	Consejo Directivo	Acorde planificación de actividades y sesiones del Consejo Directivo
14	Emisión y entrega de resolución de tesis aprobada al Docente de Tesis I	Consejo Directivo / Estudiante	1 semana
15	Asentamiento de Notas de la materia de Tesis I	Docente de Tesis I	Periodo para asentamiento de notas

Fuente: Propia

Tabla 19. Subproceso: Desarrollo y aprobación de tesis.

PROCESO: TITULACIÓN DE LA CISIC			
SUBPROCESO: Desarrollo y Aprobación de Tesis			
Objetivo: Aprobar la materia de Tesis II y aprobación de tesis			
Nro.	Actividades	Responsables	Duración
1	Matricularse en la materia de Tesis II	Estudiante	Periodo de Matrículas
2	Desarrollo de tesis	Estudiante	Tiempo establecido en el cronograma del anteproyecto
3	Revisiones y correcciones periódicas	Director de Tesis / Docente de Tesis II	Mensualmente

PROCESO: TITULACIÓN DE LA CISIC			
SUBPROCESO: Desarrollo y Aprobación de Tesis			
Objetivo: Aprobar la materia de Tesis II y aprobación de tesis			
Nro.	Actividades	Responsables	Duración
4	Verificar que la tesis este en formato APA 6	Director de Tesis / Docente de Tesis II	1 semana
5	Revisión del último borrador de Tesis	Director de Tesis / Docente de Tesis II	1 mes
6	Asentamiento de 1era nota de Tesis	Director de Tesis	1 semana
7	Entregar anillados (desarrollo de tesis y anteproyecto) de tesis y cd's según formato indicado en Decanato	Estudiante	1 a 2 semanas
8	Proponer miembros del Tribunal de Grado y solicitud de fecha para defensa práctica o privada	Director de Tesis / Estudiante	1 semana
9	Revisión de aptitud legal para defensa práctica y ratificación o cambio de Tribunal de Grado.	Consejo Directivo (HCD) / Secretario Abogado	Acorde planificación de actividades y sesiones HCD
10	Fijar fecha para defensa práctica	Coordinador de Carrera	15 días posterior a resolución del Consejo Directivo
11	Entrega de anillados a miembros de Tribunal de Grado	Decanato	2 a 3 días posterior aprobación del Tribunal de Grado
12	Defensa Práctica o Privada	Estudiante / Tribunal de Grado	1 día
13	Proponer cambios y/u observaciones	Tribunal de Grado	1 día
14	Revisión y correcciones de Anillados.	Tribunal de Grado	1 mes
15	Asentamiento de 2da nota de Tesis	Tribunal de Grado	1 semana
16	Entregar resolución y notas al Docente de Tesis II	Consejo Directivo / Estudiante	1 semana
17	Asentamiento de nota correspondiente a la materia de Tesis II	Docente de Tesis II	Periodo para asentamiento de notas
18	Revisar que el estudiante cumpla con todos los requisitos para egresamiento	Secretario Abogado	1 semana
19	Revisión de aptitud legal para defensa pública y ratificación o cambio de Tribunal de Grado.	Consejo Directivo (HCD) / Secretario Abogado	Acorde planificación de actividades y sesiones del HCD
20	Nombrar presidente del Tribunal de Grado, día y hora para defensa pública	Consejo Directivo (HCD)	Planificación de actividades y sesiones del HCD

PROCESO: TITULACIÓN DE LA CISIC			
SUBPROCESO: Desarrollo y Aprobación de Tesis			
Objetivo: Aprobar la materia de Tesis II y aprobación de tesis			
Nro.	Actividades	Responsables	Duración
21	Notificación a miembros de Tribunal de Grado y al Tesista	Coordinador de Carrera	1 a 2 semanas
22	Defensa Publica	Estudiante / Tribunal de Grado	1 día
23	Revisión de notas y promedio de los semestres cursados	Secretario Abogado	1 día
24	Asentamiento notas de grado correspondientes a los 4 miembros del Tribunal de Grado	Tribunal de Grado	1 día

Fuente: Propia

2.3.1 Plan de desarrollo

Antecedentes:

La Universidad Técnica del Norte dentro de su planta de profesionales cuenta con personal muy capacitado para el desarrollo de cualquier aplicación que la universidad así lo requiera, hay que recalcar que la UTN se encuentra siempre innovando y en constante mejora del software que facilite el correcto cumplimiento de las actividades cotidianas para el correcto funcionamiento de la institución.

Las seguridades a nivel del área informática tienen un alto grado de confiabilidad, es por lo que el ingreso del personal y los usuarios que tengan acceso al Sistema Integrado de la UTN, deben ser restringidos.

Con el fin de no alterar las funciones del área de informática de la UTN, la mejor opción fue desarrollar la aplicación inicialmente sin tener acceso al sistema integrado de la Universidad Técnica del Norte, lo cual fue una limitante por el poco acceso a la información, pero también fue una gran ventaja porque las pruebas que se pudieron hacer fueron ilimitadas y no generaron efectos secundarios.

Actividades:

Las actividades generales que se deberían realizar para la implementación de la herramienta de estudio en un ambiente de desarrollo real son:

- Integrar las tablas de la aplicación y verificar la compatibilidad de las pantallas con el Sistema Integrado de la UTN.
- Asignar permisos a los usuarios según el tipo de usuario.
- Comprobar que los usuarios puedan visualizar las pantallas desde su puesto de trabajo.

2.3.2 Procedimiento de manejo.

Requerimientos iniciales para la aplicación

Los equipos donde se va a usar la máquina virtual deben tener instalado Oracle VM Virtualbox, que es una herramienta de virtualización de código abierto multiplataforma disponible para Windows, Linux y Mac OS X u otros sistemas operativos.

Conocimiento inicial del usuario

Los usuarios que vayan a interactuar con la aplicación deben tener un conocimiento previo de Oracle Forms como mínimo, en virtud de que no generar un mal manejo de la aplicación.

Desarrollo del procedimiento para el manejo de la aplicación

El manejo de esta aplicación en un ambiente desarrollo real es muy sencillo.

Cada usuario tendrá su propio menú los cuales van a estar divididos en 2 submenús:

- Opciones de ingreso o actualización
- Reportes

Como se detalla en la etapa VISIÓN, Es importante recordar que el sistema tiene cuatro tipos de usuarios, en consecuencia, existen cinco tipos de menús principales los cuales se activarán o se podrán visualizar de acuerdo con el tipo de usuario que ingrese a la aplicación.

Los tipos de usuarios para esta aplicación son:

- Administrativo (coordinadores, decano y subdecano, secretario abogado de la facultad)
- Auxiliar (secretaria de la carrera)
- Docente (tutores de tesis o docentes de las materias que den seguimiento al proceso de tesis)
- Estudiante (estudiantes de la carrera)

Cada usuario puede ejecutar las acciones únicamente que le correspondan dentro del proceso de titulación del estudiante, sin interferir en las funciones de otro usuario.

2.3.3 Estructura de la base de datos

2.3.3.1 Elementos que conforman la base de datos.

Con el propósito de que este sistema pueda ser usado con fines académicos a continuación se detalla las tablas que conforman la base de datos.

TABLAS INICIALES

En las tablas docentes y estudiantes se registran los docentes y estudiantes en general.

Tabla 20. TESIS_DOCENTE

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
CI_DOCENTE	VARCHAR2(10)	No	1	Cédula del Docente
NOMBRE_DOCENTE	VARCHAR2(100)	No	-	Nombre del Docente
CORREO	VARCHAR2(50)	No	-	Correo del Docente
ESTADO_TESIS1	VARCHAR2(1)	No	-	Determina si tiene asignada la materia de TESIS I

Fuente: Propia

Es importante mencionar que en la Tabla 20 se puede ver que la Tabla TESIS_DOCENTE, contiene un campo llamado ESTADO_TESIS 1, el cual permite registrar con “S” a los profesores que imparten cátedra en la materia de TESIS 1, de no ser el caso almacena la “N”.

Tabla 21. TESIS_ESTUDIANTE

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	
CI_ESTUDIANTE	VARCHAR2(10)	No	1	Cédula del Estudiante
NOMBRE_ESTUDIANTE	VARCHAR2(100)	No	-	Nombre del Estudiante

Fuente: Propia

TABLAS PARA REGISTRO DE RESOLUCIÓN CONSEJO ACADÉMICO

En la tabla TESIS_PERIODO_ACAD (Tabla 22) se almacena los periodos académicos, que constan de mes y año inicio junto con mes y año final.

El campo ACTIVO de la Tabla 22 determina si el periodo académico está vigente y tiene dos posibles valores “S” para cuando está activo o vigente ese periodo académico y “N” cuando el periodo académico este deshabilitado.

Tabla 22. TESIS_PERIODO_ACAD

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_PERIODO	VARCHAR2(10)	No	1	Código del Periodo Académico.
MES_INICIO	VARCHAR2(20)	Yes	-	Mes de inicio del periodo académico
AÑO_INICIO	VARCHAR2(4)	Yes	-	Año de inicio del periodo académico
MES_FIN	VARCHAR2(20)	Yes	-	Mes final del periodo académico
AÑO_FIN	VARCHAR2(4)	Yes	-	Año final del periodo académico
INDICADOR	NUMBER	Yes	-	variable Auxiliar
ACTIVO	VARCHAR2(1)	Yes	-	Determina si está activo el periodo académico

Fuente: Propia

Cada vez que se crea un periodo académico, el estado o el campo ACTIVO de la Tabla 22 se cambia a ‘N’ en el periodo anterior, este cambio se realiza en virtud de que cada vez que sea crea un anteproyecto automáticamente se le asigna a ese anteproyecto el periodo académico que este activo o vigente (valor del campo ACTIVO de la Tabla 22 sea igual ‘S’).

El código del periodo académico (COD_PERIODO de la tabla 22) se genera automáticamente cuando se crea un periodo académico y es la combinación de las tres primeras letras del mes de inicio y el año de inicio del periodo académico que se está creando.

Tabla 23. TESIS_GRUPO_INT_ACAD

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_GRUPO	NUMBER	No	1	Código del grupo de estudiantes y docentes que conforman el CA
ESTADO	VARCHAR2(1)	Yes	-	Determina si está activo
VIGENTE	VARCHAR2(1)	Yes	-	Determina si está vigente

Fuente: Propia

Esta tabla crea el código del grupo para asignarle un estado y si está vigente o no, el detalle de los integrantes se almacena en la Tabla 24.

La diferencia entre ESTADO y VIGENTE de la Tabla 23, está en que un grupo puede estar vigente pero no activo, por ejemplo; en un periodo académico puede haber más de un grupo de Consejo Académico (CA) pero cada resolución de CA tiene solo un grupo de integrantes.

Los valores para los campos ESTADO y VIGENTE son “S” si está activo o vigente y “N” en caso contrario.

Al registrar cada resolución del CA, se asigna automáticamente el grupo que esté vigente (“S”).

No puede existir un grupo de CA (Tabla 23) que el valor del campo ESTADO sea “N” y el valor del campo VIGENTE sea “S”.

El campo FUNCION de la Tabla 24 solo puede tener dos posibles valores “ESTUDIANTE” y “DOCENTE”.

Tabla 24. TESIS_INTEGRANTES_ACAD

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_GRUPO	NUMBER	No	1	Clave foránea de la Tabla TESIS_GRUPO_INT_ACAD
CEDULA	VARCHAR2(10)	No	2	Cédula del estudiante o docente que conforma el CA
FUNCION	VARCHAR2(15)	Yes	-	Determina si es estudiante o docente

Fuente: Propia

Tabla 25. TESIS_ACADEMICO

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_ACAD	NUMBER	No	1	Código asignado de forma secuencial, para registrar resoluciones del CA
FECHA_ACADEMICO	DATE	Yes	-	Fecha de resolución de CA
COD_PERIODO_ACAD	VARCHAR2(10)	No	-	Clave Foránea de la Tabla TESIS_PERIODO_ACAD
COD_GRUPO	NUMBER	Yes	-	Clave Foránea de la Tabla TESIS_GRUPO_INT_ACAD
NRO_RESOLUCION	VARCHAR2(50)	Yes	-	Número de Resolución de CA

Fuente: Propia

La Tabla 25 contiene la información básica de las resoluciones académicas, relaciona el periodo académico (campo COD_PERIODO_ACAD) con el grupo del CA (campo COD_GRUPO) y la resolución del CA.

Tabla 26. TESIS_DET_ACAD

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_ACADEMICO	NUMBER	No	1	Clave Foránea de la Tabla TESIS_ACADEMICO
COD_ANTEPROYECTO	NUMBER	No	2	Clave Foránea de la Tabla TESIS_ANTEPROYECTO
DET_RESOLUCION	VARCHAR2(1000)	Yes	-	Detalle de resolución de CA
ESTADO	VARCHAR2(1)	Yes	-	Si está aprobado o no
FECHA_DEFENSA_ANT	DATE	Yes	-	Fecha de defensa anteproyecto
FECHA_APROBACION_ANT	DATE	Yes	-	Fecha aprobación anteproyecto

Fuente: Propia

En la Tabla 26 se puede ver la relación de la resolución del CA (COD_ACADEMICO) y el anteproyecto (COD_ANTEPROYECTO).

También se registra el detalle de la resolución de la aprobación o desaprobación del anteproyecto.

El campo ESTADO de la Tabla 26 puede contener tres valores:

- A: Revisado y Aprobado por el CA.
- N: No aprobado por el CA.
- C: El CA sugiere correcciones.

La fecha de aprobación del anteproyecto (campo FECHA_APROBACION_ANT de la Tabla 26) solo se actualiza, si el anteproyecto es aprobado en CA.

TABLAS PARA EL PROCESO DE TITULACIÓN

Tabla 27. TESIS_DIRECTOR_PROPUESTO

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_ANTEPROYECTO	NUMBER	No	-	Clave Foránea de la Tabla TESIS ANTEPROYECTO
CI_DOCENTE_DIRECTOR	VARCHAR2(10)	No	-	Cédula del Tutor o Director de Tesis y clave foránea de la Tabla TESIS_DOCENTE
ESTADO	CHAR(1)	No	-	Aprobado, no aprobado o el consejo directivo asigno otro director
FECHA	DATE	SI	-	Fecha de cambio de Director

Fuente: Propia

Todo anteproyecto debe tener un Director o Tutor de Tesis, lo cual se guarda en la Tabla 27.

El campo ESTADO de la Tabla 27, se refiere a si el Docente aprueba (“A”), no aprobado (“N”), ser el Director de Tesis.

En el caso de que el Honorable Consejo Directivo cambie de Director el estado se cambiará a “C”.

Se asigna un valor al campo fecha solo en caso de que el director de tesis sea cambio por el Honorable Consejo Directivo.

Las líneas de investigación en el campo de la informática y la tecnología son variadas, en la Figura 9 se puede ver las líneas de investigación que tiene la FICA y la relación que tienen entre las carreras, lo cual es demuestra que es posible que se realicen trabajos de grado interdisciplinarios.

Tabla 28. TESIS_LINEA_INVESTIGACIÓN

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_LINEA	NUMBER	No	1	Código Línea de Investigación
DETALLE	VARCHAR2(500)	No	-	Detalle Línea de Investigación

Fuente: Propia

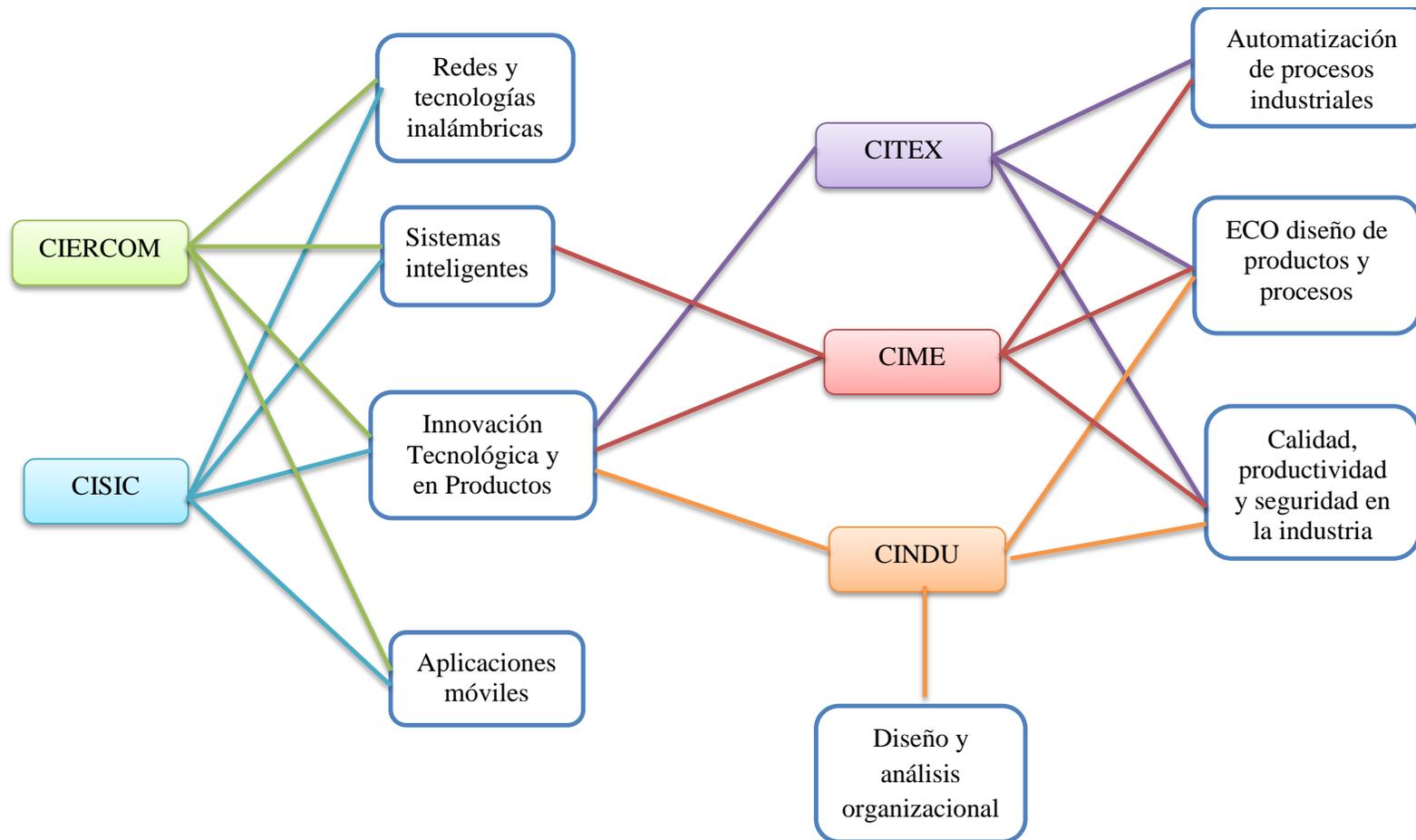


FIGURA 9. Líneas de Investigación de las carreras de la Facultad de Ingeniería en Ciencias Aplicadas y la relación que existe entre las líneas de investigación de cada carrera.

Fuente: Propia

Tabla 29. TESIS_ANTEPROYECTO

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_ANTEPROYECTO	NUMBER	No	1	Código del anteproyecto
CI_ESTUDIANTE	VARCHAR2(10)	No	-	Clave foránea de la Tabla TESIS_ESTUDIANTE
TEMA_TESIS	VARCHAR2(1000)	No	-	Título del Anteproyecto
CI_DOCENTE_TESIS	VARCHAR2(10)	No	-	Cédula del Docente de Tesis I y clave foránea de la Tabla TESIS_DOCENTE
COD_LINEA	NUMBER	No	-	Clave foránea de la tabla TESIS_LINEA_INVESTIGACION
CI_DIRECTOR	VARCHAR2(10)	Yes	-	Cédula del Director de Tesis y clave foránea de l Tabla TESIS_DOCENTE
ESTADO	VARCHAR2(1)	Yes	-	Estado del Anteproyecto

Fuente: Propia

La Tabla 29 contiene los detalles iniciales del anteproyecto, estudiante al que pertenece, el docente de la materia TESIS I, Director de TESIS y Línea de Investigación y el estado que se encuentra el anteproyecto.

Una vez que el Director acepta ser Director de Tesis el campo CI_DIRECTOR de la Tabla 29, se actualiza automáticamente.

A continuación, se detalla los valores que puede tener el campo ESTADO de la Tabla 29, en el sistema durante el proceso de Titulación.

Tabla 30. Estados el Anteproyecto

ESTADO	DESCRIPCION
E	En espera de asignación de director
P	Creado, asignado director y sin revisión
D	Anteproyecto revisado y aprobado por el director de la tesis
T	Revisado y aprobado por el docente de tesis 1
A	Revisado y aprobado en académico
N	No aprobado
H	Aprobado por consejo directivo
L	Aprobado y con tribunal de grado asignado
R	Defensa privada aprobado
S	Defensa publica aprobada, fin del proceso de titulación

Fuente: Propia

La Tabla 31 contiene la estructura básica de un anteproyecto, la cual se crea automáticamente cuando se crea el anteproyecto, por tal motivo esta tabla debe ser modificada solo en caso de que la estructura del anteproyecto cambie.

Tabla 31. TESIS_RESUMEN

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_RESUMEN	NUMBER	No	-	Código de Resumen
DETALLE_RESUMEN	VARCHAR2(200)	No	-	Detalle de Resumen

Fuente: Propia

El detalle de los tipos de resumen que se generan se encuentra en la Tabla 32.

Tabla 32. Tipos de Resumen del Anteproyecto

CÓDIGO RESUMEN	DETALLE RESUMEN
1	Tema de Tesis
2	Antecedentes
3	Situación Actual
4	Planteamiento del Problema
5	Objetivo General
6	Objetivo Específico
7	Alcance
8	Arquitectura Funcional
9	Arquitectura Tecnológica
10	Alcance de la Investigación
11	Limitaciones
12	Justificación
13	Temas Afines Realizados
14	Temario
15	Cronograma
16	Costos
17	Anteproyecto

Fuente: Propia

Tabla 33. TESIS_ANT_RESUMEN

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_ANTEPROYECTO	NUMBER	No	1	Clave foránea de la Tabla TESIS_ANTEPROYECTO
COD_RESUMEN	NUMBER	No	2	Clave foránea de la Tabla TESIS_RESUMEN
ESTADO	VARCHAR2(1)	Yes	-	Estado del Resumen

Fuente: Propia

La Tabla 33 contiene la relación del anteproyecto con los tipos de resumen asignados.

Tabla 34. TESIS_CORRECCIONES

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_CORRECCION	VARCHAR2(10)	No	1	Código de Corrección
COD_ANTEPROYECTO	NUMBER	No	-	Clave foránea de la Tabla TESIS_ANTEPROYECTO
COD_RESUMEN	NUMBER	No	-	Tipo de Resumen que requiere corrección
CONTADOR	NUMBER	Yes	-	Variable Auxiliar
DETALLE_CORRECCION	VARCHAR2(2000)	Yes	-	Descripción de la corrección
ESTADO	VARCHAR2(1)	Yes	-	Estado de la corrección
ASIGNADO_POR	VARCHAR2(1)	Yes	-	Persona que sugiere la corrección

Fuente: Propia

En el campo ASIGNADO_POR de la Tabla 34 se puede asignar el valor “T” para Docente de la materia TESIS I, “D” para Director de TESIS y “A” en el caso del CA.

Tabla 35. TESIS_OBJETIVOS

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_OBJETIVO	VARCHAR2(20)	No	1	Código de Objetivo Específicos
COD_ANTEPROYECTO	NUMBER	No	-	Clave foránea de la Tabla TESIS_ANTEPROYECTO
CONTADOR	NUMBER	Yes	-	Variable Auxiliar
DETALLE_OBJETIVO	VARCHAR2(500)	Yes	-	Detalle de Objetivos Específicos
ESTADO	VARCHAR2(1)	Yes	-	Aprobado o No

Fuente: Propia

La Tabla TESIS_OBJETIVOS, es fundamental para la creación del cronograma del anteproyecto.

Las actividades están relacionadas directamente a cada uno de los objetivos específicos y se registran en la Tabla 36.

Las fechas de Inicio y fin (Tabla 36) se calculan automáticamente con el número de días asignado en el cronograma.

Tabla 36. TESIS_CRONOGRAMA

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_CRONOGRAMA	VARCHAR2(20)	No	1	Código del Cronograma
COD_OBJETIVOS	VARCHAR2(20)	No	-	Clave foránea de la Tabla TESIS_OBJETIVOS
DETALLE_ACTIVIDAD	VARCHAR2(500)	Yes	-	Detalle de la Actividad que cumpla el objetivo específico
NRO_ACTIVIDAD	NUMBER	No	-	El número de actividad
DIAS	NUMBER	No	-	Días para el cumplimiento de la actividad
FECHA_INICIO	DATE	Yes	-	Fecha de Inicio de la actividad
FECHA_FIN	DATE	Yes	-	Fecha en el que se debe terminar la actividad
ESTADO	VARCHAR2(1)	Yes	-	Estado del Cronograma

Fuente: Propia

Tabla 37. TESIS_TEMARIO

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_TEMARIO	VARCHAR2(20)	No	1	Código de Temario
COD_CAPITULO	NUMBER	No	-	Número del Capítulo
COD_ANTEPROYECTO	NUMBER	No	-	Clave foránea de la Tabla TESIS_ANTEPROYECTO
DET_CAPITULO	VARCHAR2(100)	Yes	-	Detalla del capítulo
ESTADO	VARCHAR2(1)	Yes	-	Estado del capítulo

Fuente: Propia

La Tabla 37 contiene la estructura del Temario o Contenido de la Tesis.

Tabla 38. TESIS_DET_TEMARIO

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_TEMARIO	VARCHAR2(20)	No	1	Clave foránea de la Tabla TESIS_TEMARIO
NRO_TEMA	VARCHAR2(10)	No	2	Número de la subtema del Capítulo
CONTADOR	NUMBER	Yes	-	Variable Auxiliar
DETALLE_TEMA	VARCHAR2(100)	Yes	-	Detalle del subtema
ESTADO	VARCHAR2(1)	Yes	-	Estado del subtema

Fuente: Propia

La Tabla 38 es el detalle de la Tabla 37.

Tabla 39. TESIS_DIRECTIVO

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_DIRECTIVO	NUMBER	No	1	Código del Consejo Directivo
NRO_RESOLUCION	VARCHAR2(30)	No	-	Número de Resolución de Directivo
FECHA	DATE	Yes	-	Fecha en la que se realiza el Directivo

Fuente: Propia

La Tabla 39 contiene solo el número resolución del Consejo Directivo y la fecha para que sea fácil la búsqueda.

Tabla 40. TESIS_DEFENSA

Nombre de Columna	Tipo de Dato	Nulo	Clave Primaria	Comentario
COD_DEFENSA	VARCHAR2(20)	No	1	Código de Defensa
COD_DIRECTIVO	NUMBER	No	-	Clave foránea de l a Tabla TESIS_DIRECTIVO
COD_ANTEPROYECTO	NUMBER	No	-	Clave foránea de la Tabla TESIS_ANTEPROYECTO
CONTADOR	NUMBER	Yes	-	Variable Auxiliar
TIPO	VARCHAR2(1)	Yes	-	Tipo defensa
ESTADO	VARCHAR2(1)	Yes	-	Estado de la defensa
NOTA_DIRECTOR	NUMBER	Yes	-	Nota asignada por el Director
CI_TRIBUNAL1	VARCHAR2(10)	No	-	Cédula de Docente que cumple funciones de Tribunal de Grado
NOTA1	NUMBER	No	-	Nota asignada por el miembro del tribunal
CI_TRIBUNAL2	VARCHAR2(10)	No	-	Cédula de Docente que cumple funciones de Tribunal de Grado
NOTA2	NUMBER	No	-	Nota asignada por el miembro del tribunal
CI_TRIBUNAL3	VARCHAR2(10)	No	-	Cédula de Docente que cumple funciones de Tribunal de Grado
NOTA3	NUMBER	No	-	Nota asignada por el miembro del tribunal

Fuente: Propia

2.3.3.2 Relación entre los elementos de la base de datos

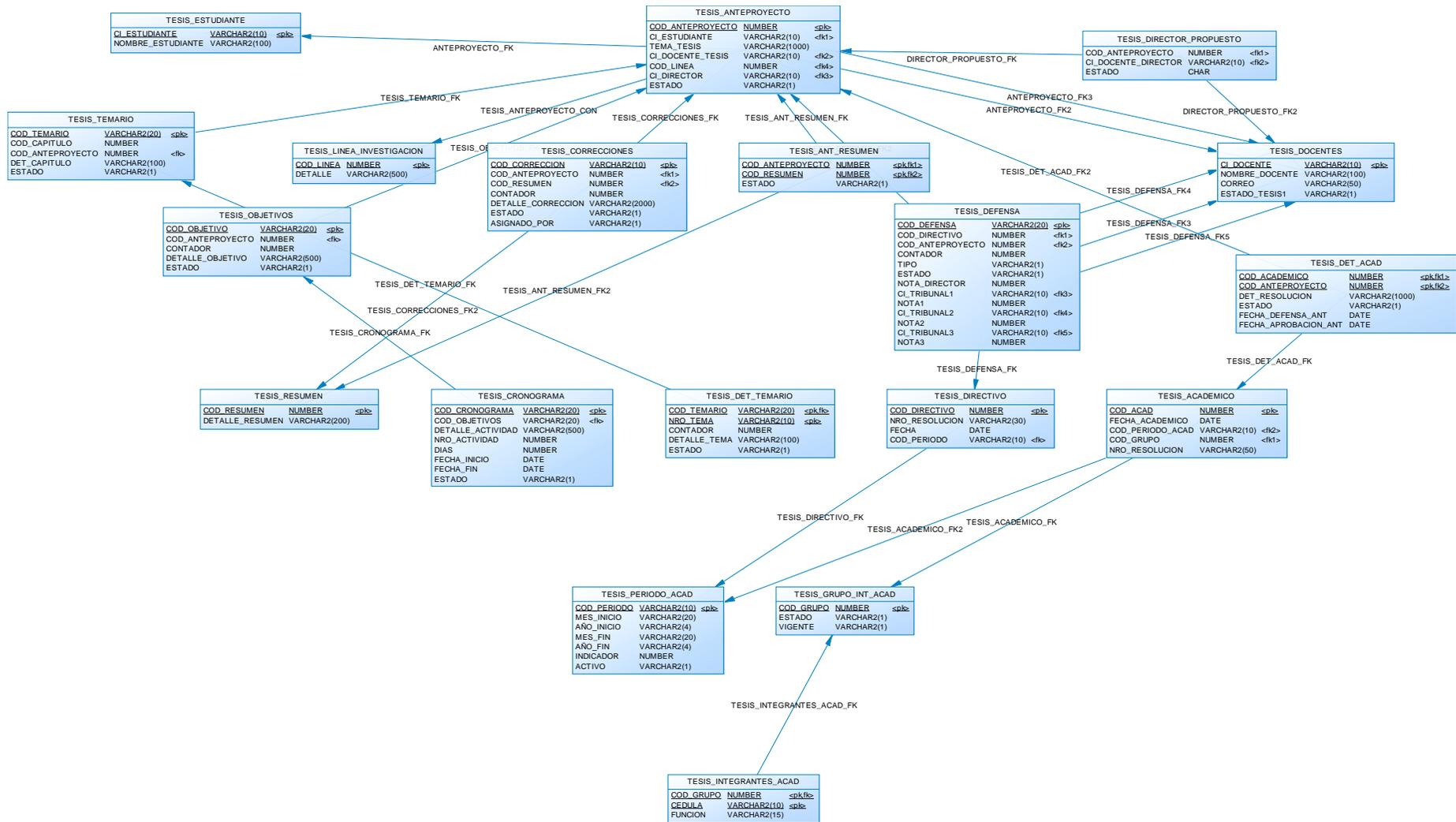


FIGURA 10. Modelo Físico de la Base de Datos

Fuente: Propia

2.4 Estabilización

2.4.1 Objetivo de realizar Pruebas

Las pruebas realizadas con el usuario sirven para corregir algunas opciones de la aplicación y volver a la aplicación más amigable con el usuario.

Esta fase de desarrollo de la aplicación toma aproximadamente un mes.

Dentro de la Metodología MSF esta fase de pruebas inicia con una solución inicial a la cual se la conoce como solución Beta, debido a que la solución es solo la primera versión del sistema en desarrollo.

Es decir que existe una alta posibilidad de que existan varias versiones antes de poner en ejecución o funcionamiento un sistema.

2.4.1 Pruebas en la ejecución del Sistema

Al estar en el aplicativo en una máquina virtual y no tener cliente final asignado esta fase, es empírica, pero como herramienta de estudio se establece un ejemplo de cómo planificar pruebas con el usuario o cliente.

El sistema está desarrollado con Oracle Forms, con base de datos Oracle.

Por lo antes mencionada antes de ejecutar la aplicación se deberán realizar las siguientes actividades:

Tabla 41. Planificación de actividades para realizar pruebas con el usuario

Actividad	Tiempo	Fecha Inicio	Fecha Fin
Socialización de la aplicación para usuarios principales (tutores de tesis, docentes de seguimientos de tesis, secretarías FICA, coordinadores, decano y subdecano)	2 días	05/04/2018	06/04/2018
Inducción de manejo de la aplicación (pantallas principales y reportes)	5 días	09/04/2018	13/04/2018
Receptar recomendaciones de los usuarios que no afecten los requerimientos iniciales	15 días	16/04/2018	30/04/2018
Socialización e interacción de los usuarios con la aplicación terminada	10 días	02/05/2018	11/05/2018
Recopilación de sugerencias y cambios al sistema	5 días	14/05/2018	18/05/2018

Fuente: Propia

2.5 Implementación.

La etapa de implementación se resume en que la versión del software sea aprobada por el cliente acorde a los requerimientos definidos previamente.

2.6 Pruebas

La forma correspondiente a la opción “DEFENSA PÚBLICA” del menú “AUXILIAR” se probó con el test de la Caja Blanca, se evaluó el siguiente código, las líneas punteadas (--) son comentarios para describir la acción del código fuente:

Asigna a cada variable nota, el valor que contienen los campos Nota Director y Nota de Tribunal 1, 2 y 3-----

```
nota:=:TESIS_DEFENSA.NOTA_director;
```

```
nota1:=:tesis_defensa.nota1;
```

```
nota2:=:tesis_defensa.nota2;
```

```
nota3:=:tesis_defensa.nota3;
```

```
suma:=nota+nota1+nota2+nota3;
```

---- La variable suma es la sumatoria de las variables que corresponden a notas----

```
promedio:=round(suma/4,2);
```

----- La variable promedio como su nombre lo indica almacena el promedio de las 4 notas, redondeado mediante la función round-----

```
:tesis_defensa.calculo:=promedio;
```

----- Asigna a un campo que no permite cambios el valor del promedio para que se pueda visualizar-----

```
if (promedio > 7) then
```

----- Si el promedio es mayor que 7 el sistema automáticamente aprueba o asigna el valor de aprobado al anteproyecto y muestra en un campo que no permite registro la palabra “APROBADO”-----

```
info:='APROBADO';
```

```
else
```

```
info:='NO APROBADO';
```

El modelo de la tabla 42 se obtuvo del libro Ingeniería de Software de Pantaleo G. y Rinaudo L (2015, p. 268), las clases de equivalencia y la información que contiene esta tabla fue aplicada al sistema de seguimiento del proceso de titulación de los estudiantes de la CISIC.

Tabla 42. Tabla para clase de equivalencia de acuerdo con el test de la Caja Blanca

Clases de Equivalencia			
Id Test	Condición Entrada	Clase Equivalencia	Test
1	Nota del Director Fija (nota) y Notas del Tribunal Variables (nota1, nota2, nota3)	notas1, 2 y 3 > 0	nota = 7, nota1 = 9, nota2 = 7, nota3 = 8
2		nota1 = 0, nota2 y nota3 > 0	nota = 7, nota1 = 0, nota2 = 8.5, nota3 = 6
3		nota1 y nota2 = 0, nota3 > 0	nota = 7, nota1 = 0, nota2 = 0, nota3 = 9
4		nota2 y nota 3 < 0, nota1 > 0	nota = 7, nota1 = 9, nota2 = -5, nota3 = -3
5		nota1 < 0, nota2 y nota3 > 0	nota = 7, nota1 = -4, nota2 = 9, nota3 = 8.5
6	Nota del Director variable (nota) y Notas del Tribunal Fijas (nota1, nota2, nota3)	nota > 0	nota = 5, nota1 = 9, nota2 = 7, nota3 = 8
7		nota = 0	nota = 0, nota1 = 9, nota2 = 7, nota3 = 8
8		nota < 0	nota = -5, nota1 = 9, nota2 = 7, nota3 = 8

Fuente: Propia

Los resultados se pueden ver en la tabla 44 del capítulo III en la sección “Comparación de Resultados”.

CAPITULO III

3.1 Valoración del Software

Para la valoración del software se aplicó precios referenciales de la zona urbana de Ibarra.

Si bien es cierto no existe una forma específica para calcular el valor de un producto software, pero se realizó un cálculo aproximado en base a la experiencia profesional, con los posibles gastos que enfrenta un emprendimiento.

Tabla 43. Detalle de valoración del aplicativo para dar seguimiento al proceso de titulación de los estudiantes de la CISIC.

Descripción	Forma de Cálculo	Valor
Costo mensual de un desarrollador junior (sin experiencia) en promedio (en este valor está incluido alimentación y transporte del desarrollador)	$800 * 4$ meses	3200
Servicios básicos (luz, agua, teléfono) promedio	(promedio luz 25 USD, promedio agua 18 USD, promedio teléfono 25 USD)*20% * 4 meses	54,4
Servicio de Cable (costo promedio)	$(19,90 * 4) * 20\%$	15,92
Arriendo Costo Promedio Ibarra	$(150 * 4) * 20\%$	120
Útiles de Oficina (papel, copias, tóner)	(Resmas de Papel 8 USD * 2 Unidades por mes en 4 meses, copias y gastos varios 200 USD en 4 meses, Tóner 75 USD * 2 Unidades unidades)*20%	82,8
Valor Total		3473,12

Fuente: Propia

El 20% fue asignado con la consideración de que el proyecto es para la Universidad Técnica del Norte y con fines académicos, este porcentaje puede variar de acuerdo con el nivel económico del cliente y a la magnitud del proyecto.

Los cálculos de la tabla 43 se hacen para un tiempo de 4 meses, tiempo establecido y aprobado por el Honorable Consejo Académico mediante la aprobación del anteproyecto.

3.2 Comprobación de Resultados

Luego de haber probado en el aplicativo los valores que se muestran en la tabla 42 que se encuentra en la sección “Pruebas” de este documento, se pudo verificar que el sistema da los resultados esperados, el detalle de los resultados obtenidos se puede visualizar en la tabla 44.

Tabla 44. Comprobación de resultados obtenidos de acuerdo con el test de la Caja Blanca que se muestra en la tabla 41.

Id Test	Promedio Obtenido	Resultado de acuerdo con el código fuente	¿El resultado obtenido es correcto?
1	7,75	APROBADO	VERDADERO
2	5,38	NO APROBADO	VERDADERO
3	4,50	NO APROBADO	VERDADERO
4	2,00	NO APROBADO	VERDADERO
5	5,13	NO APROBADO	VERDADERO
6	7,25	APROBADO	VERDADERO
7	6,00	NO APROBADO	VERDADERO
8	4,75	NO APROBADO	VERDADERO

Fuente: Propia

Con el fin de probar la usabilidad del aplicativo se realizó una encuesta a 30 jóvenes entre 17 y 18 años.

Una vez aplicada la encuesta se obtuvo los siguientes resultados:

Tabla 45. Resultado de encuesta aplicada a 30 jóvenes (número de personas)

	Muy de acuerdo	De acuerdo	Indeciso	En desacuerdo	Muy en desacuerdo
Es fácil el ingreso a la aplicación	15	10	1	4	0
El aplicativo es intuitivo	14	12	3	1	0
El ingreso de datos es sencillo	21	7	2	0	0
La navegación entre pantallas es funcional	17	10	3	0	0
Los reportes son adecuados	18	10	1	1	0

Fuente: Propia

Los rangos de edad fueron seleccionados por ser la edad aproximada en la que un estudiante ingresa a la universidad. Es importante mencionar que uno de los objetivos es que el aplicativo pueda ser usado desde los primeros niveles para que tengan una referencia de cómo desarrollar un sistema.

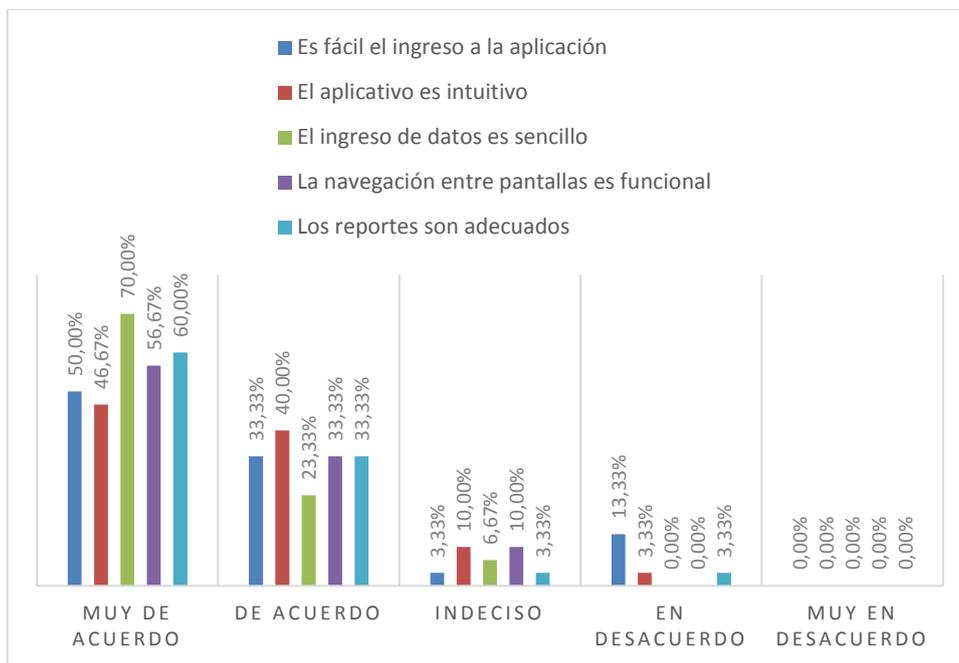


FIGURA 11. Resultado de encuesta aplicada a 30 jóvenes (porcentajes)

Fuente: Propia

Luego de haber analizado los resultados se puede observar que el ingreso al sistema tiene cierto grado de dificultad, sin embargo, todos los encuestados lograron ingresar a la aplicación.

Adicional, el 70% de los jóvenes encuestados afirmaron que el ingreso de datos es sencillo, un alto porcentaje cree que la navegabilidad entre pantallas y los reportes son coherentes con la funcionalidad del sistema.

Con respecto a si el sistema es intuitivo, que consiste en evaluar si es fácil saber que opciones del menú escoger y como seleccionar las opciones afirmaron que se les dificultó por no estar relacionados con el proceso de titulación pero que los enunciados de las opciones son claros y da una idea de la información que requiere el formulario.

Conclusiones y Recomendaciones

Conclusiones

- La tecnología debe ajustarse al entorno para que sea funcional.
- Las metodologías ágiles son una alternativa para evitar que los sistemas se vuelvan obsoletos con el paso del tiempo.
- Es evidente que, en una actualidad tecnológica, sistematizar procesos ya no es una opción, se ha vuelto una necesidad, debido a que las personas se orientan a optimizar tiempo y recursos.
- El sistematizar un proceso permite tener una herramienta administrativa que facilita a las autoridades la información oportuna para la toma de decisiones.
- El que un sistema pueda desarrollarse mediante la generación de versiones, permite al desarrollador y al cliente tener un sistema adaptable a los cambios del entorno.
- Las etapas de la metodología Microsoft Solution Framework (MSF) disminuye la cantidad de documentación en comparación con otras metodologías, que requieren un sin número de información que en ocasiones no se usa.
- Los entregables de cada etapa de la metodología MSF, producen una mejor comunicación entre desarrollador y cliente, lo cual asegura el éxito en el cumplimiento de los requerimientos del sistema en desarrollo.
- El sistema informático para el Proceso de Titulación de la Facultad de Ingeniería en Ciencias Aplicadas permite a los estudiantes tener una mejor planificación en el cumplimiento del cronograma de actividades.

Recomendaciones

- Incluir en los programas académicos el estudio de metodologías ágiles, esto va a contribuir a que el desarrollo de sistemas no se vuelva algo extenso y poco atractivo para los futuros Ingenieros en sistemas.
- Por el bien de las nuevas generaciones, es fundamental empezar a concientizar en el cuidado del medio ambiente promoviendo la disminución del uso de papel /o útiles de oficina y la sistematización de procesos.
- Los tutores de tesis tienen una gran responsabilidad no solo con el estudiante sino con la Universidad, en virtud de que el número de estudiantes graduados es un indicador para los procesos de acreditación, razón por la cual la distribución del número de tesis debe ser asignado equitativamente, facilitando el seguimiento y revisión de cada tesis.
- Establecer procesos que faciliten la titulación para personas que trabajan, debido a que existe un alto porcentaje de estudiantes que no concluyen dicho proceso, por falta de tiempo y/o permisos en las Instituciones donde laboran, hay que mencionar que lo óptimo sería primero obtener el título y luego empezar la vida profesional, pero la Universidad debería considerar que en nuestra sociedad las oportunidades de trabajo son escasas y la economía de los hogares es una necesidad primordial.

Bibliografía

- Landázuri C. (2013). Repositorio Digital UTN, Ecuador. Obtenido de la UNIVERSIDAD TÉCNICA DEL NORTE:
<http://repositorio.utn.edu.ec/bitstream/123456789/1830/1/ISC%20258%20TESIS.pdf>
- Vergara M., Rizzo M. Landázuri C. (2014). Barinas Obtenido de la Universidad Nacional Experimental de los llanos Occidentales:
<http://blogs.unellez.edu.ve/dsilva/files/2014/07/MSF.pdf>
- Villarroel L., Montalvo, C. (2008). Repositorio Dspace. Obtenido de ESCUELA POLITÉCNICA DEL EJÉRCITO:
<https://repositorio.espe.edu.ec/bitstream/21000/335/1/T-ESPE-021837.pdf>
- Sánchez F., Tituaña R. (2007). Repositorio Digital. Obtenido de ESCUELA POLITÉCNICA NACIONAL: <http://bibdigital.epn.edu.ec/bitstream/15000/187/1/CD-0585.pdf>
- Pantusin W., Viracocha E. (2009). Repositorio Institucional. Obtenido de UNIVERSIDAD TÉCNICA DE COTOPAXI: <http://repositorio.utc.edu.ec/handle/27000/457>
- Orjuela A., y Rojas M. (2008). Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo. Revista Avances en Sistemas e Informática, (5)2, 161.: <http://www.bdigital.unal.edu.co/15430/1/10037-18216-1-PB.pdf>
- Acuerdo Ministerial 034 (2014). Reformar el Acuerdo Ministerial No 131 de 11 de agosto del 2017 publicado en Registro Oficial No. 284 de 22 de septiembre del 2010. Ministerio del Ambiente:
<http://www.ambiente.gob.ec/wp-content/uploads/downloads/2014/04/AM034.pdf>
- González Kathy (2016). Presentación de Metodología MSF (Microsoft Solutions Framework).
DOCURI: https://docuri.com/download/msf_59c1ce07f581710b286325d2_pdf
- Microsoft (2013): [https://msdn.microsoft.com/es-es/library/jj161047\(v=vs.120\).aspx](https://msdn.microsoft.com/es-es/library/jj161047(v=vs.120).aspx)
- Oracle: <https://www.oracle.com/corporate/index.html#info>
- Pantaleo, G. & Rinaudo, L. (2015). Ingeniería de Software. Buenos Aires. Editorial Alfaomega, 1a edición.
- Calero C. & Moraga M. & Piattini M. (2010). Calidad del producto y proceso de software. Madrid. Editorial Ra-Ma.
- Pantaleo, G. (2016). Calidad en el desarrollo de software. Buenos Aires. Editorial Alfaomega, 2da edición
- Naranjo M., De la Portilla M., Mora N. (2013). *Estatuto Orgánico U.T.N 2013* (p. 10). Ecuador – Ibarra, Imprenta Universitaria.
- Somerville, I. (2011). Ingeniería de Software. México. Editorial Pearson Education, Novena Edición.
- Sánchez, S. & Sicilia, M. & Rodríguez, D. (2012). Ingeniería del Software – Un enfoque desde la guía SWEBOK. México. Alfaomega, Primera Edición.
- Lucero, J. (2000). Auditoría de la Calidad. En Del Peso Navarro, E. & Piattini Velthuis, M. Auditoría informática: un enfoque práctico (361-388). España, Editorial Ra-Ma
- Palacio J. (2015), Scrum Manager I – Las reglas de Scrum, Scrum Manager, Versión 2.5.1:
http://www.scrummanager.net/files/scrum_I.pdf
- Poppendieck, M., & Poppendieck, T. (2003). Lean Software Development: An Agile Toolkit for Software Development Managers. Addison Wesley.
- Piattini M, Del Peso E., (2001). Auditoría Informática Un enfoque Práctico. España-Madrid. Ra-Ma, 2da. Edición ampliada y revisada
- Carmona M. A. (2013), Blog Corporativo, España, IAT Innovación y Tecnología,
<http://www.iat.es/2013/10/gestion-riesgo-clave-excelencia-empresarial-iv/>

Anexos

Anexo 1: Procesos de Titulación de la Facultad de Ingeniería en Ciencias Aplicadas.

Anexo 2: Manual de usuario del aplicativo para seguimiento del proceso de titulación de la CISIC.

Anexo 3: Manual administrativo del aplicativo para seguimiento del proceso de titulación de la CISIC.