



**UNIVERSIDAD TÉCNICA DEL NORTE
INSTITUTO DE POSTGRADO
MAESTRÍA EN INGENIERÍA DE SOFTWARE**



**“IMPLEMENTACIÓN DEL ESTÁNDAR ISO/IEC 29110 EN EL PROCESO
DE DESARROLLO DE SOFTWARE DE LA DIRECCIÓN DE DESARROLLO
TECNOLÓGICO E INFORMÁTICO DE LA UNIVERSIDAD TÉCNICA DEL
NORTE”**

**Trabajo de Investigación previo a la obtención del Título
de
Magíster en Ingeniería de Software**

DIRECTOR

Ing. Jorge Caraguay Próceo, Msc

AUTOR:

Ing. Edwin Ramiro Madruñero Padilla

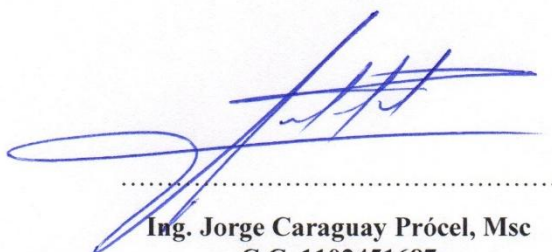
IBARRA - ECUADOR

2018

APROBACIÓN TUTOR

En mi calidad de tutor del Trabajo de Grado, presentado por Edwin Ramiro Madruñero Padilla, para optar por el grado de magister en Ingeniería de Software, doy fé que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación privada y pública y evaluación por parte del jurado examinador que se designe.

Ibarra, 05 de abril del 2018

A handwritten signature in blue ink, consisting of several overlapping strokes, positioned above a dotted line.

.....
Ing. Jorge Caraguay Prócel, Msc
C.C. 1102451687
TUTOR

APROBACIÓN DEL TRIBUNAL EXAMINADOR

En calidad de jurado examinador del presente trabajo presentado por la ING. EDWIN RAMIRO MADRUÑERO PADILLA, para optar por el título de MAGÍSTER EN INGENIERÍA DE SOFTWARE, cuyo tema es: **IMPLEMENTACIÓN DEL ESTÁNDAR ISO/IEC 29110 EN EL PROCESO DE DESARROLLO DE SOFTWARE DE LA DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO DE LA UNIVERSIDAD TÉCNICA DEL NORTE**, considerando que el presente trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación pública y privada y evaluación por parte del tribunal examinador que se designe.

Ibarra, 05 de abril de 2018



Ing. Jorge Caraguay Prócel, Msc



Ing. Alexander Guevara Msc



UNIVERSIDAD TÉCNICA DEL NORTE
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO
DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, **Edwin Ramiro Madruñero Padilla**, con cédula de ciudadanía Nro. **0401581731**, manifiesto voluntad de ceder a la Universidad Técnica del Norte los derechos Patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículo 4,5,6 en calidad de autora del trabajo de grado denominado **“IMPLEMENTACIÓN DEL ESTÁNDAR ISO/IEC 29110 EN EL PROCESO DE DESARROLLO DE SOFTWARE DE LA DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO DE LA UNIVERSIDAD TÉCNICA DEL NORTE”**, que ha sido desarrollado para optar por el Título de **MAGISTER EN INGENIERÍA EN SOFTWARE** en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autora me reservo los derechos morales del trabajo antes citado. En concordancia suscribo este documento en el momento que hago la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Ibarra, a los 05 días del mes de abril de 2018.

Firma:

Nombre: Edwin Ramiro Madruñero Padilla
Cédula: 040158173-1



UNIVERSIDAD TÉCNICA DEL NORTE
INSTITUTO DE POSTGRADO
MAESTRÍA EN CONTABILIDAD Y AUDITORÍA



AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE
LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad. Por medio del presente dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información: DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:		0401581731	
APELLIDOS Y NOMBRES:		Edwin Ramiro Madruñero Padilla	
DIRECCIÓN:		Av. Víctor Manuel Peñaherrera	
EMAIL:		edwmadru@hotmail.com	
TELEFONO FIJO:	062291323	TELÉFONO MOVIL:	0993924203
DATOS DE LA OBRA:			
TÍTULO:		"IMPLEMENTACIÓN DEL ESTÁNDAR ISO/IEC 29110 EN EL PROCESO DE DESARROLLO DE SOFTWARE DE LA DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO DE LA UNIVERSIDAD TÉCNICA DEL NORTE".	
AUTORES:		Edwin Ramiro Madruñero Padilla	
FECHA: aaaa/mm/dd		2018/03/22	
SOLO PARA TRABAJOS DE GRADO			
PROGRAMA:		PREGRADO POSTGRADO <input checked="" type="checkbox"/>	
TÍTULO POR EL QUE OPTA:		Magíster en Ingeniería de Software	
ASESOR/DIRECTOR:		JORGE CARAGUAY PRÓCEL, MSC	

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Edwin Ramiro Madruñero Padilla, con cédula de ciudadanía Nro. 0401581731, en calidad de autora y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior, artículo 144.

3. CONSTANCIAS

La autora manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asumen la responsabilidad sobre el contenido de esta y saldrán en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 05 días del mes de abril de 2018.

LA AUTORA:



Nombre: Edwin Ramiro Madruñero Padilla
Cédula: 040158173-1

DEDICATORIA

El presente trabajo dedico una primera instancia a Dios,
por ser el guía en mi camino y sobre todo en este proceso desarrollo profesional para
cumplir mis metas y objetivos.

A mi madre Gloria y mi Padre Armando,

Por ser el apoyo incondicional en mi vida brindándome su amor, apoyo y sacrificio
incondicional, siendo pilares fundamentales en mi crecimiento profesional y personal.

A mis Hermanos Armando, Luis y Lourdes por ser mis mejores amigos que siempre me
han apoyado y brindado sus consejos.

A mis niños Melany y David por ser mi inspiración todos los días.

A mi princesa Karlita por ser la mujer que siempre me brinda su apoyo y ser mi
compañera de lucha diaria para ser mejor y lograr mis sueños.

Edwin

AGRADECIMIENTO

En primer lugar, agradezco a Dios, por ayudarme a culminar con este objetivo planteado.

A la Universidad Técnica del Norte, que ha sido la institución que me ha permitido formarme profesionalmente y en especial a la Dirección de Desarrollo Tecnológico e Informático, por la colaboración y apertura en la realización del presente proyecto de investigación.

A mis maestros, que fueron el pilar fundamental de ejemplo y saberes para poder culminar con esta meta propuesta.

Un agradecimiento especial al Ing. Jorge Caraguay Prócel Msc., **Tutor** y para el Ing. Alexander Guevara Msc, **Asesor** del presente trabajo de investigación, por su tiempo, por su experiencia profesional y apoyo para la consecución del presente objetivo.

Edwin

ÍNDICE

Dedicatoria.....	vi
Agradecimiento	vii
Índice	viii
Índice de Tablas.....	xiii
Índice de Figuras	xiv
Listado de Siglas.....	xv
Introducción.....	18
Capítulo I: El Problema	20
1.1 Antecedentes.....	20
1.2. Objetivos de la investigación	22
1.2.1. OBJETIVO GENERAL	22
1.2.2. OBJETIVO ESPECÍFICO.....	23
1.3 Proposición	23
1.4. Preguntas directrices	23
Capitulo II: Marco Referencial.....	24
2 Marco Teórico	24
2.1 Gestión de Proceso	24
2.2 Proceso	24
2.3 Software	26
2.4 Ingeniería de Software	27
2.5 Proceso de Software	27
Proceso	30
Subprocesos	30
Procesos Principales.....	30
Procesos Organizacionales	30
Procesos de Soporte	30
Procesos de Adaptación	30
2.6 Proyectos de software.....	34
2.7 Metodología	36
2.8 Metodologías Agiles	36
2.9 Metodología Scrum.....	39
2.10 Calidad	42
2.10.1 Modelos de calidad de Software	42

2.10.2	ISO/IEC 25000	45
2.11	Estándar ISO/IEC 29110	47
2.11	Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte.....	50
2.11.1	Organigrama Dirección de Desarrollo Tecnológico e Informático.....	51
2.11.2	Herramientas de gestión	52
2.2.	Marco Legal	54
Capítulo III: Marco Metodológico		55
3.1	Lugar de desarrollo de la investigación	55
3.2	Tipo de Investigación	56
3.2.1	Investigación Cualitativa	56
3.2.2.	Investigación de Campo	56
3.2.3	Investigación Documental	57
3.3	Diseño de la Investigación	57
3.4	Población y muestra.....	58
3.4	Tipo de Método.....	58
3.5	Procedimiento	59
3.5.1	Análisis del Estándar ISO/IEC 29110	59
3.5.2.	Análisis de Situación Actual del Proceso de Desarrollo de Software	59
3.5.3.	Implementación de Estándar ISO/IEC 29110	60
CAPITULO IV: Resultados y Discusión.....		61
4.1.	Contextualización del estándar ISO/IEC 29110	61
4.1.1	Diagnóstico del uso de estándares y procesos en las PYMES de la industria de software ecuatoriana.....	61
4.2.	Proceso de Desarrollo de Software para los Proyectos CRM de la Dirección de Desarrollo Tecnológico e Informático	63
4.2.1	Propuesta de Proceso de Desarrollo de Software	64
4.2.2	Análisis del proceso de Desarrollo Dirección de Tecnologías de la Información e Innovación	65
4.2.2.1	Análisis de Roles.....	66
4.2.2.2	Resumen de Análisis de metodología RUP.....	67
4.2.2.3	Resumen de Análisis de metodología SCRUM	68
4.3	Implementación del Estándar ISO/IEC 29110	71
4.3.1	Análisis de Compatibilidad entre Metodología SCRUM – Estándar ISO/IEC 29110.....	71
4.3.2	Análisis de Método Híbrido ISO/IEC 29110 y Scrum	72
4.3.2.1	Roles.....	72

4.3.2.2	Documentación.....	73
4.3.2.3	Actividades.....	74
4.3.3	Modelo UTN-SCRUM	75
3	Roles del proyecto.....	75
4	Documentación que generada	76
5	Actividades desarrolladas el proceso.....	78
4.3.4	Implementación de Estándar ISO/IEC 29110	80
4.3.4.1	Gestión de proyecto.....	82
4.3.4.2	Implementación de Software.....	86
4.3.5	Implementación del estándar 29110 – Portafolio Móvil Estudiantil – UTN- móvil	93
4.3.6	Análisis de Impacto de implementación de estándar ISO/IEC 29110- perfil básico	97
4.3.6.1	Impacto Administrativo.....	98
4.3.6.2	Impacto Económico.....	98
4.3.6.1	Impacto Cultura - Organizacional	99
CAPITULO V: Conclusiones y Recomendaciones		100
Conclusiones.....		100
Recomendaciones		101
CAPITULO VI: Referencias Bibliográficas.....		102
Bibliografía		102
ANEXOS 1 – Metodología Scrum		105
Principios de SCRUM.....		105
Beneficios		105
Roles		107
1.	Scrum master	107
2.	Product owner (PO)	108
4.	Team	108
Documentos que generar en Scrum		108
Proceso.....		109
1.	Selección de requerimientos	110
5.	Planificación de la iteración.....	110
6.	Ejecución de la Iteración.....	111
7.	Inspección y adaptación.....	111
8.	Retrospectiva (máximo 4 horas).....	112
ANEXOS 2 – Contextualización del Estándar ISO/IEC 29110		113

Roles utilizados en la ISO/IEC 29110	115
Perfiles de la ISO/IEC 29110.....	116
Características de la ISO/IEC 29110	116
Beneficios del estándar ISO/IEC 29110	117
Descripción perfil del estándar ISO/IEC 29110	117
1. ISO/IEC 29110 -1	117
9. ISO/IEC 29110 -2	117
10. ISO/IEC 29110 -3.....	118
11. ISO/IEC 29110 -4-1	118
12. ISO/IEC 29110 -5-1-1	118
13. ISO/IEC 29110 -5-1-2.....	118
PROCESOS DEL PERFIL BÁSICO.....	118
GESTIÓN DE PROYECTO	119
IMPLEMENTACIÓN DE SOFTWARE	122
PRODUCTOS O ARTEFACTOS	125
PAQUETE DE DESPLIEGUE DEL PERFIL BÁSICO DE LA NORMA ISO/IEC20110.....	127
GENERACIÓN DE ARTEFACTOS DE ESTÁNDAR.....	128
Anexo 3 NORMAS DE CONTROL INTERNO.....	131
Anexo 4: Enunciado del Trabajo - Ágil.....	135
Anexo 5: Plan de Proyecto - ágil	140
Anexo 6: Especificación de requerimientos – ágil	146
Anexo 7: Diseño de arquitectura – ágil	149
Anexo 8: Resultados de Verificación (Check List).....	153
Anexo 9: Resultados de Validación (Check List)	156
Anexo 10: Solicitud de Cambios – ágil.....	158
Anexo 11: Acta de Aceptación – ágil.....	159
Anexo 12: Enunciado del Trabajo – Ágil.....	160
Anexo 13: Plan de Proyecto – Ágil	163
Anexo 14: Especificación de Requerimientos – Ágil.....	169
Anexo 15: Diseño de Arquitectura – Ágil.....	173
Anexo 16: Resultados de Verificación – Ágil.....	180
Anexo 17: Resultados de Validación – Ágil	183
Anexo 18: Análisis de Metodologías RUP Y SCRUM utilizada en la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte.....	186
Introducción	186

Metodología RUP.....	186
a) Fase de inicio.....	186
b) Fase de Elaboración.....	190
c) Fase de Construcción.....	195
d) Fase de Transición.....	199
Análisis de roles SCRUM.....	204
Metodología SCRUM	210
Anexo 19: acta de Aceptación de proyecto de Investigación.....	212

ÍNDICE DE TABLAS

Tabla 1: Característica Tipos de Procesos.....	29
Tabla 2: Procesos del Ciclo de Vida.....	30
Tabla 3: Integración con Sistemas.....	31
Tabla 4: Procesos del Ciclo de Vida.....	31
Tabla 5: Análisis de Metodologías Ágiles en DDTI	39
Tabla 6: Población y Muestra.....	58
Tabla 7: Estándares y Procesos de Software	62
Tabla 8: Similitudes entre proceso DDTI y RUP	66
Tabla 9: Roles definidos en La DDTI	67
Tabla 10 Consolidado análisis de situación actual	67
Tabla 11 Tabla de análisis detallada.....	70
Tabla 12: Tabla Resumen del análisis	70
Tabla 13: Diferencias entre estándar ISO/IEC 29110 y Metodología SCRUM.....	72
Tabla 14 Comparación de roles	73
Tabla 15: Comparativa de documentos	73
Tabla 16 Roles utilizados.....	75
Tabla 17: Roles y Responsabilidades	76
Tabla 18: Documentación U-Scrum.....	77
Tabla 19: Documentos Definidos para el método híbrido.....	77
Tabla 20: Propuesta de Método Híbrido - Gestión del Proyecto.....	78
Tabla 21: Propuesta de Método Híbrido - Ingeniería de Software.....	79
Tabla 22: Cuestionario Gestión de proyecto – planificación	83
Tabla 23 : Gestión de Proyecto - Ejecución del plan	84
Tabla 24: Gestión de Proyecto: Evaluación y control del proyecto	85
Tabla 25 Gestión de Proyecto - Cierre del proyecto.....	85
Tabla 26: Implementación del Software - Inicio de implementación del software.....	86
Tabla 27: Implementación de software - Análisis de los requerimientos de software	88
Tabla 28 Implementación de software - Diseño detallado y arquitectura del software....	90
Tabla 29: Implementación de software - Construcción del Software.....	90
Tabla 30 Implementación de software - Pruebas e Integración del software	91
Tabla 31 Implementación del software - Entrega del producto.....	92
Tabla 32: Servicios Expuestos.....	93
Tabla 33 Detalle de Herramientas	94
Tabla 34: Matriz de medición de impacto.....	97
Tabla 35 Tabla de indicadores Impacto Administrativo.....	98
Tabla 36: Indicadores impacto Económico	98
Tabla 37 : Indicadores de impacto Cultural - Organizacional.....	99
Tabla 38: Documentos SCRUM.....	109
Tabla 39: Definición de pequeña organización	113
Tabla 40: Roles del estándar ISO/IEC 29110 (Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110, 2013)	115
Tabla 41: Documentos de generación obligatoria (Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110, 2013)b	125
Tabla 42: Paquetes de despliegue de la ISO 29110.....	130
Tabla 43: Revisión de Condiciones de entrada	134

ÍNDICE DE FIGURAS

Figura 1: Proceso.....	25
Figura 2: Proceso de Software	28
Figura 3: Proceso de Software.	28
Figura 4: Modelo en Cascada.....	32
Figura 5: Modelo Incremental.....	33
Figura 6: Modelo en Espiral.....	34
Figura 7: La Agilidad y el costo del cambio	37
Figura 8: Características de Metodologías Ágiles.....	38
Figura 9 Proceso de Scrum.....	40
Figura 10 Proceso CMMI.....	43
Figura 11: Modelo Escalonado	44
Figura 12: Modelo Continuo	45
Figura 13: ISO 25000.....	46
Figura 14: Modelo de calidad ISO/IEC 25010.....	46
Figura 15: Estructura de MoProSoft	48
Figura 16: Publico objetivo ISO/IEC 29110	49
Figura 17: Orgánico Estructural UTN,.....	50
Figura 18: Ficha Institucional.....	51
Figura 19: Dirección de Desarrollo Tecnológico e Informático,	51
Figura 20: Resumen ERP,.....	53
Figura 21 CRM,.....	54
Figura 22 Tipos de Diseño no experimental.....	57
Figura 23: Proceso de Desarrollo de Software.....	64
Figura 24: Subproceso de Ingeniería de Requerimientos.....	65
Figura 25: Consolidado análisis de situación actual	68
Figura 26: Análisis Scrum.....	70
Figura 27: Comparación de Actividades	74
Figura 28: Pantalla de Login Demo	95
Figura 29: Menú de Opciones app	95
Figura 30: Procesando consulta al servicio web	96
Figura 31: Pantalla de Consulta Calificaciones.....	96
Figura 32: Roles de Scrum	107
Figura 33 Metodología Scrum.....	110
Figura 34: Estructura del estándar ISO /IEC 29110.....	114
Figura 35: Proceso del proceso del perfil básico.....	119
Figura 36: Diagrama del proceso de gestión de proyectos (ISO / IEC 29110).....	121
Figura 37: Diagrama del proceso de implementación de Software (ISO / IEC 29110)..	124
Figura 38: Conjunto de paquetes de implementación para apoyar el perfil básico de ingeniería de sistemas	126
Figura 39: Sitio de descarga del Paquete de Despliegue.....	128

LISTADO DE SIGLAS

SCRUM:	Metodología de Desarrollo Ágil
ISO:	International Organization for Standardization
PO:	Pequeña Organización
UTN:	Universidad Técnica del Norte
CMMI:	Capability Maturity Model Integration

RESUMEN

“IMPLEMENTACIÓN DEL ESTÁNDAR ISO/IEC 29110 EN EL PROCESO DE DESARROLLO DE SOFTWARE DE LA DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO DE LA UNIVERSIDAD TÉCNICA DEL NORTE.”

Autor: Edwin Ramiro Madruñero Padilla

Tutor: Ing. Jorge Caraguay Prócel, Msc

Año: 2018

La Gestión del Proceso de Desarrollo de Software una estructura principal en el proceso de desarrollo de software de una Pequeña Organización, contar con una definición de cómo realizar y gestionar cada una de las actividades de un proyecto de software para garantizar el éxito e incorporar atributos de calidad que diferencien el producto obtenido de los demás en el mercado es el objetivo principal de toda organización.

La Gestión del Proceso de Desarrollo de Software que se realiza en la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte es demasiado limitado y no cuenta con la formalidad en todas sus etapas de ejecución por lo que un proyecto de software es susceptible a sufrir cambios en cualquier instante retrasando el tiempo de entrega y dejando a criterio del cliente las características de satisfacción y de funcionalidad, el estándar ISO /IEC 29110 se encarga de atacar esta falencia en el proceso promoviendo la formalidad en todas las etapas del proceso de desarrollo mejorando la gestión del desarrollo permitiendo incorporar hitos de control en el proceso y su gestión para lograr una toma de decisiones oportuna en el proyecto.

Para la implementación del estándar que utilice la metodología SCRUM se generó un método híbrido que permite fusionar el estándar con la metodología logrando generar la documentación que la formalidad de un proyecto necesita sin limitar el entorno de ejecución de la metodología SCRUM

SUMMARY

“IMPLEMENTATION OF THE ISO / IEC 29110 STANDARD IN THE SOFTWARE DEVELOPMENT PROCESS OF THE TECHNOLOGICAL AND COMPUTER TECHNOLOGICAL DEVELOPMENT DEPARTMENT OF THE UNIVERSIDAD TÉCNICA DEL NORTE. ”

Author: Ing. Edwin Ramiro Madruñero Padilla

Tutor: Ing. Jorge Caraguay Prócel, Msc

Year: 2018

Software Development Process Management a main structure in the software development process of a Small Organization, having a definition of how to perform and manage each of the activities of a software project to guarantee success and incorporate attributes of quality that differentiate the product obtained from others in the market is the main objective of any organization.

The Management of the Software Development Process that takes place in the Technological and Computer Development Department of the Technical University of the North is too limited and does not have the formality in all its execution stages, so a software project is susceptible to undergo changes at any time by delaying the delivery time and leaving the criteria of satisfaction and functionality to the customer, the ISO / IEC 29110 standard is responsible for attacking this failure in the process by promoting formality in all stages of the development process improving the management of the development allowing to incorporate milestones of control in the process and its management to achieve an opportune decision making in the project.

For the implementation of the standard that uses the SCRUM methodology, a hybrid method was generated that allows merging the standard with the methodology, generating the documentation that the formality of a project needs without limiting the execution environment of the SCRUM methodology.

INTRODUCCIÓN

Todas las pequeñas organizaciones que se dedican al desarrollo de software deben establecer estrategias para lograr adentrarse en el mundo del mercado de software actual y ser competitivos con las grandes empresas de Desarrollo de Software.

El desarrollar un producto de software formado de dos etapas importante la Gestión y Planificación, el estándar ISO/IEC 29110 es una nueva serie de normas y técnicas de informe que alineado con la Ingeniería de Software está dirigido a las pequeñas organizaciones (con menos de 25 empleados), la mayoría PYMES

En la primera etapa del proyecto se realizó un estudio sobre el estándar, desde sus inicios hasta conocer en donde es aplicado a nivel continental como regional determinando que el estándar se encuentra más utilizado en Perú, Colombia, Brasil, este último ya cuenta con instituciones que ayudan a las pequeñas empresas a obtener la certificación.

En una segunda etapa se practicó un estudio a la Gestión del Procesos de Desarrollo de Software que cuenta la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte encontrando que para proyectos pequeños se encuentran apegados de manera no formal a la metodología SCRUM, metodología que ha sido utilizada para varios de los proyectos CRM como el Portafolio Estudiante, Docente, etc.

En una siguiente fase y después de conocer el estándar ISO/IEC 29110 y como se lleva la gestión del proceso de desarrollo de Software en la Dirección se realizó el análisis y se determinó que el estándar no se puede acoplar directamente a las metodologías ágiles, esta conclusión de igual manera apoyado en la investigación realiza por Latinoamérica a empresas como BitPerfect, en Perú por lo que apoyados en este estudio y en otras investigaciones por ejemplo en Colombia se desarrolló un método híbrido de integración del estándar con la metodologías y se elaboraron las plantillas necesarias para su aplicación, plantillas que se encuentran relacionadas con el paquete en puesta en marcha que ofrece el estándar para su aplicación.

Y como una fase final después de tener las plantillas elaboradas se procedió a aplicarla en un proyecto que fue un prototipo de aplicación móvil que consultas las calificaciones actuales del estudiante.

Finalmente se plantearon las conclusiones y recomendaciones del proyecto de investigación que darán inicio a la continuidad y mejoramiento de la puesta en marcha del

estándar en la Dirección de Desarrollo Tecnológico e Informático de la Universidad
Técnica del Norte

CAPÍTULO I: EL PROBLEMA

1.1 ANTECEDENTES

El mercado global de tecnologías de información (TI) se estimó en USD 3.8 millones de millones para el año 2016, siendo EE.UU. el mayor actor (con una participación de 28%), si bien la región que muestra el mayor crecimiento es Asia-Pacífico gracias al impulso dado por China. La mayoría del gasto en TI es efectuada por las empresas (para actividades de comunicaciones en países emergentes, mientras en los países desarrollados el rubro de mayor gasto es software). A nivel mundial, el gasto en software y servicios informáticos sumó USD 1.2 millones de millones en 2011 (80% en América del Norte y Europa). (Acebo & Nuñez, 2017)

De acuerdo al estudio realizado por (Muñoz, Gasca, & Valtierra, 2014) a; las Pymes desarrolladoras de software en varios países de América Latina (esto incluye 16 pymes de Colombia y 89 pymes en todo el mundo) se evidencia que las empresas utilizan modelos y estándares del ciclo de vida del software: el 66% hace referencia a la ISO 12207, el 26% a RUP y el 8% a la ISO 29110; este mismo estudio muestra resultados, los cuáles son los modelos y estándares enfocados en las Pymes y el 43% menciona a MesoPyme, el 29% a MoProsoft, el 14% a MSP y el restante 14% a la ISO 29110. (Valdebenito Espinosa, Guía Metodológica para la implementación del estándar ISO/IEC 29110 en Pymes TICs de Colombia, 2016).

Según (Laporte & O'Connor, Deploying Lifecycle Profiles for Very Small Entities: An Early Stage Industry View, 2011) indica que la implementación del estándar ISO/IEC29110 permite a las pequeñas empresas pueden obtener beneficios derivados de la implementación, especialmente diseñado para mejorar la gestión del proceso de desarrollo de software, además de acuerdo a (Nyce, 2014), Estados Unidos, México, Canadá, Irlanda, España, Perú Brasil, Tailandia y Japón se encuentran implementando programas para la adopción y certificación de la norma ISO/IEC 29110.

Brasil está en un proceso de acreditación como ente certificador, con tres empresas, generando auditores capaces de certificar en esta norma (Laporte, Séguin, Villas Boas, & Buasung, 2013), Perú es también un país de Latinoamérica donde en base al estándar

ISO/IEC 29110, se promulgó la norma técnica peruana NTP-RT-ISO/IEC TR 29110-5-1-2 (INDECOPI, 2012). Colombia tiene que Parquesoft por medio de Liliana Gómez Arenas CEO de GreenSQA S.A participó en la elaboración de uno de los paquetes de implementación de la norma ISO/IEC 29110, el proceso documentado fue “Deployment Package Software Testing and Integration - Basic Profile” (Valdebenito Espinosa, Guía Metodológica para la implementación del estándar ISO/IEC 29110 en Pymes TICs de Colombia, 2016)a. Además la implementación del estándar, en la empresa Sequal Knowledge & Practice ubicada en Medellín , permite un control a lo largo de las fases del desarrollo del proyecto e incluir hitos para mejorar la gestión del proceso de desarrollo, permitiendo garantizar puntos de evaluación y analizar posibles incidencias en la gestión del proceso de desarrollo que afecten el normal desenvolvimiento del proyecto, además permite la toma de decisiones en caso de que el proyecto sufra alguna alteración en sus etapas de desarrollo.

En Ecuador la industria de software genera ventas del orden de USD 500 millones (0.5% del PIB), con un crecimiento anual de 17% en los 7 años previos. La actividad más importante es la provisión de servicios informáticos (53%), al tiempo que las ventas de software al sector público representan 22% del total. La industria local, integrada por alrededor de 700 empresas, está conformada mayormente por Pymes, mostrando además una fuerte concentración de ingresos en las empresas de mayor tamaño. En 2015 proporcionó empleo a más de 10 mil personas, principalmente desarrolladores (35%) e implementadores (24%) (Acebo & Nuñez, 2017)b.

En la provincia de Imbabura, ubicada en el norte del Ecuador se encuentra ubicada la Universidad Técnica del Norte, en donde la Dirección de Desarrollo Tecnológico e Informático tiene un proceso de desarrollo de software para los proyectos CRM, planteados en base a la experiencia empírica del PROYECT MANAGER; el cual aplica conceptos de planificación, control y estandarización en el proceso de manera Ad-Hoc, sin conocimientos profundos de una normativa que brinde las directrices de planificación y gestión de un proceso.

El presente trabajo de investigación pretende ser un aporte significativo al proceso de desarrollo de software que tiene la Dirección de Desarrollo Tecnológico e Informático del Norte para los proyectos CRM que se desarrolla, garantizando la estandarización de la

gestión del proceso de desarrollo e incidiendo de manera directa a la calidad del producto esto con la implementación del estándar. Los beneficios de manejar un proceso de desarrollo estandarizado es brindar las directrices necesarias para la gestión correcta del proceso de un proyecto y apoye a la generación de la documentación necesaria, contribuyendo de manera sistemática a la planificación del proyecto

Los beneficios de la implementar un estándar en la gestión del proceso de desarrollo de software es:

- Proporciona al equipo de desarrollo de la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte tener estrategias que permitan incorporar atributos de calidad al producto.
- Incorporar controles al proceso de desarrollo de Software que incidan en la gestión del proyecto de software y en la toma de decisiones.
- Orientación a las microempresas en el establecimiento de procesos de ingeniería de software con normas internacionales.
- Estandarización del proceso de desarrollo de software para los proyectos CRM.
- Establecimiento de lineamientos para certificación en la Norma ISO/IEC 29110.

1.2. OBJETIVOS DE LA INVESTIGACIÓN

1.2.1. OBJETIVO GENERAL

Implementar el estándar ISO/IEC 29110-4-1, aplicando el Paquete de Puesta en Operación para mejorar la gestión del proceso de desarrollo de software en la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte.

1.2.2. OBJETIVO ESPECÍFICO

- Contextualizar el perfil básico del estándar ISO/IEC 29110 en el proceso de desarrollo de software para los proyectos CRM.
- Establecer la matriz de análisis de los procesos de desarrollo de software para los proyectos CRM de la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte.
- Implementar el estándar ISO/IEC 29110-4-1 al proceso de desarrollo de software para los proyectos CRM de la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte.

1.3 PROPOSICIÓN

La aplicación del estándar ISO/IEC 29110 en el proceso de Desarrollo de Software para los proyectos CRM de la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte, incide en la la gestión del proceso de desarrollo de software, mejorando la calidad del producto software.

1.4. PREGUNTAS DIRECTRICES

- 1) ¿Cuáles son las normas que establece el estándar ISO/IEC 29110, para el desarrollo de proyectos de software y sean aplicables al desarrollo de proyectos CRM?
- 2) ¿Cuál o cuáles estándares aplican en el desarrollan los proyectos CRM de software en la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte?
- 3) ¿Cuál es el impacto de la implementación de un estándar al proceso de desarrollo de software?

CAPÍTULO II: MARCO REFERENCIAL

2 MARCO TEÓRICO

2.1 GESTIÓN DE PROCESO

La Gestión de Procesos permite mantener un control, organización y sincronización del funcionamiento del proceso de Desarrollo de Software, según Juan Bravo indica que la gestión de procesos es una disciplina de gestión que ayuda a la dirección de la empresa a identificar, representar, diseñar, formalizar, controlar, mejorar y hacer más productivos los procesos de la organización para lograr la confianza del cliente. La estrategia de la organización aporta las definiciones necesarias en un contexto de amplia participación de todos sus integrantes, donde los especialistas en procesos son facilitadores. (Bravo Carrasco, 2011)

El objetivo de la gestión de procesos es aumentar la productividad de la Dirección de Desarrollo de Tecnológico e Informático; la productividad es la eficiencia y eficacia de un proceso y agregando valor para el cliente final.

La Gestión de procesos cuenta con las siguientes características:

- a) Considerar siempre al cliente.
- b) Tiene determinada la finalidad del esfuerzo para conocer el esfuerzo realizado y el grado de resultados.
- c) Satisface las necesidades del cliente.

2.2 PROCESO

Según la ISO, “Un proceso es un conjunto de actividades que están interrelacionadas y que pueden interactuar entre sí. Estas actividades transforman los elementos de entrada en resultados, para ello es esencial la asignación de recursos”. (ISO, 2014), así también (Bravo Carrasco, 2011) nos menciona que un proceso es una totalidad que cumple un objetivo útil a la organización y que agrega valor al cliente.

Los procesos de una organización aportan valor al giro del negocio y son planificados para luego ser puestos en práctica en circunstancias controladas. Una consideración adicional es que los elementos de entrada pueden ser resultados de otros procesos. De manera similar, el Diccionario de Inglés Americano de Cambridge define proceso como “una serie de acciones que se realizan para conseguir un resultado”. Esta última definición incluye el concepto de personas junto a actividades y resultados, permitiendo que los resultados puedan ser orientados a productos, personas, procesos, y servicios.

La Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte cuenta con procesos levantados y automatizados evidenciados en su sistema ERP, pero el proceso de desarrollo de software se encuentra levantado, pero sin formalizar lo cual nos ayuda a la implementación de un estándar.

Un proceso es aquel que tiene una entrada, actividades que realizan algún tratamiento de la información de entrada y que provee una salida.



Figura 1: Proceso

Fuente: (ISO, 2014)

Elaborado por: El Autor

El proceso debe tener las siguientes características:

- a) Debe de ser entendible;
- b) Tener Visibilidad, grado que las actividades puedan proporcionar resultados;
- c) Soportable para las herramientas case;
- d) Aceptabilidad, que las personas lo acepten y lo usen;
- e) Fiabilidad;
- f) Robustez;
- g) Mantenable;

h) Rapidez;

2.3 SOFTWARE

El software de computadora es el producto que construyen los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo. (Pressman, 2012), también Sommerville menciona que el software no solo son programas, sino todos los documentos asociados y la configuración de datos (SOMMERVILLE, 2006). El software se almacena en la memoria del computador y es posible que sea modificado para adaptarse a las necesidades de quien lo usa, se puede clasificar en compiladores y otros programas traductores (aplicaciones de escritorio, aplicaciones web, aplicaciones móviles, entre otros).

Los compiladores son herramientas que traducen un lenguaje de alto nivel, como Java, si Sharp C#, Phyton, PHP al lenguaje que utiliza la computadora, código binario (Sistema de numérico de ceros y unos que la computadora entiende e interpreta). Los compiladores facilitan la labor de los programadores, pueden crear otro Software usando un lenguaje más semejante al humano en lugar del lenguaje de máquina. Las aplicaciones de Software son herramientas que ayudan a los usuarios a resolver un problema en particular, ayudan a las personas sin muchos conocimientos de programación a comunicarse con una computadora y que esta responda a sus necesidades. Estas aplicaciones pueden simular y mejorar herramientas que existen en la vida real, desde las más sencillas hasta las más complejas.

El software ha sido una herramienta para los procesos automatizados por la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte, esto por cuanto ha permitido brindar servicios a la comunidad universitaria y difundir información al mundo; los procesos de matrículas, consulta de notas y entornos virtuales de aprendizaje son paquetes desarrollados dentro de la institución que han facilitado de manera permanente la interacción con la planta estudiantil.

2.4 INGENIERÍA DE SOFTWARE

La Ingeniería de Software es construir productos de software y/o mejorar los existentes realizando una similitud y aplicando las mejores prácticas existente, a su vez la ingeniería de procesos tiene como objetivo desarrollar nuevos procesos y mejorar los existentes (Ruvalcaba, 2017), así también Sonia Mariño define la ingeniería como una disciplina esencialmente objetiva, que se apoya en leyes naturales, resultados experimentales y fórmulas empíricas para proponer y respaldar soluciones a los problemas que se le presentan. Para ello, se inspira en enfoques que podrían considerarse versiones tecnológicas del método científico, en el que las evidencias ocupan claramente un lugar central. (Mariño & Alfonzo, 2017), así también se puede definir a la Ingeniería de Software como un estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software.

La Ingeniería de Software es una parte importante del desarrollo de aplicaciones de software que nos ayuda con lineamientos a seguir para que un proyecto de software y su no fracaso, la Dirección de Desarrollo Tecnológico e Informático sigue estrechamente las sugerencias descritas en la ingeniería de software.

2.5 PROCESO DE SOFTWARE

Un proceso de software tiene como fin la producción eficiente y eficaz de un producto de software que satisfaga las necesidades del cliente o usuario final. Un producto de software es complejo debido a que no podemos llegar a tener el 100% de confiabilidad de un programa, uno de los factores que incide es que es un producto intangible y para lograr mantener un porcentaje adecuado de confiabilidad que podamos ofrecer al cliente se deben seguir las normas y recomendaciones publicadas por las entidades certificadas en producción de software.

Según Fuggetta define al proceso de software como “Un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software”. (Fuggetta, 2000).

A continuación, se muestra un diagrama de proceso de software que indica cuál es la interacción entre las actividades, desarrolladores, producto, herramientas, etc.

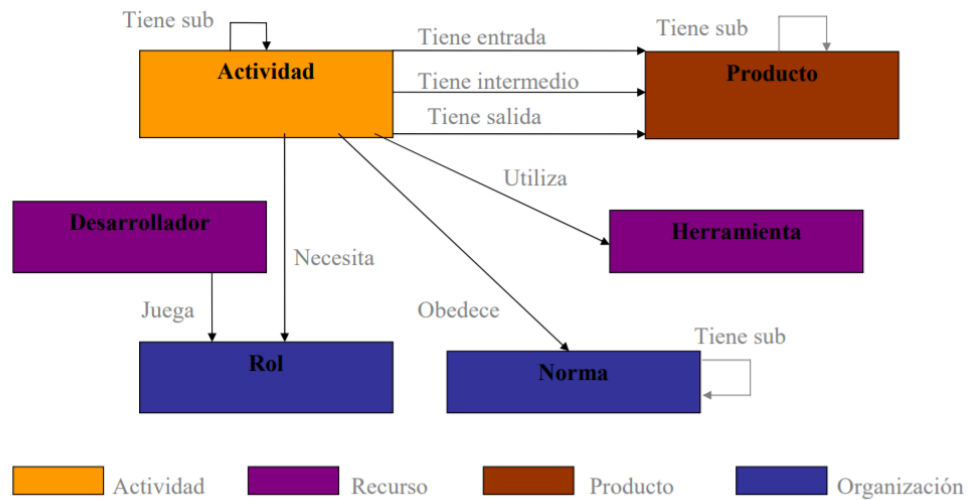


Figura 2: Proceso de Software

Fuente: (Ruiz)

Elaborado por: (Ruiz)

Los procesos de software contribuyen a la creación de aplicaciones informáticas que cuenten con atributos de calidad como son portabilidad, seguridad, funcionalidad, usabilidad, entre otros.

Los procesos de software cuentan con ciertas características que brindan características propias de un proceso de software, estas son:

- Permite estandarizar esfuerzos, promover el reusó, repetición y consistencia entre proyectos;
- Provee la oportunidad de introducir mejores prácticas de la organización;
- Permite entender que las herramientas del cómo deben ser utilizadas para soportar un proceso;
- Establece la base para una mayor consistencia y mejoras futuras.

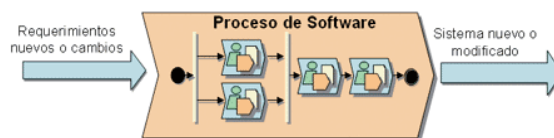


Figura 3: Proceso de Software.

Fuente: (Ruvalcaba, 2017)

Elaborado por: (Ruvalcaba, 2017)

Los procesos de software tienen relación con otros procesos que no sean los de software como procesos industriales, proceso de información, procesos de negocio, pero en realidad se encuentran relacionados tal como indica la siguiente tabla:

Tipos de Proceso	Procesos Industriales	Procesos de Información	Procesos de Negocio
Enfoque	Cosas	Datos	Relaciones
Propósito	Ensamblar, transformar componentes en obteniendo como resultado productos finales.	Procesa y transmite datos estructurados y no estructurados.	Promueve las condiciones que satisfacen las necesidades de los participantes.
Características	Pertenece a la Ingeniería Industrial	Pertenece a la Ingeniería Informática	Basada en la forma de expresión humana, encontrada en todas las lenguas
Acciones	Ensamblar, Transformar, Almacenar, Inspeccionar	Enviar, Invocar, Grabar	Solicitar, prometer, cancelar, medir

Tabla 1: Característica Tipos de Procesos

Fuente: (ISO 27207, 2015)

Elaborado por: El Autor

Como podemos apreciar en la tabla Nro. 1 existen procesos industriales, de información y de negocio y cada uno de ellos tienen sus particularidades, pero el proceso de software cubre y se relaciona con todos ellos de manera conjunta para obtener mejores resultados.

El proceso de software se encuentra relacionado con el ciclo de vida del desarrollo de software, un concepto de ciclo de vida es el que nos indica la IEE1074 donde menciona que es “una aproximación lógica a la adquisición, suministro, el desarrollo, la explotación y el mantenimiento del software”, pero en cambio el estándar ISO12207 menciona que es “un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso” (ISO 27207, 2015).

Los procesos de ciclo de vida detallados en la norma ISO 27207 se organizan de la siguiente manera:

PROCESO	SUBPROCESOS
PROCESOS PRINCIPALES	<ul style="list-style-type: none"> • Adquisición; • Suministro; • Desarrollo; • Exploración; • Mantenimiento.
PROCESOS ORGANIZACIONALES	<ul style="list-style-type: none"> • Gestión; • Infraestructura; • Mejora; • Recursos Humanos; • Gestión de Activos; • Ingeniería de Dominio.
PROCESOS DE SOPORTE	<ul style="list-style-type: none"> • Documentación; • Gestión de configuración; • Aseguramiento de calidad; • Verificación; • Validación; • Revisión conjunta; • Auditoria; • Resolución de problemas; • Usabilidad; • Evaluación de productos.
PROCESOS DE ADAPTACIÓN	

Tabla 2: Procesos del Ciclo de Vida

Fuente: (ISO 27207, 2015)

Elaborado por: El Autor

Como se indicó anteriormente existe una relación entre el contexto de los procesos y los procesos de software, se tiene una integración entre estos dos macro procesos que se describe a continuación en la siguiente imagen:

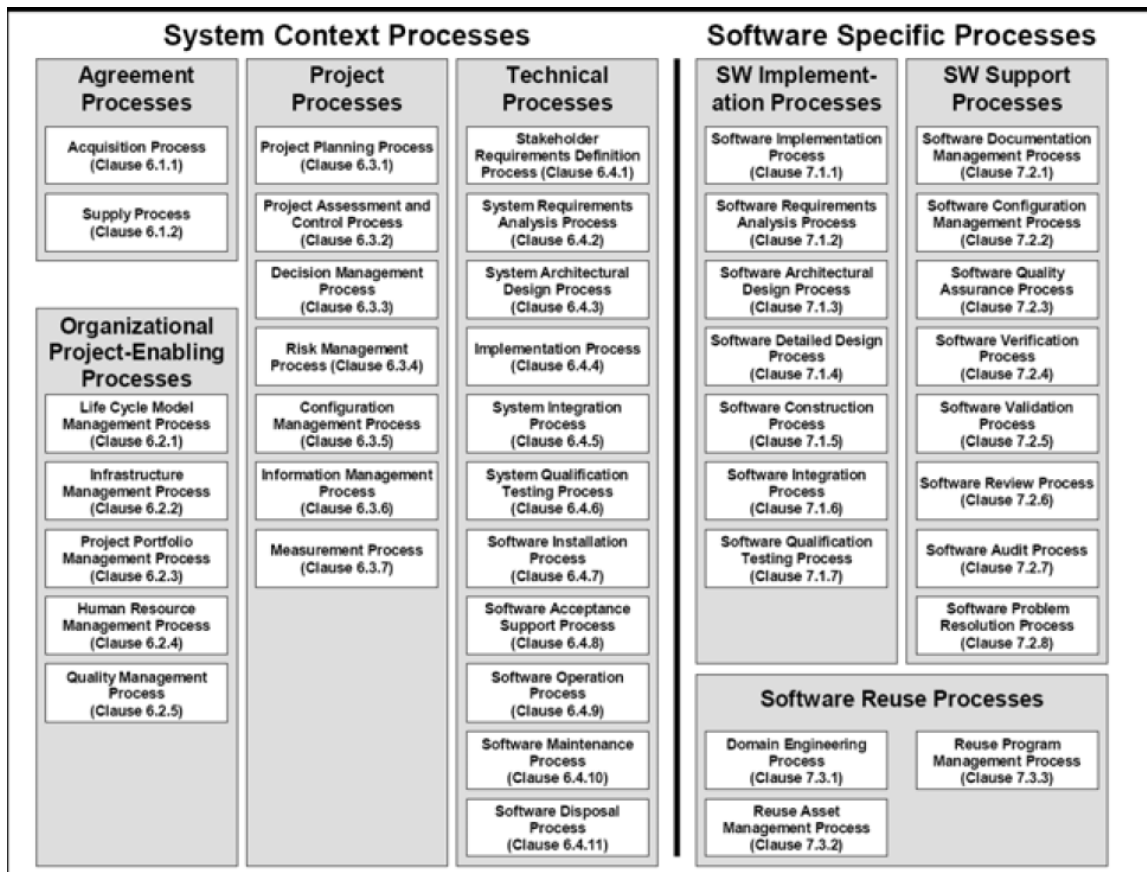


Tabla 3: Integración con Sistemas

Tabla 4: Procesos del Ciclo de Vida

Fuente: (ISO 27207, 2015)
Elaborado por: (ISO 27207, 2015)

Cada uno de los procesos de software se encuentran relacionados con los ciclos de vida tradicionales para el desarrollo de software los más conocidos son: En cascada, Incremental, En Espiral, Prototipado

- MODELO EN CASCADA

El modelo en cascada contiene las siguientes características:

- Cada fase empieza cuando finaliza la anterior
- Para pasar de una fase a otra es necesario conseguir todos los previos de la etapa anterior
- Ayuda a prevenir que se sobrepasan las fechas de entrega y los costos esperados
- Al finalizar cada fase el personal técnico puede revisar el progreso del proyecto.

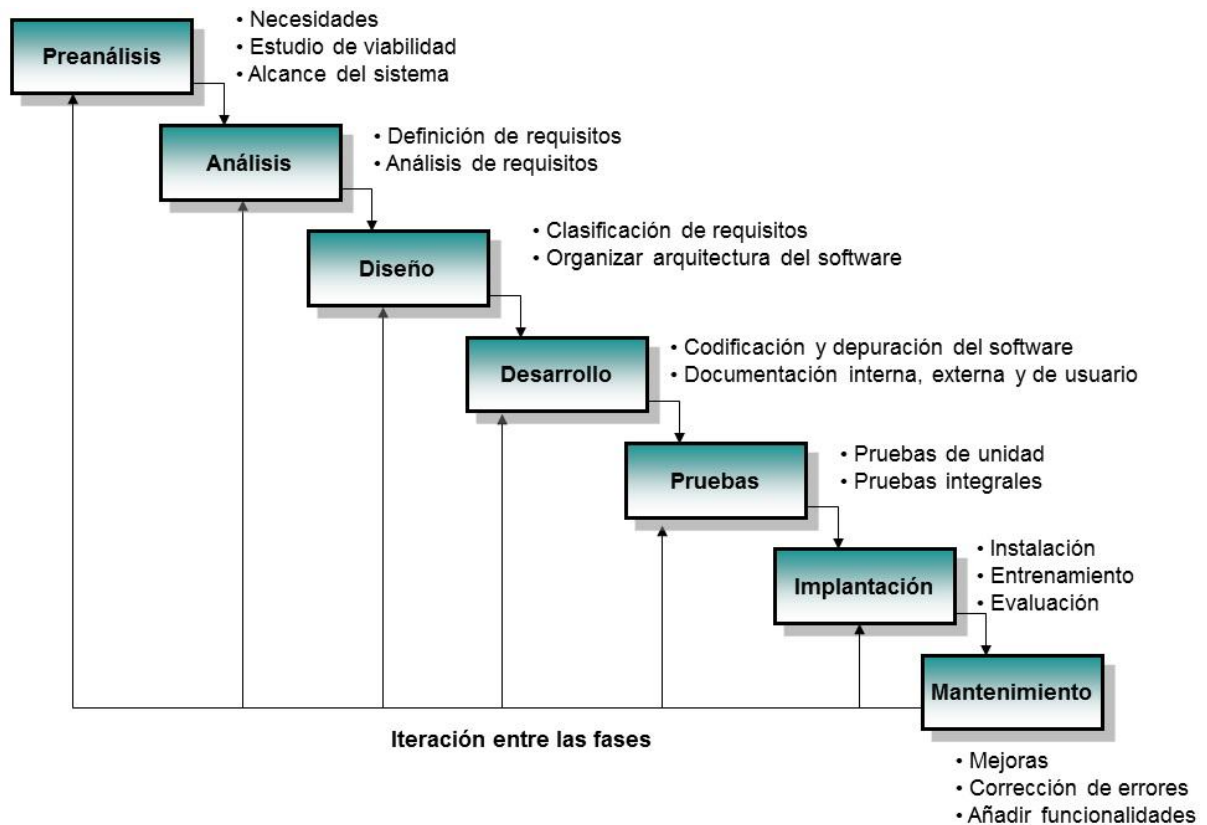


Figura 4: Modelo en Cascada

Fuente: <http://imprukogico.blogspot.com/2014/06/ciclo-de-vida-del-software-modelo-de.html>

Elaborado por: Prukogi BM

- MODELO INCREMENTAL

El modelo incremental tiene las siguientes características:

- Corrige las necesidades de una secuencia no lineal de pasos de desarrollo
- El sistema se crea añadiendo requerimientos funcionales al sistema
- El sistema no tiene una estructura monolítica, con fechas fijas de entrega si no que el producto sale incrementalmente

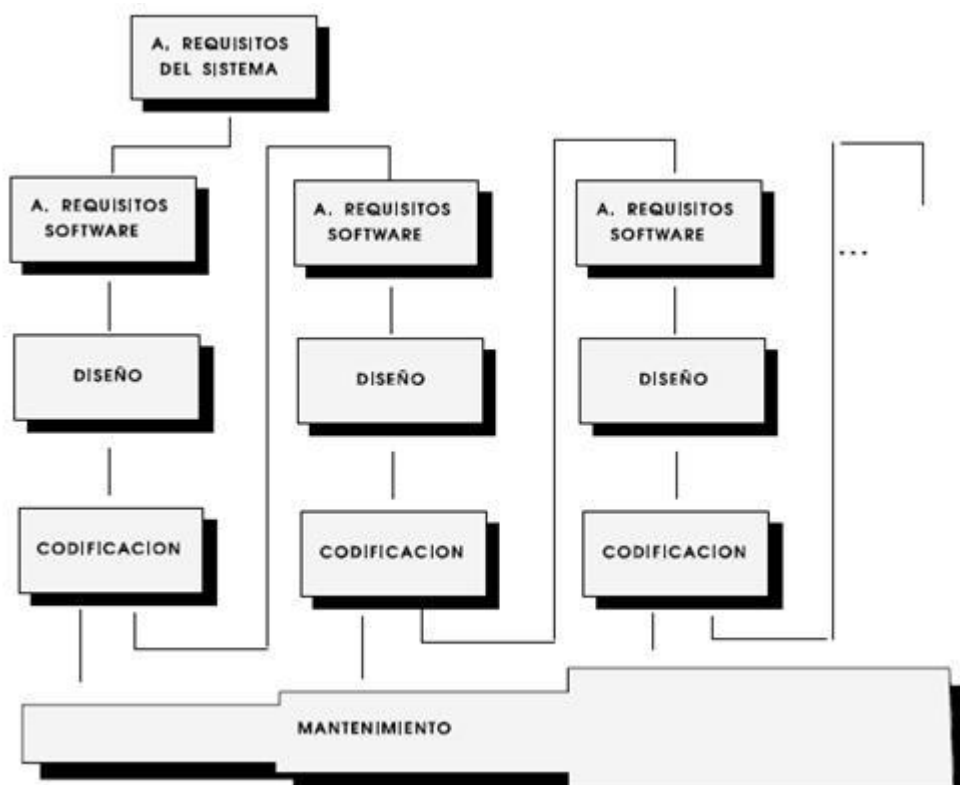


Figura 5: Modelo Incremental

Fuente: <https://isoft3cv2.wordpress.com/>

Elaborado por: <https://isoft3cv2.wordpress.com/>

- **MODELO EN ESPIRAL**

El modelo de proceso evolutivo, cambiando la naturaleza de construcción de software a ser más iterativa, construyendo prototipos con los aspectos controlados, según (Pressman R. S., 2012) indica que es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada

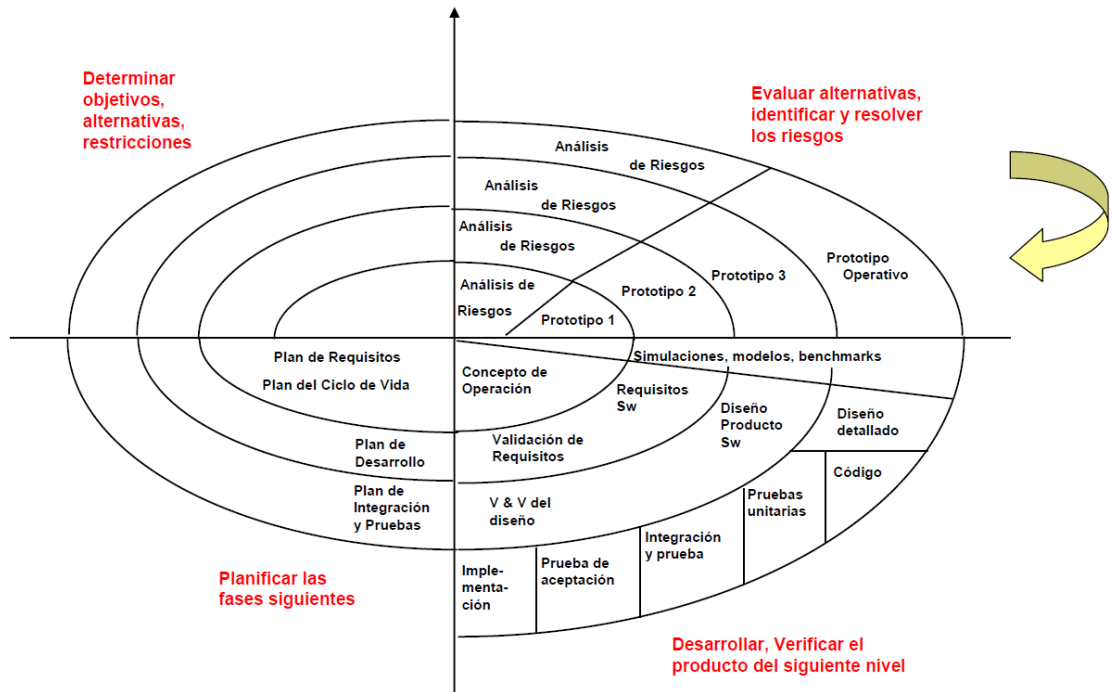


Figura 6: Modelo en Espiral

Fuente: (Pressman R. , 2010)

Elaborado por: (Pressman R. S., 2012)

2.6 PROYECTOS DE SOFTWARE

Los proyectos de software desarrollados en la Universidad Técnica del Norte se encuentran publicados y brindando el servicio a la comunidad universitaria, es por ello que definimos como un proyecto de software el cual tiene como objetivo crear un producto, el cual cumpla con las expectativas y necesidades del cliente (usuario final); este proyecto es exitoso cuando alcanza las metas en el tiempo, presupuesto planificado y cuenta con atributos de calidad sin extender estos tres aspectos. Para alcanzar dicho éxito, es recomendable Gestionar el Proyecto de Software de la manera adecuada, tomando en consideración las reglas, normativas existentes y buenas prácticas estandarizadas. Los

Proyectos de Software pueden variar mucho entre ellos, sin embargo, las metas más importantes para cualquier proyecto son las siguientes:

- Entregar el software al cliente en el tiempo establecido;
- Mantener los costos generales dentro del presupuesto;
- Entregar un producto de software que cumpla con las necesidades y expectativas de cliente;
- Contar con un equipo de desarrollo feliz y en correcto funcionamiento.

Para alcanzar estas metas, hay que considerar que la gestión de un proyecto de software es algo más retadora que en otros tiempos debido a:

- **El Software es intangible**, no puede ser visto o tocado y, en consecuencia, no es posible evidenciar el progreso del proyecto a simple vista. El gestor de proyecto de Software recurre a otras personas para obtener evidencia del avance y de esta forma revisar el progreso respecto al plan inicial.
- **Un proyecto de Software y las circunstancias de este solo ocurren una vez**, especialmente si es un proyecto grande, es probable que sucedan cambios tecnológicos en el contexto del proyecto, lo cual puede resultar en que la experiencia de un gestor de proyectos se torne obsoleta. En otras palabras, los proyectos de Software varían con respecto a proyectos anteriores en distintos aspectos por lo cual es difícil replicar una gestión anterior. De igual manera, las lecciones aprendidas de un proyecto pasado pueden ser no aplicables a un proyecto nuevo.
- **La última razón es la variabilidad y especificidad de los procesos de Software**. Cada organización maneja sus propios procesos de Software por lo que varían significativamente, a pesar de existir normas para la estandarización y mejora de los procesos, aún no es posible predecir en qué momento y bajo cuáles circunstancias un proceso de Software puede acarrear problemas.

2.7 METODOLOGÍA

La metodología de software es un enfoque, una manera de interpretar la realidad o la disciplina en cuestión, que en este caso particular correspondería a la Ingeniería de Software; es así que la metodología utilizada para el desarrollo de software es considerada como una estructura para planificar y controlar el procedimiento de creación de aplicaciones informáticas. (Gomez, 2017).

Conocer sobre que significa una metodología y como se integran en el proceso de desarrollo de un proyecto de software aporta a conocer desde donde inicia la planificación de un proyecto general y e enfoque que brinda Scrum como una metodología de gestión de proyectos adaptada al desarrollo de software de manera ágil.

2.8 METODOLOGÍAS AGILES

Según (Pressman R. S., 2012), el desarrollo ágil combina una filosofía con un conjunto de lineamientos de desarrollo. La filosofía pone el énfasis en: la satisfacción del cliente y en la entrega rápida de software incremental, los equipos pequeños y muy motivados para efectuar el proyecto, los métodos informales, los productos del trabajo con mínima ingeniería de software y la sencillez general en el desarrollo. Los lineamientos de desarrollo enfatizan la entrega sobre el análisis y el diseño (aunque estas actividades no se desalientan) y la comunicación activa y continua entre desarrolladores y clientes. (Pressman R. S., 2012).

Las metodologías ágiles juntamente con los Ingenieros de Software y demás participantes de un proyecto (gerente, clientes, usuarios, etc.) trabajan conjuntamente en el proyecto ágil formando una alianza para encaminarse a los resultado o producto final; esto es importante porque el ambiente de negocio que un modelo de negocio basado en software evolucionando rápidamente, es así que una metodología de desarrollo ágil cumple las siguientes actividades comunicación, planeación, modelado, construcción y despliegue las cuales son fundamentales y apoyan a obtener el producto final.

Conocer que el producto está bien desarrollado es primordial y en el desarrollo ágil la única forma es realizar pruebas unitarias y después medir el grado de satisfacción del cliente.

A continuación, se muestra en una figura donde se evidencia costo del cambio y de desarrollar con una metodología tradicional y una metodología ágil

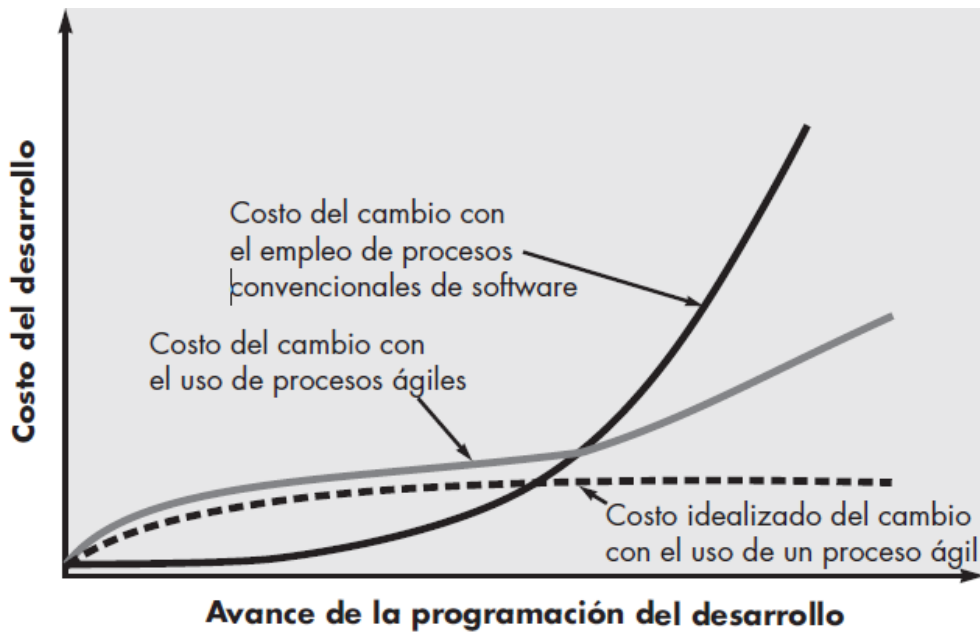


Figura 7: La Agilidad y el costo del cambio

Fuente: (Pressman R. , 2010)

Elaborado por: (Pressman R. S., 2012)

Las metodologías ágiles se fundamentan en 12 principios de agilidad para aquellos que la quieran alcanzar:

1. La prioridad más alta es satisfacer al cliente
2. Son bienvenidos los requerimientos cambiantes, aunque en etapas avanzadas el agilismo los denomina cambios para beneficio del cliente
3. Entregar con frecuencia software que funcione
4. Las personas de negocio y los desarrolladores trabajan juntos
5. Hay que desarrollar con personas motivadas
6. El método más proactivo para transmitir la información es mediante el equipo de desarrollo.
7. La medida principal del software es que funcione
8. Los procesos ágiles promueven el desarrollo sostenible
9. La atención continua promueve a la excelencia técnica
10. Es necesario la simplicidad
11. Las mejores arquitecturas, requerimientos y diseños surgen en los propios equipos de la organización

12. El equipo reflexiona a intervalos regulares sobre ser eficaz.

A continuación, se indica en la siguiente imagen una tabla donde se indica las principales diferencias entre las metodologías ágiles

- SCRUM (Metodologías Scrum);
- XP (Metodología Extreme Programming);
- DSDM (Metodología Dynamic Systems Development Method);
- FDD (Metodología Desarrollo Basado en Funciones);
- ASD (Metodología Desarrollo de Software Adaptable);
- Crystal (Metodología Crystal)

Table 1 Comparison of Agile Methodologies

Characteristic	XP	Scrum	DSDM	FDD	ASD	Crystal
Development approach	Iterative increments	Iterative increments	Iterative	Iterative	Iterative	Incremental
Recommended iteration time period	One to six weeks	Two to four weeks	80% solution in 20% total time	Two days to two weeks	Four to eight weeks	Depending on method from the family
Project team	Smaller teams Less than twenty members	All sizes (Scrum of Scrums concept)	All sizes Independent team	Many members More than one team	Smaller teams Five to nine team members	All sizes Depending on method from the family
Team communication	Informal Daily stand-up meetings	Informal Daily stand-up meetings	Documentation based	Documentation based	Informal Face-to-face	Informal Face-to-face
Project size	Smaller projects	All types of projects	All types of projects	More complex projects	Smaller projects	All types of projects Depending on method from the family
Customer involvement	Customer involved	Customer through the role of Product Owner	Customer through frequent releases	Customer through reports	Customer through releases	Customer through incremental releases
Project documentation	Only basic documentation	Only basic documentation	Documentation exist	Documentation is important	Only basic documentation	Only basic documentation
Specialties	TDD, User stories, Refactoring	Sprint, Product and Sprint backlog, Planning Poker, Scrum master	Prototyping	UML diagrams	Learning cycle	Adaptable methods family, All types of projects and team sizes
Advantages	Open workspace, customer as a part of the team, well defined best practices, feedback	High level of communication and collaboration	Requirements priority approach, efficient project management	Reports and documentation enables multitasking	Development of high risk components first, importance of learning cycle	Methodologies that adjusts to project type and size
Disadvantages	Weak documentation, lack of discipline, customer presence is mandatory	Weak documentation, poor control over project	Complex documentation	Individual code ownership, not applicable to smaller projects	Poor method documentation	Efficient coordination of bigger teams

Figura 8: Características de Metodologías Ágiles

Fuente: (Flores , 2014)

Elaborado por: (Flores , 2014)

Según la investigación realizada se realizó un análisis entre todas las metodologías ágiles, tradicionales como RUP y enfocadas al proceso de desarrollo que se plantea para la Dirección de Desarrollo Tecnológico e Informático.

	DDTI	SCRUM	XP	DSDM	FDD	ASD	RUP
Ingeniería de requerimientos							
	Está enfocado a proyectos de cualquier tipo	√	X	√	X	X	X
	Genera documentación básica	√	√	√	√	√	√
	Equipo de trabajo de todos los tamaños	√	X	X	X	X	X
	Rapidez en comunicación entre miembros del equipo	√	√	X	X	√	X
	Reuniones de seguimiento constantes	√	√	X	X	X	X
	Alto nivel de Colaboración	√	X	X	X	X	X
Construcción o Codificación							
	El equipo puede ser multidisciplinario	√	√	√	√	√	√
	Utiliza una Técnica de desarrollo Guiado por Pruebas	√	√	X	X	X	X
	Rapidez en comunicación entre miembros del equipo	√	√	X	X	√	X
Despliegue de Producto							
	Entrega de productos progresivamente con el cliente	√	√	√	√	√	X
	Resumen	100%	70%	40%	30%	50%	0%

Tabla 5: Análisis de Metodologías Ágiles en DDTI

Fuente: (Flores , 2014)

Elaborado por: El Autor

Donde cómo podemos ver por la naturaleza de los proyectos que la Universidad propone Scrum es la metodología que mejor se adapta a las necesidades institucionales con un 100% de compatibilidad, seguida de la Metodología Extreme Programming con un 70%, siendo las dos metodologías las mejores puntuadas, pero para objeto de nuestro estudio vamos a seleccionar la metodología Scrum.

2.9 METODOLOGÍA SCRUM

Un proyecto Scrum implica un esfuerzo de colaboración para crear un nuevo producto, servicio, o cualquier otro resultado como se define en la Declaración de la visión del proyecto. Los proyectos se ven afectados por las limitaciones de tiempo, costo, alcance,

calidad, recursos, capacidades organizativas, y otras limitaciones que dificultan su planificación, ejecución, administración y, finalmente, su éxito. (Satpathy, 2016)

Las metodologías ágiles representan una alternativa para el desarrollo de sistemas de software, centrada en el factor humano y el producto software, valorizando la relación con el cliente y el desarrollo incremental del software. (Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110)

Define un marco para gestión de proyectos, el desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 Minutos del equipo de desarrollo para coordinación e integración (Becerra & Sanjuan, 2014).

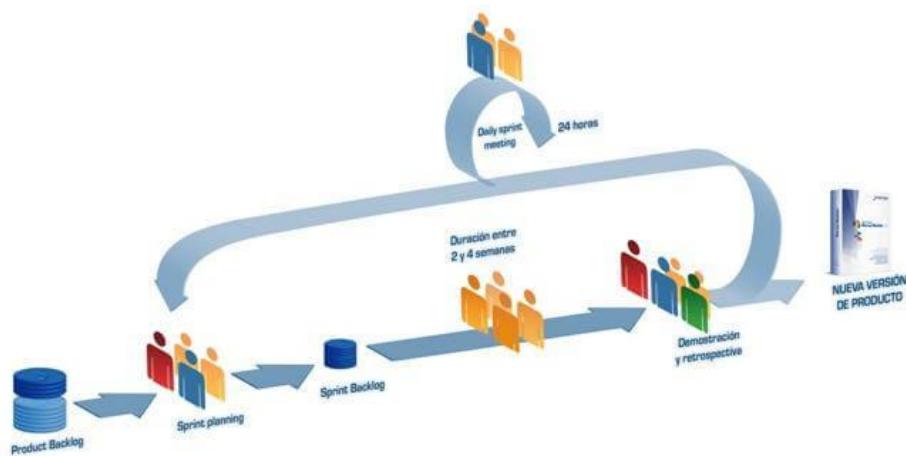


Figura 9 Proceso de Scrum

Fuente: (Sanchez., 2017)

Elaborado por: (Sanchez., 2017)

La metodología nos brinda los siguientes beneficios:

- **Cumplimiento de expectativas**

El cliente establece sus expectativas indicando su aporte a cada requisito, el equipo los estima y se prioriza.

- **Flexibilidad a cambios**
La metodología está diseñada para una alta capacidad de reacción a los cambios de requerimientos generados.
- **Reducción del tiempo para utilizar**
El cliente puede comenzar a utilizar el producto (funcionalidades más importantes), sin tener que esperar que se finalice.
- **Mayor calidad del software**
La metodología y la necesidad de generar una versión funcional que acepte el cliente afianza el desarrollo y promueve una mayor calidad en el producto de entrega.
- **Mayor productividad**
Se consigue a la eliminación de la documentación innecesaria de elaborar, además de mantener un equipo de trabajo motivado.
- **Maximiza el retorno de la inversión**
La producción de software únicamente con las presentaciones y la automatización que aporte al negocio.
- **Predicciones de tiempos**
Con el uso de una metodología se conoce el tiempo medio que se demora en desarrollar un producto
- **Reducción de riesgos**
La priorización de las necesidades del cliente y desarrollarlas en primer lugar y conociendo la velocidad de avance del proyecto se puede despejar riesgos que se presentan en el transcurso del proyecto.

La metodología Scrum ha sido utilizada por la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte para los proyectos CRM como portafolio

estudiantil y portafolio docente, cada una del proyecto se encuentran formados por varios módulos que se comunican entre ellos y funcionan de manera autónoma.

La metodología contribuye al proyecto de investigación porque brinda lineamientos básicos para la aplicación de metodologías ágiles, la implementación de un estándar es posible siempre que exista una metodología que sirva de apalancamiento optimizando el tiempo y la entrega de los productos de software solicitados por el cliente; los cuales siempre serán prioridad.

Para mayor detalla de esta metodología por favor revisar el Anexo 1 – Metodología Scrum

2.10 CALIDAD

En esta sección se mencionará algunas normas y criterios utilizados como referencia la evaluación de la calidad de desarrollo del software. Algunos de los modelos son: CMMI, ISO/IEC; para la evaluación de la calidad existen varios modelos que ayudan a la evaluación nosotros mencionaremos a los más conocidos como es CMMI que nos ayudan a gestionar la calidad de procesos y la ISO/IEC 25000 para la evaluación de la calidad de software

Una de las interrogantes que se tiene en cuanta es que un proceso que no actúa verifica la calidad de sus procesos de desarrollo de software tiene una alta probabilidad de construir un producto que carezca de calidad es por ello por lo que podemos decir que la calidad del proceso es directamente proporcional a la calidad del producto. (Calderón Macías, Martínez Munné, & Escofet, 2016)

2.10.1 MODELOS DE CALIDAD DE SOFTWARE

- **CMMI - CAPABILITY MATURITY MODEL INTEGRATION**

El CMMI es un enfoque de mejora de procesos, conjunto de normas de calidad. Estas se aplican a los diferentes procesos que hay que llevar a cabo para lograr producir software con calidad, es importante indicar que las normas ISO 90003, este modelo nos dice que hay que hacer y cómo hay que hacerlo (Cuéllar Chacón & Orduz Tarazona, 2015).

El marco CMMI podemos indicar que proporciona la estructura necesaria para crear los modelos la formación y los componentes de evaluación de CMMI. Para permitir el uso de múltiples modelos dentro del marco CMMI, los componentes de los modelos se clasifican como comunes a todos los modelos CMMI o aplicables a un modelo específico. El material común se denomina “CMMI Model Foundation” o “CMF.” (Institute, 2010).

Este modelo permite:

- ✓ Describir los componentes del modelo y sus relaciones.
- ✓ Comprender las áreas de proceso.
- ✓ Localizar información relevante en el modelo.
- ✓ Aplicar los conocimientos a su entorno de trabajo y en un equipo de evaluación de componentes y sus relaciones de un modelo.

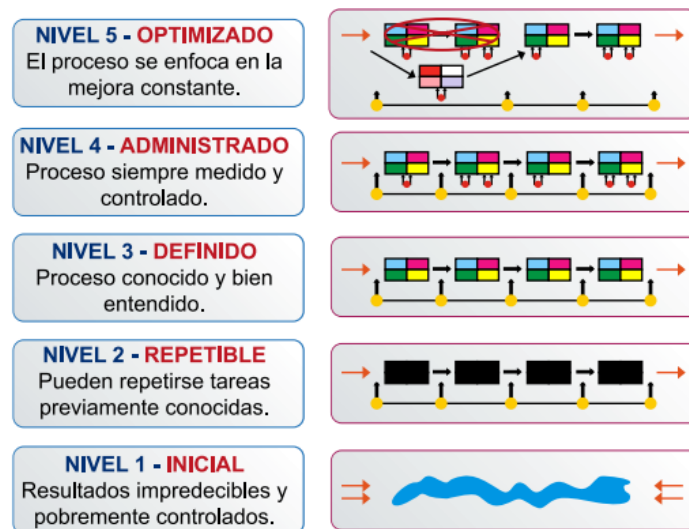


Figura 10 Proceso CMMI

Fuente: (Institute, 2010)

Elaborado por: (Institute, 2010)

El modelo CMMI cuenta con dos representaciones: escalonada y continua, las cuales no son iguales y cada organización está en la capacidad de seleccionar cuál de ellas adoptar dependiendo de la definición de los procesos que tenga. Si existe una equivalente que nos dice que un Nivel de Madurez equivale a tener en un conjunto de áreas de proceso determinado un determinado Nivel de Capacidad.

Modelo Escalonado: Basado en la madurez de la organización, es un modelo de software (CMM-SW) que establece 5 niveles de madurez para categorizar a una organización en función de qué áreas de procesos consiguen sus objetivos y se gestionan

con principios de ingeniería. La visión escalonada definirá a la organización dándole en su conjunto un nivel de madurez del 1 al 5 tal como se muestra en la siguiente figura (Larrondo Petrie , Medina García , & Méndez Giraldo, 2009)

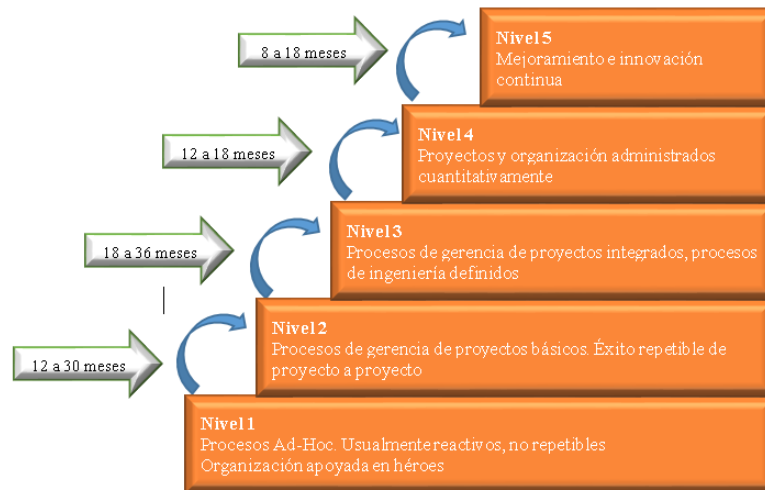


Figura 11: Modelo Escalonado

Fuente: (Larrondo Petrie , Medina García , & Méndez Giraldo, 2009)

Elaborado por: El Autor

Modelo Continuo: es un modelo para ingeniería de sistemas (SE-CMM) que establece 6 niveles posibles de capacidad para una de las 22 áreas de proceso implicadas en la ingeniería de sistemas. Al contrario del modelo escalonado, este modelo no agrupa los procesos en 5 modelos, al contrario, este modelo muestra la representación de niveles de capacidad de cada una de las áreas de proceso. Los 6 niveles definidos en CMMI para definir la capacidad de los procesos en el modelo continuo es:

- ✓ **0 - Incompleto:** El proceso no se realiza, o no se consiguen sus objetivos.
- ✓ **1 - Ejecutado:** El proceso se ejecuta y se logra su objetivo.
- ✓ **2 - Gestionado:** Además de ejecutarse, el proceso se planifica, se revisa y se evalúa para comprobar que cumple los requisitos.
- ✓ **3 - Definido:** Además de ser un proceso gestionado se ajusta a la política de procesos que existe en la organización, alineada con las directivas de la empresa.
- ✓ **4 - Cuantitativamente gestionado:** Además de ser un proceso definido, se controla utilizando técnicas cuantitativas.

- ✓ **5 - En Optimización:** Además de ser un proceso cuantitativamente gestionado, de forma sistemática se revisa y modifica o cambia para adaptarlo a los objetivos del negocio. Mejora continua.

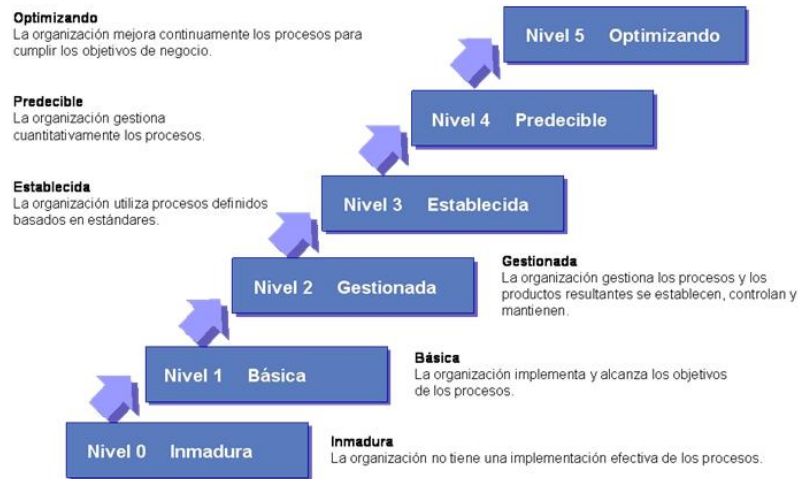


Figura 12: Modelo Continuo

Fuente: <http://www.sei.cmu.edu/cmmi/>.

Elaborado por: <http://www.sei.cmu.edu/cmmi/>.

2.10.2 ISO/IEC 25000

ISO/IEC 25000, conocida como SQuaRE (*System and Software Quality Requirements and Evaluation*), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones. (ISO25000, s.f.)



Figura 13: ISO 25000

Fuente: <http://iso25000.com/index.php/normas-iso-25000>

Elaborado por: <http://iso25000.com/index.php/normas-iso-25000>

La calidad puede ser interna es aquella que es medible cuando las características intrínsecas del propio producto es decir el código fuente, mientras que la calidad externa es medible a partir del comportamiento del producto de software (durante las pruebas); es decir en uso mediante la efectividad por parte de los usuarios.

Las normas que forman parte de los modelos, términos y definiciones comunes referenciados por todas las otras notas de la familia de las 25000, en el caso de este estudio se apoya en la norma ISO/IEC 25010 la cual indica el modelo de calidad de un producto de software se puede interpretar como el grado en que dicho software satisface los requisitos de sus usuarios aportando de manera fundamental un valor agregado. Son precisamente estos (funcionalidad, rendimiento, seguridad, mantenibilidad) los que se encuentran representados en un modelo de calidad, en la siguiente imagen puede revisar las 8 definiciones de calidad que nos brinda este modelo.



Figura 14: Modelo de calidad ISO/IEC 25010

Fuente: <http://iso25000.com/index.php/normas-iso-25000/iso-25010>

Elaborado por: <http://iso25000.com/index.php/normas-iso-25000/iso-25010>

De las 8 definiciones de calidad que presenta la norma ISO/IEC 25010 nos enfocaremos en la que indica la Adecuación de la Funcionalidad, esto debido a que este trabajo de investigación está enfocado a la gestión del proceso de desarrollo de software que se encuentra planteado para la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte y tomando en cuenta que la funcionalidad se encuentra formada por el proceso de levantamiento de requerimientos, su documentación, el diseño del modelo en base a la documentación levantada entre otros aspectos.

- **Adecuación Funcional**

La adecuación funcional representa la capacidad de un producto para tener funciones que satisfagan al cliente en un determinado tiempo y cumpliendo con las condiciones especificadas, las características pueden ser:

- **Complejidad funcional:** Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
- **Corrección Funcional:** Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
- **Pertinencia Funcional:** Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados

2.11 ESTÁNDAR ISO/IEC 29110

Es una norma ISO que lleva como denominación Ingeniería de Software - Perfiles de Ciclo de Vida en Pequeñas Entidades, esta norma es basada en las siguientes normas: mexicana NMX-I-059-NYCE-2011 (MoProSoft), en la ISO/IEC 12207, la ISO/IEC 15289, la ISO/IEC 15504 entre otras. La propuesta mexicana fue ofrecer sus procesos de la categoría operación como un perfil básico, la categoría de gerencia como perfil intermedio y la categoría alta Dirección como perfil Avanzado.

MoProSoft se define como un modelo de procesos para el desarrollo y mantenimiento de software dirigido a las pequeñas y medianas organizaciones, que tiene por objetivo principal incorporar las mejores prácticas en gestión e ingeniería de software.

La incorporación de este modelo en la industria permitirá elevar la capacidad de los productos que desarrollan o mantienen software que ejecutan y servicios de software con calidad y alcanzar niveles internacionales de competitividad. (Pairazamán Díaz & Escobar Nuñez, 2016)

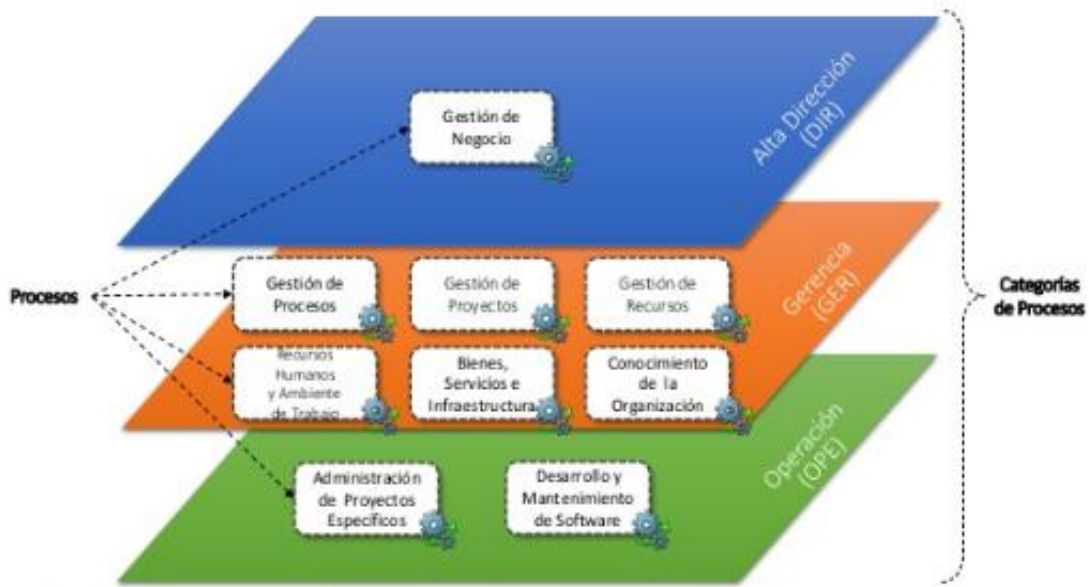


Figura 15: Estructura de MoProSoft

Fuente: (Torres Garibay & Armenta Fonseca, 2016)

Elaborado por: (Torres Garibay & Armenta Fonseca, 2016)

Actualmente, el estándar cuenta con empresas que ofrecen la certificación; NYCE ofrece a los usuarios los servicios acreditados de Evaluación y Certificación de Procesos de Desarrollo de Software basado en la ISO / IEC 29110 Ingeniería de Software – Perfiles del Ciclo de Vida en Pequeñas Entidades, con lo cual las empresas demuestran que desarrollan software de calidad, por medio de la mejora continua de sus procesos.

El estándar ISO / IEC 29110 es una norma que va dirigida para las pequeñas organizaciones que buscan estar posicionadas a sus productos de software en el mercado de manera fuerte y con atributos de calidad tanto en el producto como en el proceso con el que fue desarrollado y así contar con los mismos niveles de competitividad que las grandes empresas de software. La norma fue desarrollada para mejorar la calidad del producto y/o servicio de software, optimizando el desempeño de la organización, sin pretender excluir el uso de diferentes metodologías de Ciclo de Vida tales como: Cascada, Iterativo, Incremental, Evolutivo o Ágil. La ISO/IEC 29110 se divide en 5 partes de acuerdo al tipo

de audiencia a la que está dirigida, es decir, al campo de aplicación de cada una conformando el siguiente marco de trabajo (Framework) (NYCE, 2016):

ISO/IEC 29110	Título	Público objetivo
Parte 1	Descripción general	Microempresas, clientes, asesores, productores de normas, proveedores de herramientas y de metodología.
Parte 2	Marco y taxonomía	Productores de normas, proveedores de herramientas y de metodología. No está diseñado para microempresas.
Parte 3	Guía de evaluación	Asesores, clientes y microempresas
Parte 4	Especificaciones del perfil	Productores de normas, proveedores de herramientas y de metodología. No está diseñado para microempresas.
Parte 5	Gestión y guía de ingeniería	Microempresas y clientes

Figura 16: Público objetivo ISO/IEC 29110

Fuente: (Valdebenito Espinosa, Guía Metodológica para la implementación del estándar ISO/IEC 29110 en Pymes TICs de Colombia, 2016)c

Elaborado por: (Valdebenito Espinosa, Guía Metodológica para la implementación del estándar ISO/IEC 29110 en Pymes TICs de Colombia, 2016)c

El estándar cuenta con algunas características como son:

- a) Especifica el desarrollo y mantenimiento de software.
- b) Se encuentra dirigido para las pequeñas organizaciones que cuenta entre 1 a 25 personas laborando.
- c) Se encuentra orientado a proyecto internos como externos.
- d) Se puede certificar de conformidad con vigencia de 3 años y vigilancia anual.

Una descripción más a profundidad sobre el estándar la podemos encontrar en el Anexo 2: Contextualización del estándar ISO/IEC 29110

2.11 DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO DE LA UNIVERSIDAD TÉCNICA DEL NORTE

En la época de los años 70. Un importante sector de profesionales que sentían la necesidad de que el norte del país cuente con un Centro de Educación Superior que responda a los requerimientos propios del sector comienza a dar sus primeros pasos constituyéndose como Universidad Técnica del Norte (UTN,2017).

La institución cuenta con un diagrama orgánico que se indica a continuación:

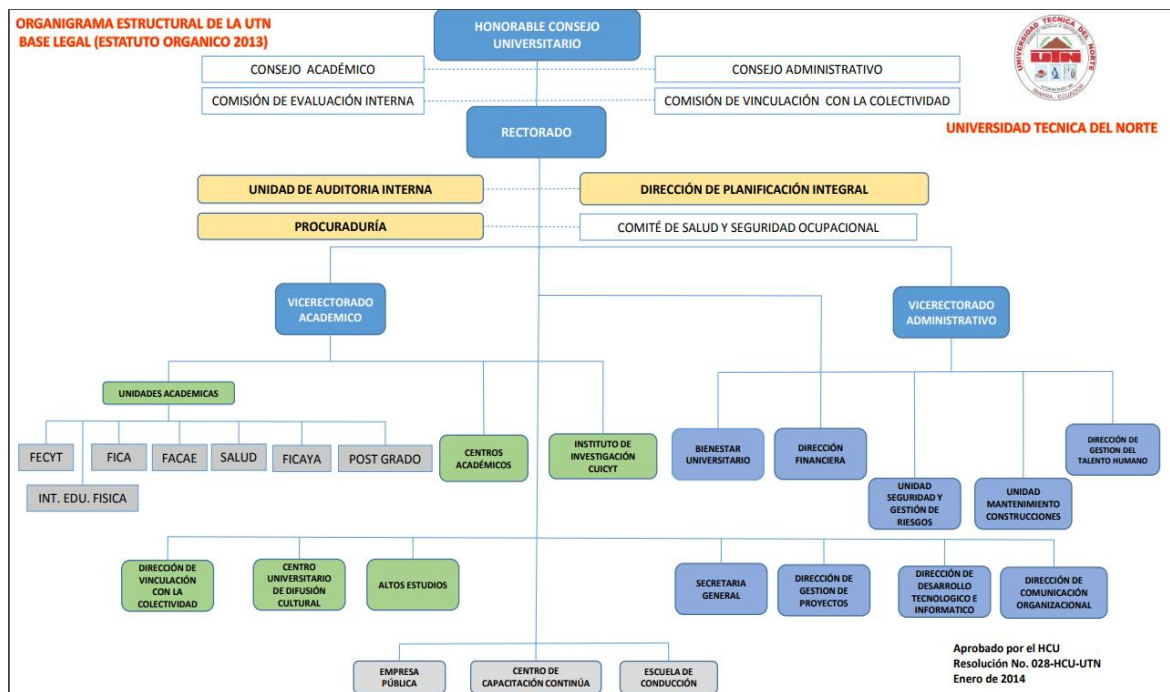


Figura 17: Orgánico Estructural UTN,

Fuente: A1_organigrama_estructural_utn.pdf

Elaborado por: Universidad Técnica del Norte

La universidad cuenta con una Dirección de Desarrollo Tecnológico e Informático que se ha encargado de la gestión, desarrollo y automatización de los procesos que tiene levantado la Universidad; permitiendo un salto tecnológico y brindando mas facilidades a estudiantes, docentes y personal administrativo. La ficha institucional es:

Ficha Institucional

DATOS GENERALES

- ID: 059
- Nombre: UNIVERSIDAD TÉCNICA DEL NORTE
- Sigla: UTN
- Tipo: Universidad
- Financiamiento: Estatal
- Subclase: Matriz
- RUC: 1060001070001
- Registro CONESUP: LEY No. 43 REGISTRO OFICIAL No. 482
- Fecha de Creación: 18/07/1986
- Teléfono: (593 6) 2997800 ext. 7070
- Fax: (593 6) 2997800 ext. 7001
- Casilla Postal: 100105
- Portal Web: www.utn.edu.ec
- Provincia: IMBABURA
- Cantón: IBARRA
- Ciudad: IBARRA
- Calle Principal: Av. 17 de JULIO
- Calle Secundaria: GRAL. JOSE MARIA CORDOVA
- Número de la Dirección: 5-21

Figura 18: Ficha Institucional

Fuente: http://www.utn.edu.ec/web/uniportal/?page_id=2008

Elaborado por: Universidad Técnica del Norte

2.11.1 ORGANIGRAMA DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

La Dirección de Desarrollo Tecnológico e Informático cuenta con el siguiente diagrama estructural



Figura 19: Dirección de Desarrollo Tecnológico e Informático,

Fuente: Resolución Consejo Universitario

Elaborado por: La Universidad Técnica del Norte

Según (Universidad Técnica del Norte, 2017), donde consta el Plan de Desarrollo de Informático, en la misión de la Dirección de Informática indica: corresponde administrar los servicios centrales de informática, computación y comunicaciones, y en especial apoyar a las facultades y otras unidades de la universidad, en temas del área.

También indica en sus apartados:

- Se tiene que proponer y desarrollar proyectos que involucren tecnologías computacionales y de información que aseguren la competitividad tecnológica local, nacional e internacional.
- Planificar, organizar, ejecutar y controlar la plataforma operativa

La universidad ha promovido el desarrollo de proyectos para la automatización de los procesos académicos y administrativos, los cuales son desarrollados para ser una herramienta de gestión institucional.

2.11.2 HERRAMIENTAS DE GESTIÓN

Son todos los sistemas, aplicaciones, controles, soluciones de cálculo, metodologías, procesos que ayuden a la gestión de la Universidad, algunos de los tipos de herramientas de gestión de software son:

- **Planificación de Recursos Empresariales**

(MUÑIZ, 2000) define como “Un sistema de planificación de los recursos y de gestión de la información que, de una forma estructurada, satisface la demanda de necesidades de la gestión empresarial”; centralizando la automatización de los procesos de una organización utilizando metodologías y herramientas tecnológicas que ayuden de manera rápida la automatización todo esto apoyado en las mejores prácticas e implementando o normas y estándares de calidad

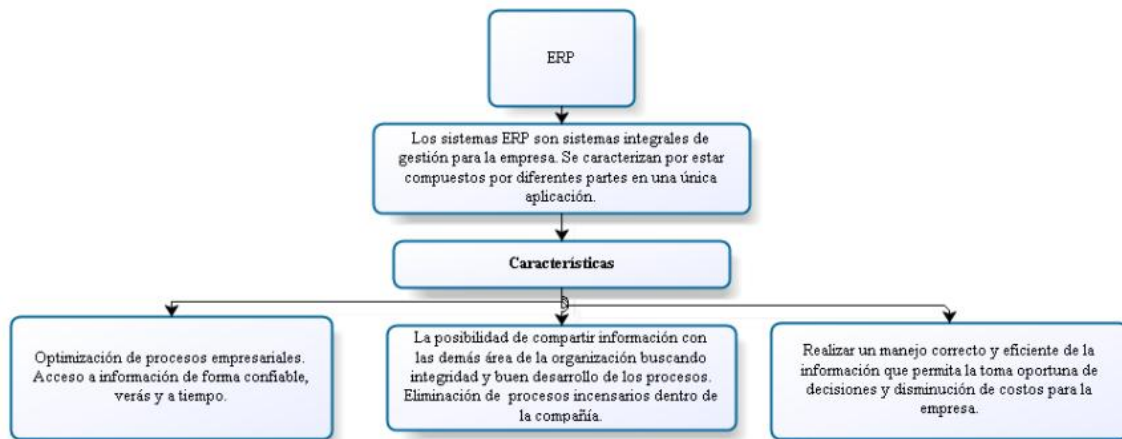


Figura 20: Resumen ERP,

Fuente: (Miranda Realpe, 2017)a

Elaborado por: (Miranda Realpe, 2017)a

- **Módulos del Sistema Integrado**

La universidad ha desarrollado proyectos de software integrados que aporten en la automatización de los procesos, a continuación, menciono algunos de los módulos

- Módulo de presupuesto
- Módulo de planificación
- Módulo de contabilidad
- Módulo de Compras
- Módulo de Tesorería

- **Gestión de Relaciones con los Clientes (CRM)**

Según (Renart, 2010), CRM asocia aplicaciones concretas o bases de datos para gestionar la información necesaria que permita desarrollar la relación por parte de la empresa con el cliente, además podemos indicar que CRM es una estrategia de negociación apoyada en las tecnologías de la información, diseñada para mejorar las interacciones humanas en el ambiente de negocios.

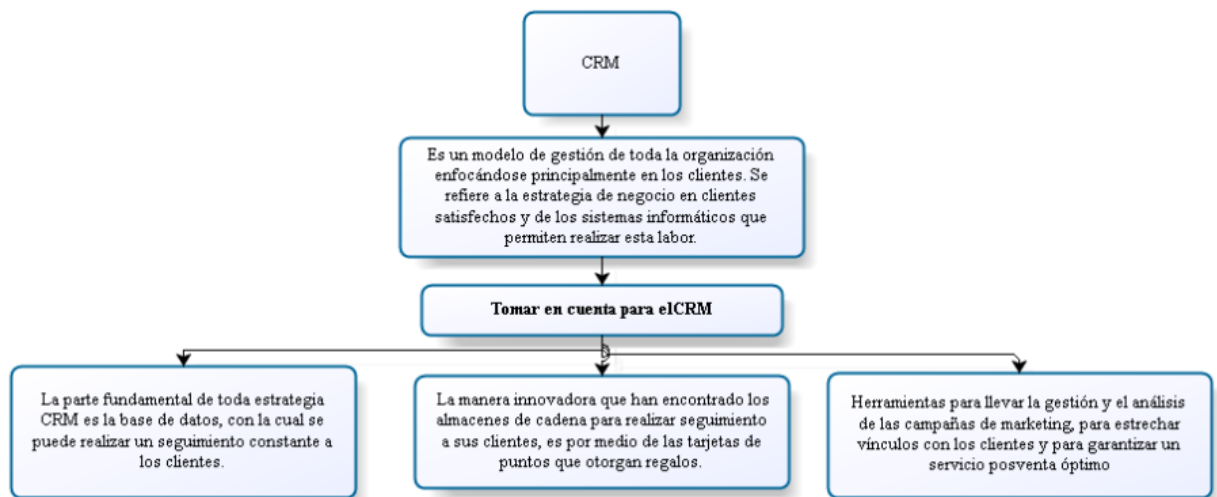


Figura 21 CRM,

Fuente: (Miranda Realpe, 2017) b

Elaborado por: (Miranda Realpe, 2017) b

2.2. MARCO LEGAL

Las NORMAS DE CONTROL INTERNO para las entidades, organismos del sector público y de las personas jurídicas de derecho privado que dispongan de recursos públicos, en su literal 410 habla sobre los procesos de Tecnologías de la Información, Las entidades y organismos del sector público deben estar acopladas en un marco de trabajo para procesos de tecnología de información que aseguren la transparencia y el control, así como el involucramiento de la alta dirección, por lo que las actividades y procesos de tecnología de información de la organización deben estar bajo la responsabilidad de una unidad que se encargue de regular y estandarizar los temas tecnológicos a nivel institucional.

Además, nos indica sobre el proceso de Desarrollo de software y/o adquisición de un software aplicativo donde me indica que la unidad de Tic´s es la encargada de regular los procesos de desarrollo adquisición de software con lineamientos, metodologías y procedimientos y más adelante en su literal 5 indica que el proceso de desarrollo debe ser documentado y de calidad, además de contar con normas y procedimientos nacionales e internacionales.

Para mayor información, podemos consultar la normativa de Control Interno en el Anexo 3 “Normas de Control Interno” <http://www.ministeriointerior.gob.ec/wp-content/uploads/downloads/2014/03/NORMAS-TECNICAS-DE-CONTROL-INTERNO.pdf>

CAPÍTULO III: MARCO METODOLÓGICO

En el presente capítulo se presenta información para el cumplimiento de los objetivos planteados en la investigación, se hace referencia primeramente a las modalidades de investigación.

3.1 LUGAR DE DESARROLLO DE LA INVESTIGACIÓN

La investigación se desarrollará en la Dirección de Desarrollo Tecnológico Informático de la Universidad Técnica del Norte, ubicado en la ciudad de Ibarra, provincia de Imbabura.

ID: 059

Nombre: UNIVERSIDAD TÉCNICA DEL NORTE

Sigla: UTN

Tipo: Universidad

Financiamiento: Estatal

Subclase: Matriz

RUC: 1060001070001

Registro CONESUP: LEY No. 43 REGISTRO OFICIAL
No. 482

Fecha de Creación: 18/07/1986

Teléfono: (593 6) 2997800 ext. 7070

Fax: (593 6) 2997800 ext. 7001

Casilla Postal: 100105

Portal Web: www.utn.edu.ec

Provincia: Imbabura

Cantón: Ibarra

Ciudad: Ibarra

Calle Principal: Av. 17 de Julio

Calle Secundaria: Gral. José María Córdova

Núm. Dirección: 5-21

Unidades de Observación:

- Dirección de Desarrollo Tecnológico e Informático.
- Personal del Área de Desarrollo de Software.

3.2 TIPO DE INVESTIGACIÓN

Para el análisis del trabajo de investigación se trabajó con los tipos de investigación cualitativa, cuantitativa, documental, exploratoria y descriptiva

3.2.1 INVESTIGACIÓN CUALITATIVA

El trabajo de investigación se realizó por medio de entrevistas, observación de la gestión del proceso de desarrollo de software en la Dirección de Desarrollo Tecnológico e Informático, ubicado en la ciudad de Ibarra; analizando las actividades problemáticas tal como indica (Hernández, Fernández, & Baptista, 2006) expone “La investigación cualitativa se fundamenta más en un proceso inductivo (explorar y describir, y luego generar perspectivas teóricas). Van de lo particular a lo general”.

3.2.2. INVESTIGACIÓN DE CAMPO

(Arias, 2006) define:

“La investigación de campo es aquella que consiste en la recolección de todos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variables algunas, es decir, el investigador obtiene la información, pero no altera las condiciones existentes. (p.31)”

El presente trabajo de investigación se aplicó para la evaluación de la situación inicial del proceso de gestión de Desarrollo de Software, como por ejemplo para conocer que el proceso no se encuentra formalizado para el desarrollo de los proyectos.

3.2.3 INVESTIGACIÓN DOCUMENTAL

(Morales, 2016) manifiesta que:

“En el proceso de investigación documental se dispone, esencialmente, de documentos, que son el resultado de otras investigaciones, de reflexiones de teóricos, lo cual representa la base teórica del área objeto de investigación, el conocimiento se construye a partir de su lectura, análisis, reflexión e interpretación de dichos documentos.” (p 2)

Se ha revisado documentación generada de otros procesos de investigación sobre la implementación del estándar, este es el caso del proyecto de instalación que se realizó en Perú; este tipo de investigación se lo realizo en revistas científicas, libros, tesis, etc.

3.3 DISEÑO DE LA INVESTIGACIÓN

La investigación no experimental se realiza sistemática y empíricamente, las variables independientes no se manipulan porque ya ha sucedido. La inferencia sobre las variables se realiza sin intervención o influencia directa y dicha relación se observan tal y como se ha dado en el contexto natural.

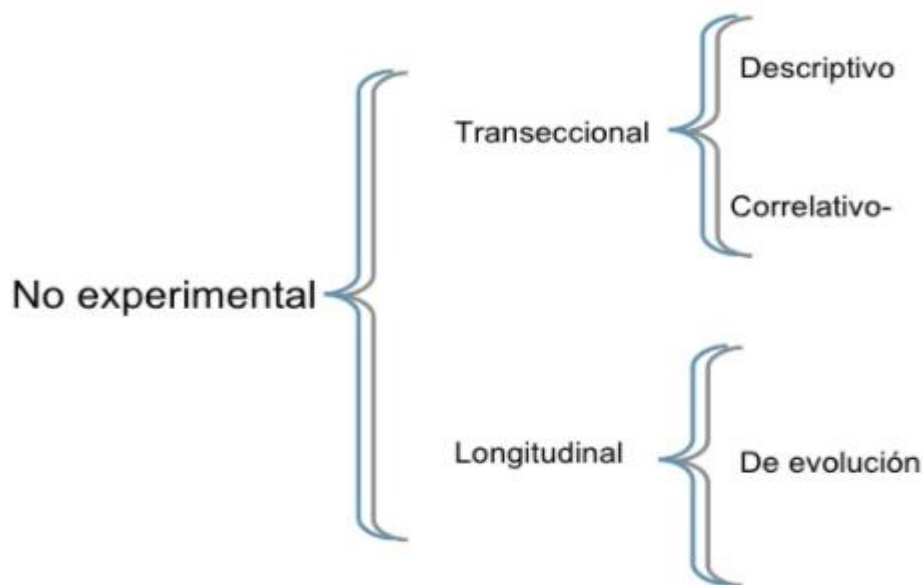


Figura 22Tipos de Diseño no experimental

Fuente: <https://es.slideshare.net/uabcpsique/diseos-no-experimentales-transversales-transversales-descriptivos-y-exploratorios>

Elaborado por: <https://es.slideshare.net/uabcpsique/diseos-no-experimentales-transversales-transversales-descriptivos-y-exploratorios>

El presente trabajo de investigación utilizó un diseño no experimental – Transeccional porque se practica una investigación de manera empírica observando las variables de manera directa evitando la manipulación de la información por ejemplo se observó el proceso de solicitud de un proyecto de software desde la fase inicial donde el cliente realiza el pedido, pasando por las fases de levantamiento de requerimientos, desarrollo hasta llegar la implementación del producto final

3.4 POBLACIÓN Y MUESTRA

La investigación se realizará en la Dirección de Desarrollo Tecnológico Informático de la Universidad Técnica del Norte, se tomará en cuenta las siguientes personas:

Población	Frecuencia	Porcentaje
Director de Tecnología	1	20
Personal de Desarrollo	4	80%
Total	5	100%

Tabla 6: Población y Muestra

Fuente: Universidad Técnica del Norte

Elaborada por: El Autor

La población donde se desarrollará el trabajo de investigación no supera 100 personas, se tomará como muestra al total de la población para la aplicación de encuestas que ayuden a recabar información.

3.4 TIPO DE MÉTODO

Para el presente trabajo de investigación se utilizó el **Método Hipotético – Deductivo** utilizado en la comunidad científica como en la vida cotidiana, se trata de encontrar la solución a la problemática empleando un razonamiento lógico; consiste en emitir hipótesis o proposiciones que me ayuden a encontrar la solución. (Cerrano, 2012)

El método Deductivo según (Cerrano Sánchez, 2012)

“El **Método Deductivo** lo empleamos corrientemente tanto en la vida ordinaria como en la investigación científica. En el camino lógico para buscar la solución a los problemas que nos planteamos”

Este método nos ayuda a poder deducir la situación actual de la institución y tener ideas claras de cómo funcionan los procesos lógicamente y donde implementar la evaluación de control interno para obtener resultados óptimos.

3.5 PROCEDIMIENTO

Para el presente trabajo de investigación se desarrolló el siguiente proceso para el cumplimiento de los objetivos planteados:

3.5.1 ANÁLISIS DEL ESTÁNDAR ISO/IEC 29110

Se desarrolló un tipo de investigación Documental, utilizando el método inductivo, para conocer las definiciones, características y alternativas de implementación, donde se encontró que el estándar se encuentra implementado en Colombia, Perú, Argentina, Canadá; donde ya se ha tenido resultados.

Además, se determinó que es demasiado complicado tratar de implementarlo de manera directa, es por ello que se han desarrollado nueve paquetes de Puesta en Operación, donde cada uno de estos paquetes se encuentran relacionados de manera directa con las cada una de las fases de Desarrollo de un proyecto de software.

3.5.2. ANÁLISIS DE SITUACIÓN ACTUAL DEL PROCESO DE DESARROLLO DE SOFTWARE

Para el cumplimiento de este objetivo se utilizó el tipo de investigación De Campo, ya que se realizó una observación del fenómeno directamente en el ambiente donde se desarrolló, donde se pudo determinar que existe una falda de formalización en la gestión del proceso de desarrollo de software.

Al ser la metodología RUP la guía para los proyectos ERP que se desarrolló en los proyectos de software desarrollados en la Dirección de Desarrollo Tecnológico e

Informático, se estableció una matriz de actividades y tareas a ser desarrolladas por la metodología obteniendo indicadores de cumplimiento, de igual manera se creó la relación de utilizar una metodología águila como es SCRUM.

3.5.3. IMPLEMENTACIÓN DE ESTÁNDAR ISO/IEC 29110

Para la implementación del estándar en un proyecto real se desarrollaron las siguientes actividades:

- a) Conocer las especificaciones del estándar
- b) Conocer y obtener cada uno de los paquetes de puesta en operación traducidos al idioma español.
- c) Establecer la relación entre los paquetes de despliegue y la metodología SCRUM.
- d) Definición de metodología alternativa que soporte la implementación del estándar y la metodología SCRUM.
- e) Elaboración de plantillas o artefactos que faciliten seguir los lineamientos de la Metodología SCRUM, como también del estándar ISO/IEC 29110.
- f) Revisión de plantillas con personal de la Dirección de Desarrollo Tecnológico e Informático.
- g) Llenado de plantillas en un proyecto real.

CAPÍTULO IV: RESULTADOS Y DISCUSIÓN

4.1. CONTEXTUALIZACIÓN DEL ESTÁNDAR ISO/IEC 29110

La industria de software demuestra su potencial con tasas de crecimiento que ascienden al 17 % por año, pese a que uno de sus pendientes es crear mayores capacidades para poder internacionalizarse. En 2015 este sector llegó a generar \$ 45 millones en exportaciones, apenas un 0,2 % de todo lo que el país vendió al mundo en ese año. (Expreso, 207).

Las PYMES se clasifican tomando en cuenta tres condiciones que son el número de personas que laboran, el valor bruto en ventas anuales y su total de activos. Una microempresa tiene como máximo 9 empleados y hasta \$100.000 dólares en ventas anuales, así también en activos; una pequeña empresa tiene entre 10 a 49 empleados y tiene registrado una venta anual de \$1'000.000 de dólares.

De acuerdo con la Superintendencias de Compañías en el 2011 indico que existían 633 de las cuales 277 se presentan como microempresas y solo 7 se consideran grandes empresas, según el criterio de expertos proyectan que al menos el 90% de las empresas se de software se repostan como pequeñas y medianas empresas desarrolladoras de software.

4.1.1 DIAGNÓSTICO DEL USO DE ESTÁNDARES Y PROCESOS EN LAS PYMES DE LA INDUSTRIA DE SOFTWARE ECUATORIANA

Un estándar establece un marco de referencia común al que debe hacer referencia la industria para un proceso, actividad y/o entregables del ciclo de vida del software. Se puede indicar que un proceso de software es un conjunto de actividades de cierta menara técnicas como administrativas que son requeridas para la construcción de un producto de software.

En la siguiente tabla podemos mencionar la variedad de estándares que existen para cada uno de los procesos del ciclo de vida del desarrollo de software

CATEGORIA	ALTERNATIVAS
Ciclo de vida del Software	ISO/IEEE 12207:2008
Calidad y Evaluación de Producto	ISO9126 ISO25000 ISO14598
Calidad y Evaluación de Proceso	ISO9001:2008 ISO9000-3 EFQM
Madurez del Proceso	ISO15504:2003 CMMI IT-Mark COMPETISOFT
Ingeniería de Software	IEEE SESC
Procesos de Software	CASCADA MSF RUP XP TSP Light MECPDS MoPRoSoft METRICA v3
Gestión de Proyecto	SCRUM PMBOK

Tabla 7: Estándares y Procesos de Software

Fuente: (Sánchez, 2012)

Elaborada por: El Autor

En el 2003, se realizó un estudio con una muestra de 77 empresas desarrolladoras de software de Quito, Guayaquil y Cuenca. Cuando se les preguntó a las empresas sobre la familiaridad con algunos estándares de calidad, el 94,8% de las empresas dijeron conocer acerca de ISO 9001, el 48% acerca de MSF y el 29,8% acerca de CMM. El 36,3% de las empresas encuestadas utilizan estándares de calidad en el desarrollo de software, de los

cuales, sólo el 24,6% corresponde a estándares internacionalmente reconocidos. Un resultado alentador es que, al momento del estudio, el 37,6% de las empresas estaba analizando la posibilidad de implantar algún estándar. (Sánchez, 2012)

En otro estudio realizado en el 2007, con empresas representativas de Quito y Guayaquil, se determinó que el 41,5% utilizó en sus proyectos el proceso Cascada, el 22% Espiral, el 17.1% MSF, el 7.3% RUP. En cuanto a la duración de los proyectos, el 43% estaba entre 3 y 9 meses, el 30% menos de 3 meses, el 16% más de 12 meses, y el 11% entre 9 y 12 meses. En su mayoría los proyectos duraron y costaron más de lo estimado. La complejidad técnica es media, pero la situación del negocio es difícil. (Sánchez, 2012)b

En el 2017 se practicó una encuesta a empresas desarrolladoras de software para conocer de la provincia de Imbabura sobre el conocimiento y practica del estándar ISO/IEC 29110 para la calidad de la Gestión del Proceso de Desarrollo de Software, y se determinó que las empresas conocen a nivel básico del estándar ISO/IEC 29110, pero que no es aplicado, se utilizan otras alternativas como CMMI, Metodologías de desarrollo de software como RUP, XP y el modelo en cascada, ISO/IEEE 12207.

La contextualización del estándar se detalla más en el Anexo 2 “Contextualización del estándar ISO / IEC 29110”

4.2. PROCESO DE DESARROLLO DE SOFTWARE PARA LOS PROYECTOS CRM DE LA DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

La Dirección de Desarrollo Tecnológico e Informático cuenta con un proceso de Desarrollo de Software basado en la metodología RUP que utiliza procesos de las fases de desarrollo estándar como son las fases de análisis, diseño, construcción, pruebas y despliegue definidas para un proyecto de software; no obstante la Dirección a evolucionado en la aplicación de metodologías de desarrollo de software y es así que existen proyectos que se ha desarrollado con la metodología SCRUM, uno por temas de Tesis de Grado (pregrado) y otros proyectos desarrollados por el equipo de desarrollo de la Dirección, a continuación se detallan los proyectos ejecutados con esta metodología:

- Portafolio docente
- Portafolio estudiantil

- Bolsa de empleo
- Seguimiento a graduados
- Portafolio de autoridades
- Portafolio de carreras
- Portafolio de directores
- Matriculación en línea
- UTN en Cifras

Las aplicaciones mencionadas no siguieron la metodología RUP con la que se desarrolló los módulos del ERP, pero siguieron la metodología SCRUM por ser aplicaciones que requerían salgan a producción en tiempos cortos y tenían requisitos cambiantes por lo que se concentró en la entrega del producto y la satisfacción del cliente.

4.2.1 PROPUESTA DE PROCESO DE DESARROLLO DE SOFTWARE

Para el desarrollo de aplicaciones CRM con la metodología SCRUM se propone el siguiente proceso a ser ejecutado.

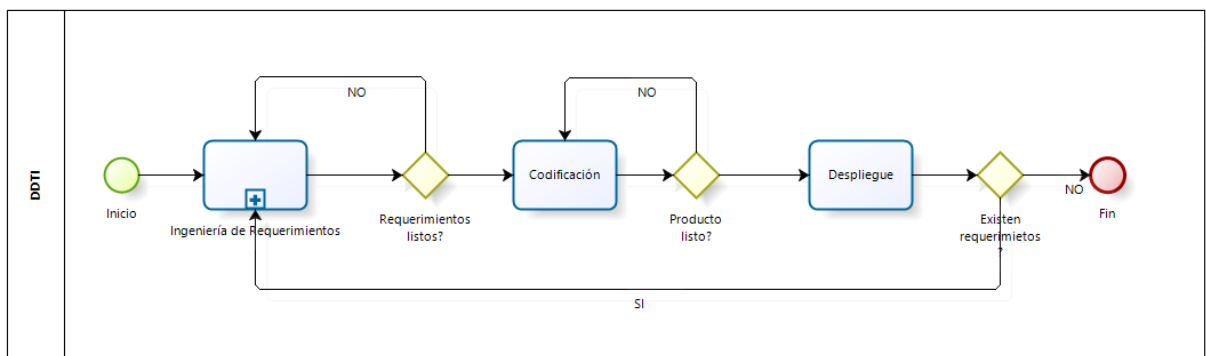


Figura 23: Proceso de Desarrollo de Software

Fuente: Universidad Técnica del Norte

Elaborada por: El Autor

Donde, en el proceso de detectan tres macroprocesos que son:

- **Ingeniería de Requerimientos:** El proceso de ingeniería de requerimientos es aquel que tiene como objetivo realizar el levantamiento, modelado de los requerimientos para ser ejecutados en el Sprint según scrum, este macroproceso está conformado por subprocesos que se detallan en la siguiente figura:

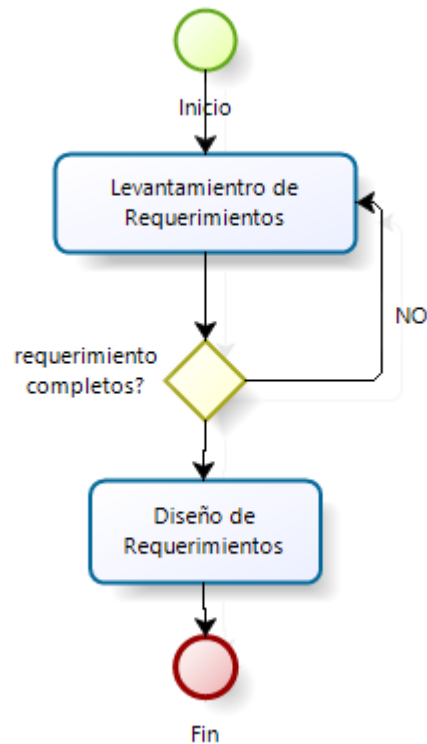


Figura 24: Subproceso de Ingeniería de Requerimientos

Fuente: Universidad Técnica del Norte

Elaborada por: El Autor

- **Codificación:** Es el proceso donde se desarrolla tanto el back-end como el front-end de la aplicación, además como es un desarrollo ágil, es basado en las pruebas el desarrollo
- **Despliegue:** El despliegue es poner el desarrollo ya revisado y probado en producción para que el cliente pueda utilizarlo de manera inmediata.

4.2.2 ANÁLISIS DEL PROCESO DE DESARROLLO DIRECCIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN E INNOVACIÓN

Se realizó el análisis profundo al Proceso de Desarrollo de Software que maneja Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte se tomó en cuenta que:

- a) Los proyectos que contemplan el ERP fueron desarrollados utilizando la metodología RUP.
- b) La metodología SCRUM, está siendo recientemente utilizada para realizar proyectos CRM en el área de desarrollo de software, pero para la documentación la Dirección se apoyan en la Metodología RUP.
- c) Los proyectos CRM fueron desarrollados tomando como referencia RUP y SCRUM para lo referente a Fases de Desarrollo, documentación generada.

Para realizar el análisis de las fases del proceso de desarrollo se toma como referencia las fases de la metodología RUP, ya que la Dirección de Desarrollo Tecnológico e Informativo conoce estas fases y de cierta manera ha sido aplicada a los proyectos CRM, es por ello por lo que en la tabla 3 se realizó una similitud de las fases que a continuación se presenta:

Proceso de desarrollo DDTI	Ciclo de vida del RUP	SCRUM
Análisis de requerimientos	Fase de Inicio	Sprint
Diseño de Software	Fase de Elaboración	
Construcción o Codificación	Fase de Construcción	
Pruebas	Fase de Transición	
Despliegue	Fase de Transición	

Tabla 8: Similitudes entre proceso DDTI y RUP

Fuente: Universidad Técnica del Norte

Elaborado por: El Autor

4.2.2.1 ANÁLISIS DE ROLES

Los roles que la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte definen son la siguiente:

- Jefe de Proyecto
- Desarrollador

ROL	DESCRIPCION
Jefe de Proyecto	El jefe de proyecto gestiona los recursos necesarios para la ejecución del proyecto, coordina, organiza y planifica las actividades a realizarse a los integrantes del equipo del proyecto. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos.
Desarrollador	Recolección de los requisitos y especificación de requerimientos del sistema. Construcción de prototipos. Elaboración del Modelo de Análisis y Diseño. Diseño de la base de datos. Colaboración en la elaboración de las pruebas funcionales. Elaborar modelos de implementación y despliegue

Tabla 9: Roles definidos en La DDTI

Fuente: (CHAVÉZ GUAMIALAMA, 2015)

Elaborado por: El Autor

4.2.2.2 RESUMEN DE ANÁLISIS DE METODOLOGÍA RUP

Para practicar este análisis se establecieron todas las actividades que se desarrolla en RUP y se contrasto con todas las actividades que desarrolla la Dirección de Desarrollo Tecnológico e Informático a fin de determinar el porcentaje de cumplimiento por etapas del proceso de desarrollo de software que propone la metodología

A continuación, se muestra un resumen del análisis realizado

	Total	Actividades realizadas	%	Actividades no realizadas	%
Fase de Inicio	83	28	33.73%	55	66.27%
Fase de Elaboración	136	39	28.68%	97	71.32%
Fase de construcción	106	29	27.36%	77	72.64%
Fase de Transición	125	34	27.20%	91	72.80%
Total	450	130	28.89%	320	71.11%

Tabla 10 Consolidado análisis de situación actual

Fuente: Análisis y tabulación realizada

Elaborado por: El autor



Figura 25: Consolidado análisis de situación actual

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

Con la finalidad de profundizar en el conocimiento y el grado de aplicación de un proceso de desarrollo ese realizó el análisis de los roles que la Dirección de Desarrollo Tecnológico e Informático utiliza para los proyectos.

El detalle del análisis se encuentra en el Anexo 5. “Análisis de Metodologías RUP Y SCRUM utilizada en la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte”

4.2.2.3 RESUMEN DE ANÁLISIS DE METODOLOGÍA SCRUM

La Metodología SCRUM aplicada en la Dirección de Desarrollo Tecnológico e Informático apoya a la gestión del proyecto de software, las interacciones que realiza el desarrollador en la construcción del producto son las mismas que utiliza RUP, con la diferencia de la magnitud del producto.

El análisis se realizó de la metodología SCRUM define 6 actividades a ser desarrolladas en el proceso de desarrollo de Software y estas son:

- **Planificación la Iteración**

Se realiza la planificación de las iteraciones y esta se divide en dos partes que es la planificación del cliente priorizando los requerimientos y la segunda parte

que es la planificación del equipo de desarrollo para las actividades a desarrollar en cada sprint.

- **Ejecución de la Iteración**

Se ejecuta en bloques cortos, cada ejecución de un bloque debe dar como resultado un entregable potencial.

- **Reunión diaria de sincronización del equipo**

El objetivo de esta reunión es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad, al poner de manifiesto puntos en que se pueden ayudar unos a otros (ÁGILES, 2017)

- **Demostración de los requerimientos completados**

Se realizan reuniones informales con el cliente para hacer una demostración del avance de los requerimientos.

- **Retrospectiva**

De cada reunión se obtiene un feedback para llevar a cabo las mejoras continuas en el proyecto y en los futuros proyecto.

- **Refinamiento de lista de requerimientos y cambios de proyecto**

Se realizan reuniones con el cliente para realizar el refinamiento de los requerimientos e ir trabajando con los objetivos.

Para realizar el análisis de la metodología scrum se elaboró un cuadro donde se tiene las actividades realizadas y las tareas a ejecutar, las cuales se detallan en la siguiente tabla:

Actividad	Tarea	Si	No
Planificación de Iteración	El equipo conoce todos los requerimientos	X	
	El equipo interviene en la planificación de cada iteración	X	
	El equipo realiza la priorización de las actividades	X	
Ejecución de Iteración	Las actividades son ejecutadas de manera independiente	X	
	Las actividades se ejecutan en bloques cortos	X	
	Codificación de Requerimientos	X	
Reuniones diarias	Se realizan reuniones diarias con el equipo	X	

	Las reuniones tienen una duración de entre 15 a 20 min	X	
	El equipo comparte novedades del desarrollo	X	
	Se plantean posibles soluciones	X	
	Se ejecuta un tablero de cumplimiento	X	
Retrospectiva	Se realiza una retroalimentación del proyecto	X	

Tabla 11Tabla de análisis detallada

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

Tareas ejecutadas	12
Tareas no ejecutadas	0
Total	12

Tabla 12: Tabla Resumen del análisis

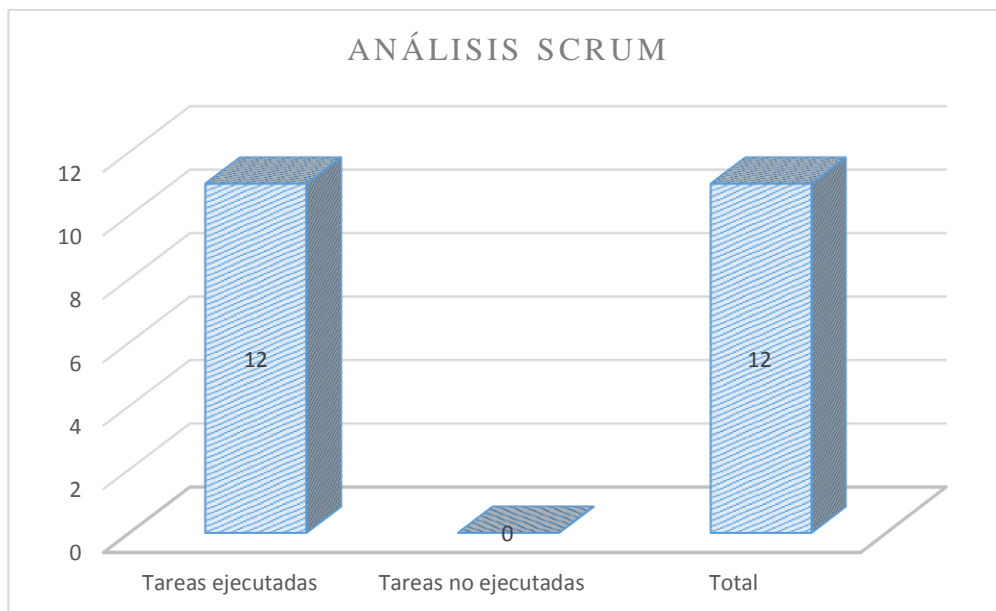


Figura 26: Análisis Scrum

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

Del análisis realizado se determinó que la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte ejecuta todas las tareas que se plantea en el estudio para aplicar la metodología SCRUM

El detalle del análisis se encuentra en el Anexo 5. “Análisis de Metodologías RUP Y SCRUM utilizada en la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte”

4.3 IMPLEMENTACIÓN DEL ESTÁNDAR ISO/IEC 29110

Para la implementación del estándar en el proceso de desarrollo de software que tiene definido la Dirección de Desarrollo Tecnológico e Informático se realizó un análisis para saber si el estándar se ajustaba de manera idónea sin tener que realizar ningún tipo de ajuste al procedimiento. Para ello se elaboraron las siguientes actividades:

4.3.1 ANÁLISIS DE COMPATIBILIDAD ENTRE METODOLOGÍA SCRUM – ESTÁNDAR ISO/IEC 29110

Para el análisis de compatibilidad entre la metodología scrum y el estándar ISO/IEC 29110 se tomaron se consideró características del estándar y de la metodología scrum; estos criterios son roles definidos, documentación generada, ciclo de vida del proceso de desarrollo.

El ciclo de vida de un proyecto de software definido por el estándar ISO/IEC29110 indica que consta de dos grandes procesos: Gestión de Proyectos e Ingeniería de Software; cada uno de estos procesos cuenta con sus actividades definidas y artefactos específicos a ser generados; en cambio la metodología scrum se encuentra enfocada a la gestión de proyectos y su principal prioridad es tener entregables listos para el cliente final mediante la ejecución de iteraciones dejando con baja prioridad la generación de documentación y formalización de proyectos.

A continuación, se muestra una matriz comparativa entre la metodología Scrum y el estándar ISO/IEC 29110 donde se evidencia las diferencias de compatibilidad encontradas:

Estándar ISO/IEC 29110	Metodología SCRUM	CONCLUSION
Genera documentación por cada actividad que se ejecuta en los dos macroprocesos del desarrollo de software como es Gestión de Proyectos e Ingeniería de Software	La Metodología Scrum considera realizar entregables funcionales a los clientes y generar el mínimo de documentación	Entre el estándar y la metodología tienen criterios distintos el estándar fomenta la formalidad del proyecto y generación de documentos mientras que la metodología se centra en generar entregables funcionales para que el cliente pueda utilizarlos
El estándar cuenta con 4 roles definidos: cliente, líder técnico, programadores, analista	Scrum cuenta con 3 roles: Product Owner, Scrum Master, Scrum Team	Cuentan con menor número de roles, pero las actividades de un rol pueden estar integrados en otro

Tabla 13: Diferencias entre estándar ISO/IEC 29110 y Metodología SCRUM

Fuente: La investigación

Elaborado por: El autor

En base a la investigación realizada se pudo concluir que el estándar ISO/IEC 29110 no es compatible de manera transparente con la metodología de desarrollo SCRUM; debido a la diferencia de enfoques que se tiene; estas diferencias afectan de manera específica en desarrollo de documentación, entrega de productos, integrantes en el equipo de desarrollo.

Por lo tanto, la única manera de implementar el estándar ISO/IEC 29110 y lograr una integración con la metodología scrum de manera transparente para el equipo de desarrollo y el cliente final es desarrollar un Método Híbrido entre el estándar ISO/IEC 29110 y Scrum.

4.3.2 ANÁLISIS DE MÉTODO HÍBRIDO ISO/IEC 29110 Y SCRUM

Para el desarrollo del método híbrido se consideró los siguientes factores: Roles, documentación y actividades

4.3.2.1 ROLES

Se revisaron las competencias de los roles entre el estándar ISO/IEC 29110 y Scrum, se observa una relación entre cliente, líder de proyecto y equipo; los demás roles se

encuentran contemplados en el equipo Scrum. Scrum no existe diferencia, aun cuando scrum no define funciones específicas para los miembros del equipo. A continuación, se muestra una tabla donde se indica la relación y los roles que pueden ser asumidos por los miembros del equipo para suplir los roles del estándar.

ISO 29110 \ Scrum	Product Owner	Scrum Master	Scrum Team
Cliente	X		
Líder de Proyecto		X	
Equipo			X
Analista			*
Desarrollador			*
Programador			*

Tabla 14 Comparación de roles

Fuente: (Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110, 2013)

Elaborado por: El autor

4.3.2.2 DOCUMENTACIÓN

Se estableció un análisis de comparativo entre los documentos que se generan en el estándar ISO/IEC 29110 y la metodología scrum dando como resultado que los documentos de scrum como el Product Backlog, sprint Backlog deberán ser ajustados para cumplir con las definiciones del estándar. A continuación, se muestra en la tabla el análisis elaborado e investigado.

Documentos ISO 29110	Documentación SCRUM	Análisis
Declaración de Trabajo	Product Backlog	No tiene una estructura definida por lo que se puede realizar un acercamiento al producto
Configuración de Software	-	Scrum no presenta un documento establecido formalmente
Solicitud de Cambio	Spring Backlog	No tiene una estructura definida por lo que se puede realizar un acercamiento al producto
Plan de Proyecto	Product Backlog	No tiene una estructura definida por lo que se puede realizar un acercamiento al producto
Registro de Aceptación	-	Scrum no presenta un documento establecido formalmente
Mínutas de Reuniones	-	Scrum no presenta un documento establecido formalmente

Tabla 15: Comparativa de documentos

Fuente: (Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110, 2013)

Elaborado por: El autor

4.3.2.3 ACTIVIDADES

Como se menciona el estándar ISO/IEC 29110 tiene dos procesos Gestión de Proyecto e Ingeniería de Software que abarcan el ciclo de vida del desarrollo de un proyecto. Las actividades realizadas en el proceso de Gestión de Proyecto equivalen a la recepción del Product Owner con la lista de requerimientos que se utiliza para crear el Product Backlog. Las actividades relacionadas con le ejecución del plan de proyecto y evaluación y control se relacionan con los sprint y el cierre del proyecto con la entrega final del proyecto de software. El proceso de Ingeniería de Software se realiza en cada sprint.

Actividades AP	Actividad IS
Planificación del Proyecto	Iniciación de la Implementación
<ul style="list-style-type: none"> - Revisar la Declaración de trabajo - Establecer tareas a realizar con dependencia y duración - Establecer puntos de V&V - Definir equipo de trabajo con roles y responsabilidades - Definir capacitaciones - Estimar esfuerzo, costo y calendario 	<ul style="list-style-type: none"> - Revisar el Plan de Proyecto con el equipo de trabajo y establecer tareas a realizar - Establecer el compromiso del equipo y el Lider - Establecer el ambiente de Implementación
Ejecución del Plan de Proyecto	Análisis de Requerimientos de Soft
<ul style="list-style-type: none"> - Registrar el progreso del proyecto - Analizar y evaluar los cambios y su impacto. - Aprobar los cambios en el Plan. - Mantener reuniones con el equipo de trabajo y el cliente. - Actualizar el Repositorio 	<ul style="list-style-type: none"> - Revisar tareas asignadas - Elicitar, analizar y especificar requerimientos - V&V los requerimientos - Control de versiones
Evaluación y Control del Proyecto	Arquitectura y Diseño Detallado del software
<ul style="list-style-type: none"> - Evaluar el progreso del Plan - Identificar y evaluar desviaciones y problemas de costo, calendario, técnicos. - Documentar cambios y acciones correctivas. - Actualizar el Repositorio 	<ul style="list-style-type: none"> - Diseñar arquitectura. Componentes - Rever especificación de requerimientos - Verificar Diseño y casos de prueba - Control de versiones
Cierre del Proyecto	Construcción
<ul style="list-style-type: none"> - Realizar la entrega del producto según lo acordado. - Realizar soporte al cliente - Finalizar el proyecto y firmar aceptación. 	<ul style="list-style-type: none"> - Rever el diseño para determinar secuencia de construcción. - Codificar. - Trazabilidad.
	Prueba e integración
	<ul style="list-style-type: none"> - Integrar componentes - Realizar pruebas y documentar - Verificar líneas base
	Entrega
	<ul style="list-style-type: none"> - Controlar Documentación - Entrega del producto

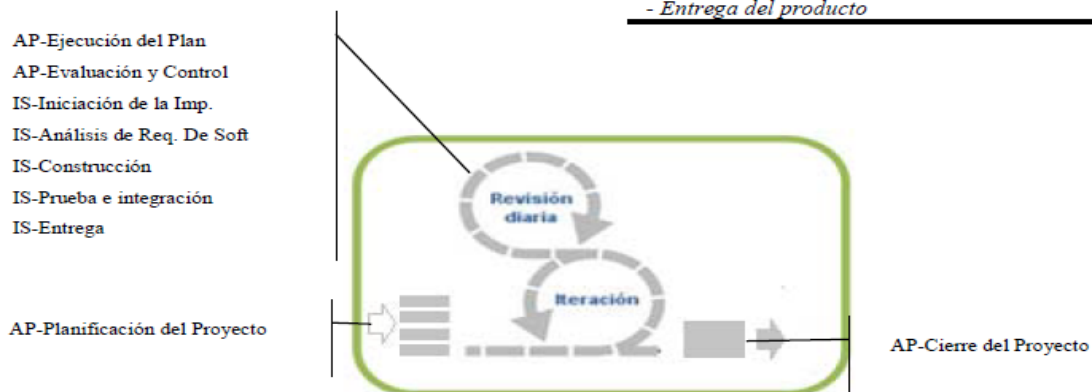


Figura 27: Comparación de Actividades

Fuente: (Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110, 2013)

Elaborado por: El autor

4.3.3 MODELO UTN-SCRUM

La elaboración de una metodología híbrida que permita la integración de la metodología scrum con el estándar ISO/IEC 29110; integración que sea de bajo impacto para el equipo desarrollador y el cliente. Que al momento de realizar la implementación en un proyecto no tenga un impacto alto para las partes que conforman el proyecto.

Para la elaboración de la metodología híbrida se tomaron en cuenta 3 factores importantes que inciden de manera directa en el desarrollo de un proyecto de software como son: Roles del proyecto, Documentación a ser generada, Actividades desarrolladas el proceso.

3 ROLES DEL PROYECTO

De manera directa se puede observar en el análisis de roles que los roles con los que cuenta la metodología scrum no satisfacen de manera directa la implementación del estándar y es por ello que se hace necesario la implementación de nuevos tipos de roles adaptados al estándar, tal como muestra la siguiente tabla:

	ROL	COMPETENCIAS
UPO	UTN - Product Owner	Product Owner/ Cliente
USM	UTN - Scrum Master	Scrum Master / Líder Técnico
UST	UTN - Scrum Team	Scrum Team/ Equipo- Programadores- Desarrolladores
USA	UTN- Scrum Analyst	Analista

Tabla 16 Roles utilizados

Fuente: (Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110, 2013)

Elaborado por: El autor

Cada rol cumple funciones específicas y el rol de analista son realizadas por el rol scrum Team, asignadas como actividades a un miembro del equipo. Cada uno de estos roles debe cumplir con una serie de responsabilidades que contribuyen al desarrollo del proyecto

ROL	RESPONSABILIDADES
Product Owner/ Cliente	<ul style="list-style-type: none"> - Establecer las características del producto - Decidir sobre las fechas de puesta en marcha y los contenidos del producto - Es el responsable de la rentabilidad del producto - Es responsable de establecer una priorización según las características y asignarle un valor - Ajustar las características y prioridades por iteración (según la necesidad). - Acepta o rechaza el producto - Definir los objetivos del proyecto
Scrum Master / Líder Técnico	<ul style="list-style-type: none"> - Velar que los miembros del equipo cumplan con los principios de desarrollo ágil - Facilita las reuniones del equipo - Apoyar al quipo en las dificultades que tengas eliminando los impedimentos - Proteger al equipo de interrupciones
/ Equipo- Programadores- Desarrolladores/ Analista	<ul style="list-style-type: none"> - Priorizar los requisitos para cada iteración - Identificar todas las tareas necesarias para completar un requisito - Estimar el esfuerzo necesario para cada tarea - Cada miembro del equipo se auto asigna tareas - Realiza validación, levantamiento, actualización de requerimientos. - Elabora el modelo de análisis y diseño del sistema. - Interviene en la elaboración del plan de pruebas. - Interviene en las pruebas del sistema

Tabla 17: Roles y Responsabilidades

Fuente: El autor, (Contraloria General de la Republica de Costa Rica, 2017), (ÁGILES, 2017)

Elaborado por: El autor

4 DOCUMENTACIÓN QUE GENERADA

La generación de documentos tiene una diferencia entre el estándar ISO/IEC 29110 y Scrum; donde se evidencia que la documentación que genera scrum es muy reducida e insuficiente para satisfacer el estándar. U-Scrum propone una estructura de los documentos Product Backlog y Sprint Backlog para realizar la formalización del proceso de desarrollo evitando eliminar o reducir la flexibilidad de scrum, además incorpora el documento de gestión de la configuración.

UTN – Scrum	Observación
U-Product Backlog	Artefacto básico, con la información requerida por la descripción del trabajo y el plan de estándar se fue completado con la información inicial del proyecto y actualizado en el transcurso del desarrollo del proyecto
U-Configuración del Software	Documento donde se irán incorporando todos los registros de las actividades realizadas.
U-Sprint Backlog	Incorpora la solicitud de cambio en cada uno de los sprint
U-Acta de aceptación	Documento donde se acepta el producto
U-Acta de reunión	Documento donde se registra las decisiones, condiciones de cada una de las reuniones

Tabla 18: Documentación U-Scrum

Fuente: (Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110, 2013)

Elaborado por: El autor

En base al análisis realizado se determinó que hace falta establecer algunos artefactos adicionales que se deben elaborar utilizando la metodología Scrum, artefactos que no deben afectar al proceso de desarrollo, por lo cual definió las siguientes plantillas o artefactos complementarios.

Estándar ISO/IEC 29110	SCRUM	U-Plantillas
Declaración de Trabajo	Product Backlog	Enunciado del trabajo – ágil
Plan de Proyecto	Product Backlog	Plan de proyecto – ágil
Configuración de Software	-	Plan de proyecto – ágil
Solicitud de Cambio	Spring Backlog	Solicitud de cambio – ágil Diseño de arquitectura – ágil Especificación de Requerimientos – ágil
Registro de Aceptación	-	Resultados de verificación – ágil Resultados de Validación - ágil Acta de aceptación – ágil
Minutas de Reuniones	-	Actas de trabajo – ágil

Tabla 19: Documentos Definidos para el método híbrido

Fuente: (Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110, 2013)

Elaborado por: El autor

5 ACTIVIDADES DESARROLLADAS EL PROCESO.

Las actividades propuestas en U-Scrum propone realizar un ajuste a los procesos de Gestión de Proyectos e Ingeniería de Software se encuentran ajustadas a la estructura de la metodología Scrum.

La Gestión de Proyectos queda establecida en las siguientes etapas Inicio, planificación de proyecto, ejecución y evaluación y cierre; así pues, todo el proceso de ingeniería de software se define con las fases de inicio de implementación, análisis de requerimientos, ejecución y entrega.

Proceso Original	Proceso Ajustado	Observación
Planificación del Proyecto	Inicio	En esta fase se recibe la petición del U- Product Owner y el U-Scrum Master da inicio al proyecto con la elaboración del U - Product Backlog y asignado al analista
Ejecución del plan del proyecto	Planificación del proyecto	EL U-Scrum Master y el analista analizaran la factibilidad, riesgos, tareas a realiza, dependencias, duración, punto de verificación y validación estimara esfuerzos, costos y elaborar el cronograma. Basado en esto se elabora el documento de planificación y las secciones de gestión de la configuración
Evaluación y control del proyecto	Ejecución y evaluación de proyecto	El U-Scrum master y el analista evalúa el avance del proyecto, analiza posibles cambios, estima costos de cambios, elabora nuevos calendarios registra en el la sección de Gestión de la Configuración que está en el documento de planificación
Cierre del Proyecto	Cierre	El U-Scrum master registrar el acta de aceptación del proyecto y finalización del proyecto.

Tabla 20: Propuesta de Método Híbrido - Gestión del Proyecto

Fuente: La Investigación.

Elaborado por: El autor

Para el proceso de Ingeniería de software se establece la siguiente tabla donde se describe la similitud del método híbrido al que se ajustara el proceso de Ingeniería de Software

Proceso Original	Proceso Ajustado	Observación
Iniciación de la Implementación del software	Inicio de implementación y análisis de requerimientos	El U-Scrum Master y el Scrum Team realizan la reunión de inicio donde definen el primer sprint
Análisis de los requerimientos de software	Ejecución	En cada sprint se realizan las actividades de análisis de requerimientos, diseño y prototipado o desarrollo por pruebas, diseñan la arquitectura, codifican y ejecutan las pruebas e integraciones. Después de ello se elabora una pre entrega al cliente final, cabe señalar que estas entregas parciales suman al final del proyecto la entrega total de proyecto.
Diseño Detallado y Arquitectura del Software		
Construcción del Software		
Pruebas e integración del software		
Entrega del producto	Entrega	El U-Scrum master registra la entrega total del proyecto (sumando todos los entregables parciales)
Documentación de Mantenimiento		

Tabla 21: Propuesta de Método Híbrido - Ingeniería de Software

Fuente: La Investigación.

Elaborado por: El autor

Debido a que la Dirección de Desarrollo Tecnológico e Informático es un área que gestiona, administra y desarrolla los proyectos de software y tomando en cuenta que todos los productos de software que son parte del ERP – Universitario, su prioridad es recibir los proyectos de software, desarrollarlos y ponerlos en producción en el tiempo más corto posible debido a que las demás áreas necesitan los productos.

Para la creación de las plantillas se tuvo dos facetas

- a. Creación de las plantillas de productos para proyectos ágiles
- b. Ajustar las plantillas creadas de acuerdo con las necesidades institucionales

De estas dos consideraciones y de la investigación realizada y después de la reunión sostenida con el equipo de la Dirección de Desarrollo Tecnológico e Informático se quedó en elabora las siguientes plantillas para la elaboración del Método Híbrido de Implementación del estándar ISO/IEC 29110 con la metodología de desarrollo Scrum.

1. Enunciado del trabajo – ágil
Anexo 4: Enunciado del trabajo - ágil
2. Plan de Proyecto – ágil
Anexo 5: Plan de Proyecto - ágil
3. Especificación de requerimientos – ágil
Anexo 6: Especificación de requerimientos – ágil
4. Diseño de arquitectura – ágil
Anexo 7: Diseño de arquitectura – ágil
5. Resultados de Verificación (Check List)
Anexo 8: Resultados de Verificación (Check List)
6. Resultados de Validación (Check List)
Anexo 9: Resultados de Validación (Check List)
7. Solicitud de Cambios – ágil.
Anexo 10: Solicitud de Cambios – ágil.
8. Acta de Aceptación – ágil.
Anexo 11: Acta de Aceptación – ágil.

4.3.4 IMPLEMENTACIÓN DE ESTÁNDAR ISO/IEC 29110

La implementación del estándar ISO/IEC 29110 requiere de consideraciones que debe ser tomadas en cuenta para su implementación en el proceso de desarrollo, una de las características más importantes es que el proceso de desarrollo tanto para los Procesos de Gestión(GP) como Proceso de Implementación(IS) no sea intrusivo en la dinámica de la Dirección de Desarrollo Tecnológico e Informática.

Se realizó una reunión con trabajo con el equipo de desarrollo de la Dirección de Desarrollo Tecnológico e Informático para presentar el modelo híbrido que será utilizado

para la implementación y obtener información sobre la opinión del equipo de trabajo y sus criterios sobre el proceso de Gestión de Proyecto e Implementación de Software.

Primeramente, se evaluaron los requerimientos iniciales que se debe cumplir para la implementación del estándar ISO/IEC 29110 y de las cuales todas las condiciones son cumplidas por la Dirección de Desarrollo Tecnológico e Informático; a continuación, se detallan los puntos cumplidos:

- Existe un acta de reunión que funciona como contrato;
- Se realiza una reunión de inicio de proyecto donde se levanta un documento que indica las especificaciones funcionales que debe implementar el software;
- Los proyectos del equipo son de extrema importancia para la Universidad por lo que el tiempo de entrega es un factor crítico y relevante por lo que después de llevarse a cabo la reunión inmediatamente se realiza la codificación del requerimiento y pruebas para poner en producción y que en el transcurso se vayan afinando detalles del proyecto;
- Sus concentraciones se atienden en terminar de desarrollar el software funcionando;
- Después de haber llevado a cabo la reunión de inicio de proyecto se establece los personas responsables y roles que tendrá el proyecto;
- Se realiza reunión para revisar el avance del proyecto y modelado de los requerimientos funcionales;
- Las funcionalidades atendidas según la prioridad que establezca el director de la Dirección de Desarrollo Tecnológico e Informático después de un análisis con los funcionarios solicitantes;

Antes de iniciar la implementación del estándar en el proceso de desarrollo se practicó una encuesta al personal encargado del desarrollo de software de la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte que están conformados por:

- Ing. Juan Carlos García (director)
- Ing. Lenin Chávez
- Ing. Fernanda Rivera
- Ing. Evelin Enríquez
- Ing. Juan Carlos Rodríguez

4.3.4.1 GESTIÓN DE PROYECTO

Para este proceso de estableció un cuestionario para cada proceso, los cuales a continuación se detallan:

➤ Planificación

Pregunta	Respuesta	Observación
¿El equipo conoce el alcance y los objetivos del proyecto?	Generalmente cuando existe una propuesta de un proyecto se realiza una reunión con todos los implicados y se establecen los objetivos a cumplir.	
¿El equipo conoce los requerimientos generales del proyecto?	Existe una reunión previa donde se socializa los requerimientos del proyecto.	
¿Se designan de manera formal roles y responsabilidades en el equipo de proyecto? Ejemplo: analistas, desarrollador, tester, administrador de Base de Datos.	Uno o más roles son ejecutados por una persona debido a que se cuenta con poco personal	
¿Cómo se definen las tareas, y dependencias entre tareas del proyecto?	Según el expertis de cada persona se define las tareas	
¿Cómo se definen las tareas, y dependencias entre tareas del proyecto?	Se elabora un cronograma con las actividades a ser desarrolladas, el cronograma es revisado por el jefe del proyecto.	
¿Existe un documento de plan de proyecto donde se formalice la planificación inicial del proyecto?	Si existe un documento que me ayuda a formalizar el plan del proyecto.	
¿Dónde se registran las tareas para la ejecución del proyecto?	Las tareas se registran en el plan del proyecto y en el cronograma de actividades	

¿Se han asignado recursos a las primeras tareas del proyecto?	Depende del tipo de proyecto y la tarea.	
¿Se valida el plan de proyecto con los Stakeholders?	Si el plan del proyecto es validado para que sea aprobado.	
¿Se ha definido la Gestión de la Configuración a utilizar en el proyecto?	Si se define la gestión de la configuración, el control de versiones tanto de la documentación como de las fuentes.	
¿Cómo se verifica el contenido del plan de proyecto y quienes son los responsables de hacerlo?	Quien se encarga de verificar es el líder del proyecto mediante un análisis conjuntamente con el responsable	
¿Cree usted que la implementación de un estándar a la planificación de un proyecto ayudaría a mejorar el proceso y la calidad del producto?	Si mejoraría la planificación de un proyecto.	

Tabla 22: Cuestionario Gestión de proyecto – planificación

Fuente: Cuestionario Gestión de proyecto – planificación

Elaborado por: El autor

➤ Ejecución Del Plan

Pregunta	Respuesta	Observación
¿El equipo conoce la metodología de trabajo? Ejemplo: RUP, Scrum, eXtreme programming	El equipo conoce la metodología RUP y teóricamente scrum	
¿El equipo de trabajo conoce las tareas que debe desarrollar y el tiempo para culminarlos?	Si el equipo conoce el tiempo de ejecución de cada tarea y el límite para cumplirlo	
¿El equipo mantiene un avance de las tareas desarrolladas?	Si mantiene un avance con ayuda del cronograma de actividades	

¿Con que frecuencia se reúnen los miembros del equipo del proyecto para comunicar su avance sobre el proyecto?	Se reúnen cada semana	
¿Conoce el proceso de recepción de Solicitudes de Cambios de los usuarios?	Si, se recibe a través de un formulario y mediante un oficio enviado a la Dirección de Desarrollo tecnológico e Informático.	
¿Cuál es el proceso a considerar para aceptar o rechazar una solicitud de cambio?	Se mantiene una reunión con los analistas de sistemas para estudiar el caso y resolver de una manera optima	
¿El equipo conoce las políticas de seguridad sobre el manejo de repositorio del proyecto?	Si se conoce, basándose en la norma de control interno y los reglamentos institucionales.	
¿Cómo se comunican los cambios ocurridos en el proyecto?	Se comunica mediante oficios y comunicados a los analistas de sistemas	
¿Cuáles son las políticas de respaldo que usted conoce?	Las definidas de control interno.	

Tabla 23 : Gestión de Proyecto - Ejecución del plan

Fuente: Gestión de Proyecto - Ejecución del plan

Elaborado por: El autor

➤ Evaluación y control del proyecto

Pregunta	Respuesta	Observación
¿Cuáles son los canales de comunicación que conoce el equipo de para difundir algún tipo de novedad?	Se realizan reuniones con todos los involucrados y mediante informes	
¿Con que frecuencia se realizan las reuniones con el equipo del proyecto para resolver problemas?	Cada semana o dependiendo del caso, muchas veces pueden ser diarios	

¿Se cuenta con una bitácora donde se registran los incidentes encontrados en el proyecto y las alternativas de solución?	Si, mediante un listado de situaciones presentadas.	
¿Se actualiza el plan de proyecto para reflejar el avance real del proyecto?	Depende de la forma como se va efectuando el avance, en muchas ocasiones lo realiza el jefe de proyecto	
¿Qué herramientas utiliza para mostrar el avance del proyecto respecto a lo planificado?	Existe un módulo dentro del sistema informático integrado para el registro del avance, corresponde el módulo de planificación	

Tabla 24: Gestión de Proyecto: Evaluación y control del proyecto

Fuente: Cuestionario Gestión de Proyecto: Evaluación y control del proyecto

Elaborado por: El autor

➤ Cierre del proyecto

Pregunta	Respuesta	Observación
¿El equipo conoce la fecha de finalización del proyecto?	Si, este tiempo se planifica al inicio del proyecto a consideración de todos los involucrados	
¿Se elabora un acta de cumplimiento para la entrega recepción del proyecto?	Si, justamente el acta de cumplimiento es el medio de verificación de que ha finalizado el proyecto	
¿Cuál es el proceso de cierre del proyecto?	Una vez que se hayan completado las pruebas y documentado los entregables se realiza una revisión final con el líder de proyecto para luego ponerlo en marcha	
¿Conoce el proceso de versión del proyecto una vez terminado?	Las versiones del proyecto se van registrando dependiendo de las revisiones con el jefe del proyecto e involucrados.	

Tabla 25 Gestión de Proyecto - Cierre del proyecto

Fuente: Cuestionario Gestión de Proyecto - Cierre del proyecto

Elaborado por: El autor

4.3.4.2 IMPLEMENTACIÓN DE SOFTWARE

Para este proceso de estableció un cuestionario para cada proceso, los cuales a continuación se detallan:

➤ **Inicio de implementación del software**

Pregunta	Respuesta	Observación
¿Qué se necesita para iniciar la implementación de un software? Por ejemplo: ¿creación de base de datos, creación de paquetes, establecer la arquitectura de software, etc.?	Se necesita la especificación de requerimientos del software para luego según la metodología a aplicar se decide el diseño y modelado de la base de datos, definición de arquitectura.	
¿Existen tareas iniciales definidas y asignadas para los miembros del equipo?	Se realiza una distribución de tareas a cada uno de los miembros del equipo.	
¿El equipo conoce las tareas iniciales del proyecto?	Si, se realiza en base a reuniones previas.	

Tabla 26: Implementación del Software - Inicio de implementación del software

Fuente: Cuestionario Implementación del Software - Inicio de implementación del software

Elaborado por: El autor

Análisis de los requerimientos de software

Pregunta	Respuesta	Observación
¿Cómo se realizan las reuniones con los stakeholders: ¿presencial, vía telefónica, etc.?	Se realizan de forma presencial	
¿Existe un tiempo definido para las reuniones de levantamiento de requerimientos?	Se, este tiempo es planificado en base a un cronograma establecido previamente	
¿Cómo se realiza la documentación de levantamiento requerimientos, indicados por los stakeholders?	Se sigue el formato de la metodología a aplicar.	
¿Quién se encarga de verificar si el levantamiento de requerimientos es válido?	El líder del proyecto es la persona que se encarga de las verificaciones	
¿Quién es el encargado de priorizar la atención de los requerimientos?	El líder del proyecto	
¿Cómo se valida los requerimientos de una implementación	Mediante las pruebas	
¿Existe una metodología para las reuniones del equipo de trabajo?	Puede utilizarse la metodología de FocusGroup	
¿Con que frecuencia se realizan las reuniones con los stakeholders?	Las reuniones se las realiza dos a tres veces por semana	
¿Cómo se realiza la documentación de los requerimientos?	Siguiendo la metodología de desarrollo que se aplicó al proyecto en ciertos casos RUP	
¿Se concentran todos los requerimientos de los stakeholders dentro de un solo documento de especificación de requisitos?	Si, el documento me permite conocer todos los requerimientos del sistema	

¿Se presenta evidencia de la verificación de la correcta documentación de los requerimientos?	Si, se realiza una presentación al usuario final para su verificación y validación	
¿Se registra un histórico de los cambios que los stakeholders han solicitado sobre el producto?	Si, mediante ficha de requerimientos	
¿Existe un ambiente de pruebas, test, desarrollo para mostrar los productos a los stakeholders o se utiliza el mismo ambiente de desarrollo?	Es el ambiente de desarrollo.	
¿Se realiza un análisis de integración entre los productos de la organización?	Si se realiza un análisis para la integración de los procesos y entre los módulos	
¿En qué momento se actualiza el manual de usuario?	Cuando existe un nuevo requerimiento y se implementa en el aplicativo.	

Tabla 27: Implementación de software - Análisis de los requerimientos de software

Fuente: Cuestionario Implementación de software - Análisis de los requerimientos de software

Elaborado por: El autor

➤ **Diseño detallado y arquitectura del software**

Pregunta	Respuesta	Observación
¿Se posee un conjunto de tecnologías definido por cada tipo de proyecto de Software? Ejemplo: Aplicación web, aplicación móvil, aplicación de escritorio.	Se utiliza de acuerdo al uso y usuario por ejemplo los portafolios están orientados a la web y a los móviles	
¿Cómo se evidencia el cumplimiento de los requisitos no funcionales?	Mediante pruebas de estrés	
¿Conoce si el sistema se comunica con otros sistemas existentes?	Si debido a que es una solución integrada, con una base de datos centralizada.	

¿Conoce si se preverá servicios existentes en la Universidad?	Esto es de acuerdo a las necesidades institucionales y los requerimientos de usuarios siempre tratando de brindar el mejor servicio.	
¿Se cuenta con un diseño de la infraestructura de despliegue del sistema? Ejemplo: nodo cliente, nodo servidor, nodo de base de datos, firewall.	Si, se cuenta con infraestructura	
¿Cuenta con alguna herramienta CASE, que se utilice para el diseño y modelado del sistema?	Si, estas herramientas ayudan al modela del software	
¿Cómo se realiza el diseño de los componentes de software, los módulos, componentes del software e interfaces?	Mediante herramientas adecuadas que permiten la facilidad y rapidez de momento de diseño. Ejemplo Enterprice Architect	
¿Se tiene un formato para documentar la arquitectura y diseño detallado del software?	Si de acuerdo a la metodología RUP	
¿En qué fase del proyecto se consolida la documentación de la arquitectura del sistema?	En la fase de inicio de la metodología RUP	
¿Dónde se almacenan los casos de pruebas y procedimientos de pruebas?	No se ejecuta los casos de pruebas solo se ejecutan y se informa los resultados	
¿Cómo se verifica la trazabilidad entre los componentes de software hacia los requerimientos?	Según las iteraciones	
¿Qué tipo de pruebas se realizan por componentes de software?	Corresponden a pruebas de caja negra o funcionales	
¿Cuál es el proceso de pruebas que se realiza para cada componente de software?	Se encuentran definidas por la metodología de desarrollo	
¿Quién diseña los casos de pruebas y procedimientos de pruebas?	La unidad de Test o pruebas	
¿Con que frecuencia se elabora los documentos de casos de uso, arquitectura	Frecuentemente, se elaboran todos los documentos indicados.	

de software, casos de pruebas, procedimientos de pruebas, etc.?		
---	--	--

Tabla 28 Implementación de software - Diseño detallado y arquitectura del software

Fuente: Cuestionario Implementación de software - Análisis de los requerimientos de software

Elaborado por: El autor

➤ **Construcción del Software**

Pregunta	Respuesta	Observación
¿Existe un estándar de codificación?	Si, definido mediante la metodología y según los estándares definidos por la Dirección	
¿Cuenta con estándares para el desarrollo de componentes de Base de Datos?	Existes licenciamiento para definición de comportamientos	
¿Cuentan con componentes reutilizables por proyectos?	Existen componentes reutilizables.	
¿Cuentan con componentes reutilizables por proyectos?	Si, se respaldan la versión anterior y se pone en producción la nueva versión	
¿En qué momento se diseñan las pruebas unitarias?	En el transcurso del desarrollo	
¿En qué momento se ejecutan las pruebas unitarias?	En el transcurso del desarrollo	
¿Existen reglas para definir el desarrollo de los componentes de software como terminado? Ejemplo: código compatible, cubierto el 90% y añadido al repositorio de versionamiento	Si, las reglas son definidas por el director de la Dirección	

Tabla 29: Implementación de software - Construcción del Software

Fuente: Cuestionario Implementación de software - Construcción del Software

Elaborado por: El autor

➤ **Pruebas e Integración del software**

Pregunta	Respuesta	Observación
¿Cómo se realiza la integración de componentes de software desarrollados por el equipo de desarrollo?	La integración se realiza mediante paquetes de base de datos	
¿En qué ambiente se despliega el software integrado para las pruebas?	En el ambiente de producción antes de la puesta en marcha	
¿Cuántos ambientes poseen para desarrollar un proyecto de software?	Un ambiente de desarrollo	
¿Cuál es el procedimiento de despliegue de un proyecto de software?	Deploy el comprimido de la aplicación o sincronización de los ejecutables	
¿Cuál es el procedimiento en caso de fallar un despliegue de un software integrado?	Volver a realizar la iteración del despliegue hasta que salga correcto	
¿Qué tipo de pruebas se usan antes de realizar el despliegue de un producto de software?	Pruebas de caja negra y funcionales	
¿Quién es el encargado de ejecutar las pruebas al software?	El equipo de tester	
¿Cuáles son los criterios de aceptación de ejecución de las pruebas? Ejemplo: 100% de casos de pruebas exitosos, etc.?	Con un porcentaje del 95% de aceptación del usuario final	
¿Cuál es el proceso de declaración de un error, fallo del sistema? ¿El tester encuentra un problema, e desarrollador lo corrige...?	En caso de encontrar un error o fallo se procederá la revisión del código fuente para resolver el problema	
¿Cuál es el proceso de verificación de trazabilidad entre los casos de pruebas y el procedimiento de pruebas ejecutado hacia los módulos o componentes de software implementados?	Mediante el cumplimiento de los requisitos de software	

Tabla 30 Implementación de software - Pruebas e Integración del software

Fuente: Cuestionario Implementación de software - Pruebas e Integración del software

Elaborado por: El autor

➤ **Entrega del producto**

Pregunta	Respuesta	Observación
¿Los manuales de operación de mantenimiento se encuentran documentados?	Si se realiza la documentación según los lineamientos de la metodología que se aplica	
¿Los manuales de operación y de mantenimiento se encuentran verificados?	Se realiza una revisión de los manuales para verificar su correcta elaboración	
¿Los manuales de operación y de mantenimiento se encuentran verificados?	Si se encuentra documentado	
¿El manual de usuario se encuentra verificado y validado?	Si se encuentra verificado	
¿El sistema se encuentra listo para el despliegue en el ambiente de producción?	Si	
¿Quién y cómo se verifica que se cumplan los requisitos de entrega indicados en el acta de aceptación?	Con los usuarios finales	
¿Cuál es el proceso de atención de las solicitudes de modificación solicitada por los stakeholders luego del despliegue del sistema en producción?	Mediante oficios y fichas	

Tabla 31 Implementación del software - Entrega del producto

Fuente: Entrevista con el director de la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte

Elaborado por: El autor

Conclusión:

Al realizar el análisis de la encuesta planteada y realizando un análisis cualitativo se determinó que la Dirección de Desarrollo Tecnológico e innovación de la Universidad Técnica del Norte cuenta con las facilidades para implementar un software, pero que el tiempo de entrega de los productos no debe ser de larga duración ya que los usuarios solicitan un producto por la necesidad de gestionar la información, entregar información a la unidades requirentes sean estas internas como a las entidades de control. Además de contar con un equipo de desarrollo pequeño acorde a las necesidades del estándar.

4.3.5 IMPLEMENTACIÓN DEL ESTÁNDAR 29110 – PORTAFOLIO MÓVIL ESTUDIANTIL – UTN-MÓVIL

Con la finalidad de probar la implementación del estándar ISO/IEC 29110 en la Gestión del Proceso de Desarrollo de Software de la Dirección de Desarrollo Tecnológico e Informático se estableció juntamente con el director de la Unidad que se desarrolle un prototipo de aplicación móvil para la Universidad y que únicamente de desarrollo el módulo de consulta de notas.

Al ser un desarrollo prototipo, se definió que el aplicativo móvil se quedaría como su nombre lo indica de prototipo y que no será necesario dejarlo implementado en un ambiente de producción debido a que la universidad se encontraba tramitando un proyectos de aplicación móvil, pero esto cuando se inició el proyecto no se conocía, es más el nuevo proyecto móvil nació de la propuesta del presente proyecto.

Para el desarrollo del prototipo se establecieron las siguientes actividades:

- a. Definición de Servicios Rest a utilizar

Para el proyecto se solicitó a la Dirección de Desarrollo Tecnológico E Informático se exponga 2 servicios web; uno para gestión de la autenticación y el segundo para consulta de las calificaciones

Enlace	Descripción
http://app.utn.edu.ec:9071/ords/acad_micoa5/apk/login?id=xxxxxxxxx&pw=Exxxxxx xxx	Este servicio me permite enviar el usuario y el password y verificar que exista las credenciales
http://app.utn.edu.ec:9073/ords/acad_mico/notas/notas/xxxxxxxxx	Este servicio me permite enviando como parámetros un identificador consultar las notas del semestre actual

Tabla 32: Servicios Expuestos

Fuente: Universidad Técnica del Norte

Elaborado por: El Autor

Todas las transacciones se realizaron en formato JSON para agilizar el tiempo de respuesta.

- b. Llenado de documentación del proyecto

Se llenaron los siguientes artefactos:

- Enunciado del trabajo – ágil
Anexo 12: Enunciado del Trabajo
- Plan de Proyecto – ágil
Anexo 13: Plan de Proyecto – ágil
- Especificación de requerimientos – ágil
Anexo 14: Especificación de Requerimientos - ágil
- Diseño de arquitectura – ágil
Anexo 15: Diseño de arquitectura – ágil
- Resultados de Verificación (Check List)
Anexo 16: Resultados de Verificación – ágil
- Resultados de Verificación (Check List)
Anexo 17: Resultados de Validación

c. Definición de Herramientas.

Para el desarrollo del prototipo, que es una aplicación móvil se establece las siguientes herramientas:

Especificación	Valué
Herramienta IDE	Android Studio
Lenguaje	Java
Sistema Operativo Cliente	Android v5

Tabla 33Detalle de Herramientas

Fuente: El Autor

Elaborado por: El Autor

d. Desarrollo de Aplicación

El desarrollo se lo llevo a cabo, como parte de mostrar la funcionalidad de la metodología y el Estándar.

- **Pantalla de Login**



Figura 28: Pantalla de Login Demo

Fuente: El Autor

Elaborado por: El Autor

- **Pantalla de Menú de opciones**



Figura 29: Menú de Opciones app

Fuente: El Autor

Elaborado por: El Autor

- **Procesando consulta de Calificaciones**

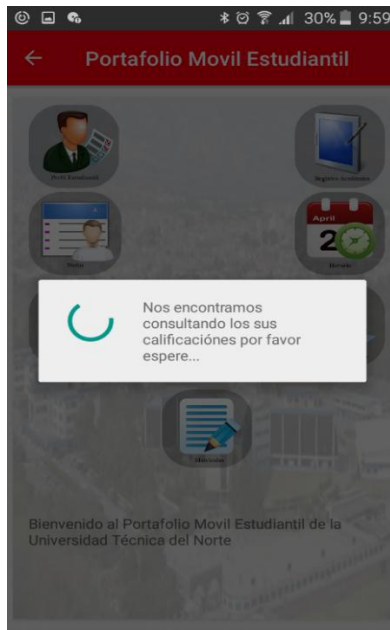


Figura 30: Procesando consulta al servicio web

Fuente: El Autor

Elaborado por: El Autor

- **Pantalla de presentación de calificaciones**

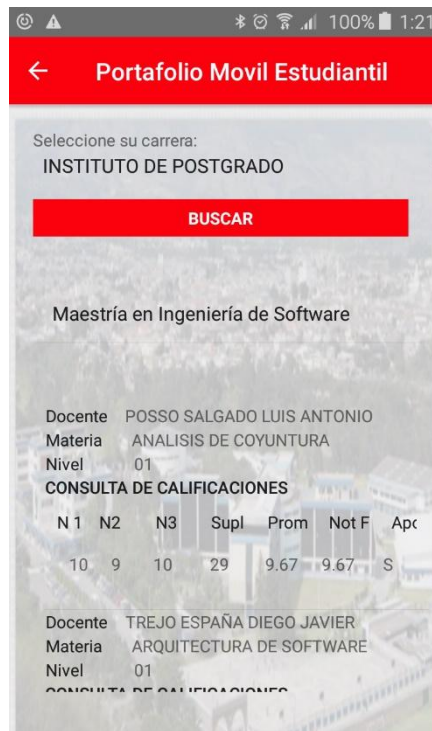


Figura 31: Pantalla de Consulta Calificaciones

Fuente: El Autor

Elaborado por: El Autor

4.3.6 ANÁLISIS DE IMPACTO DE IMPLEMENTACIÓN DE ESTÁNDAR ISO/IEC 29110- PERFIL BÁSICO

En la aplicación de este trabajo de investigación los posibles impactos que originan son: económico, administrativo, cultural, los cuales pueden ser de carácter positivo o negativo.

El impacto es medido utilizando la siguiente tabla de ponderaciones identificando las variables que la generan

VALORACIÓN DE IMPACTOS	
Nivel de Impacto	Calificación
Impacto alto negativo	-3
Impacto medio negativo	-2
Impacto bajo negativo	-1
No existe impacto	0
Impacto bajo positivo	1
Impacto medio positivo	2
Impacto alto positivo	3

Tabla 34: Matriz de medición de impacto

Fuente: La Investigación

Elaborado por: El Autor

Para el cálculo se utiliza la siguiente fórmula:

$$\text{Nivel de Impacto} = \frac{\epsilon \text{ de Calificación}}{\text{Número de Indicadores}}$$

$$NI = \frac{\epsilon}{n}$$

Donde:

- **NI:** Nivel de impacto
- **Σ=** Sumatoria de calificaciones.
- **n=** Número de indicadores

4.3.6.1 IMPACTO ADMINISTRATIVO

Nro.	Indicadores	Nivel de Impacto							Total
		-3	-2	-1	0	1	2	3	
1	Políticas							X	3
2	Manuales							X	3
3	Establecimiento de formatos							X	3
		Sumatoria Total							9
Impacto:		Alto positivo (9/3)							3

Tabla 35 Tabla de indicadores Impacto Administrativo

Fuente: La Investigación

Elaborado por: El Autor

Nivel de impacto 3.

La aplicación del estándar ISO/IEC 29110 a la gestión del proceso de Desarrollo de Software que se propone a la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte tiene un impacto alto positivo, apoya a dar el seguimiento, mantener control de los proyectos y aporta a la toma temprana de decisiones que ayuden al desarrollo y posterior culminación del proyecto.

4.3.6.2 IMPACTO ECONÓMICO

Nro.	Indicadores	Nivel de Impacto							Total
		-3	-2	-1	0	1	2	3	
1	Incremento de la rentabilidad							X	3
2	Incremento productividad							X	3
3	Acceso a nuevos mercados							X	3
4	Mejoramiento de la calidad de productos y producciones							X	3
		Sumatoria Total							12
Impacto:		Alto positivo (12/4)							3

Tabla 36: Indicadores impacto Económico

Fuente: La Investigación

Elaborado por: El Autor

Nivel de impacto 3

La aplicación del estándar ISO/IEC 29110 a la gestión del proceso de Desarrollo de Software que se propone a la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte tiene un impacto alto positivo, incrementa la rentabilidad del producto ya que mejora la calidad del producto y por consiguiente abre las puertas a nuevos mercados volviendo competitiva a nivel regional.

4.3.6.1 IMPACTO CULTURA - ORGANIZACIONAL

Nro.	Indicadores	Nivel de Impacto							Total
		-3	-2	-1	0	1	2	3	
1	Transformación cultura y clima empresarial							X	3
2	Nuevas metodologías de trabajo							X	3
3	Nuevos sistemas de gestión							X	3
		Sumatoria Total							9
Impacto:		Alto positivo (9/3)							3

Tabla 37 : Indicadores de impacto Cultural - Organizacional

Fuente: La Investigación

Elaborado por: El Autor

Nivel de impacto 3

La aplicación del estándar ISO/IEC 29110 a la gestión del proceso de Desarrollo de Software que se propone a la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte tiene un impacto alto positivo, promueve la concientización y el trabajo en equipo cambiando y transformado el clima y cultura organizacional.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- 1) En la Gestión del Proceso de Desarrollo de Software que tiene la Dirección de Desarrollo Tecnológico e Informático no cuenta con actividades de generación de documentación para llevar a cabo la formalidad de un proyecto; lo cual provoca deficiencias en el seguimiento y control de la gestión del proyecto y limita la definición de hitos de control que ayuden a la toma de decisiones que aporte a la entrega del producto.
- 2) El estándar ISO/IEC 29110 se aplica como una alternativa para las pequeñas organizaciones a nivel mundial, teniendo un crecimiento exponencial en gran parte en América del Sur, Canadá y México; apoyando a la formalidad a los productos de software elaborados y brindando atributos de calidad que fortalecen la competitividad de las pequeñas organizaciones.
- 3) El estándar ISO/IEC 29110 por sí solo no se adapta de manera natural a una metodología de desarrollo de software ágil como SCRUM, esto por la naturaleza de la metodología de priorizar la entrega del producto antes que a formalidad; no obstante, para su implementación se puede desarrollar una metodología Híbrida que adopte características, roles y funciones de SCRUM y además adopte criterios de documentación.
- 4) La Investigación de campo ayuda a determinar las falencias que existen en la generación de documentación de la Gestión del Proceso de Desarrollo de Software y la formalidad de las actividades desarrolladas.
- 5) El Método Híbrido facilita la implementación del estándar ISO/IEC29110 con la metodología SCRUM permitiendo adjuntar los roles, procesos y actividades del estándar a la metodología sin ser intrusiva en la Gestión del Proceso de Desarrollo de Software que tiene la Dirección de Desarrollo Tecnológico e Informático.
- 6) La Generación de plantillas apoya a la gestión del proceso de desarrollo de software de la Dirección de Desarrollo Tecnológico e Informático brindando atributos de calidad y manteniendo documentado un proyecto de software fortaleciéndola formalidad de un proyecto.

RECOMENDACIONES

- 1) Establecer actividades que permiten llevar la formalidad en la Gestión del Proceso de Desarrollo de Software que tiene la Dirección de Desarrollo Tecnológico e Informático que permita aportar a la definición de hitos de control en un proyecto de software.
- 2) Con la publicación del estándar ISO/IEC 29110 por la ISO, se recomienda realizar un estudio, seguimiento e implementación en la Gestión del Proceso de Desarrollo de Software de las Pequeñas organizaciones a nivel del Ecuador con la finalidad de poder tener competitividad a nivel internacional y ubicar a los productos de software desarrollados en Ecuador a estándares internacionales.
- 3) Llevar un control de las falencias en el proceso de desarrollo e implementar planes de mitigación que permitan solucionar de manera permanente y apegada a las directrices establecidas por el estándar ISO/IEC 29110.
- 4) Controlar la carga de trabajo para el personal que trabaja en la Dirección de Desarrollo Tecnológico de la universidad Técnica del norte debido a que con la implementación de la metodología Scrum y el estándar será un cambio brusco desarrollar tareas con por tiempos.
- 5) Aplicar el estándar ISO/IEC 29110 con la metodología Scrum en los proyectos CRM que desarrolle la Dirección de Desarrollo Tecnológico e Informático a fin de mantener proyectos formalmente establecidos y documentados.
- 6) Automatizar las plantillas del estándar ISO/IEC 29110 y la Gestión del Proceso de Desarrollo de Software del a fin de poder contar con una herramienta informática que facilite el acceso al registro, consulta de la información de los proyectos de software.

CAPÍTULO VI: REFERENCIAS BIBLIOGRÁFICAS

BIBLIOGRAFÍA

- Cuéllar Chacón, C., & Orduz Tarazona, R. (2015). Modelos de Calidad en el Desarrollo de Software. *Formación en Ambientes Virtuales de Aprendizaje*.
- Acebo, M., & Nuñez, A. (2017). ESTUDIOS INDUSTRIALES ORIENTACIÓN ESTRATEGICA PARA LA TOMA DE DEISIONES. *ESCUELA SUPERIOR POLITECNICA DEL LITORAL*.
- ÁGILES, P. (2017). *PROOYECTOS ÁGILES*. Obtenido de PROOYECTOS ÁGILES: <https://proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting/>
- Alaimo, D. M. (2013). *PROYECTOS ÁGILES CON SCRUM*. BUENOS AIRES: KLEER.
- Arias, F. (2006). *El proyecto de investigación, Introducción a la metodología*. Caracas: Venezuela.
- Baldoneo, J., & Montequín Rodríguez, V. (2017). *Modelo CMMI y métodos ágiles en la gestión de proyectos software*. Oviedo: Univeridad de Oviedo.
- Becerra, P., & Sanjuan, M. (2014). Revisión de Estado del Arte del Ciclo de Vida de Desarrollo de Software Seguro con la Metodología SCRUM. *I+D en TIC Universidad Simón Bolívar*.
- Bravo Carrasco, J. (2011). *Gestión de Procesos*. Evolucion.
- Calderón Macías, F., Martínez Munné, X., & Escofet, C. (2016). *El Estándar ISO y su Aportación al Proceso de Calidad del Desarrollo de Software*. Catalunya: Universidad de Catalunya.
- Cerrano Sánchez, J. (2012). *Metodos de Investigación*. Madrid.
- Cerrano, J. (2012). *Los Métodos de Investigación*. Dias - Santos.
- CHAVÉZ GUAMIALAMA, L. (2015). *SISTEMA DE ENTORNO VIRTUAL ENSEÑANZA - APRENDIZAJE DE LA UNIVERSIDAD TÉCNICA DEL NORTE CON LA IMPLEMENTACIÓN DE LOS MÓDULOS DE CUESTIONARIOS, INTERACCIÓN ESTUDIANTE MAESTRO, SEGURIDAD Y AUDITORIA*. IBARRA: UNIVERSIDAD TÉCNICA DEL NORTE.
- Contraloría General de la Republica de Costa Rica. (12 de 2017). *Rational Unifiend Process*. Obtenido de Rational Unifiend Process: http://cgrw01.cgr.go.cr/portal/page?_pageid=413,2070526&_dad=portal&_schema=PORTAL
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2009). INFORMACIÓN BÁSICA DE SCRUM. *Certified Scrum Training Worldwide | www.ScrumTI.com*.
- Expreso. (2017). La industria de software crece el 17 % anual pese a trabas para internacionalizarse. *Diario Expreso*, <http://www.expreso.ec/economia/la-industria-de-software-crece-el-17-anual-pese-a-trabas-para-internacionalizarse-EB1175622>.
- Flores, E. (2014). *Apuntes USACH*. Obtenido de Apuntes USACH: <http://apuntesusach.herokuapp.com/tutorial/scrum>
- Fuggetta, A. (2000). Software process: a roadmap. *ACM*, 25-34.
- García, M. C., Garzás, J., & Piattini, M. (s.f.). LA MEJORA DE PROCESOS EN PEQUEÑAS EMPRESAS Y LA ISO/IEC 29110. *KYBELECONSULTING*.
- Gomez, K. (27 de Jul de 2017). *Top 5 Metodologías de Desarrollo de Software*. Obtenido de Top 5 Metodologías de Desarrollo de Software:

- <https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/metodologias-de-desarrollo-de-software>
- Hernadéz, G., Gonzalez, W., & Laporte, C. (2012). *Entry profile-DP-Proyect Manager*. Canada.
- Hernández, R., Fernández, C., & Baptista, P. (2006). *Metodología de la Investigación*. Mexico: McGraw Hill.
- Institute, S. E. (2010). *CMMI® para Desarrollo, Versión 1.3*. Software Engineering Process Management Program.
- ISO. (24 de noviembre de 2014). *NUEVAS NORMAS ISO 9001:2015*. Obtenido de <http://www.nueva-iso-9001-2015.com/2014/11/iso-9001-entendiendo-enfoque-basado-procesos/>
- ISO 27207. (2015). *Space data and information transfer systems -- CCSDS Space Link Protocols over ETSI DVB-S2 Standard*. Switzerland: ISO.
- ISO25000. (s.f.). *La familia de normas ISO/IEC 25000*. Obtenido de La familia de normas ISO/IEC 25000: <http://iso25000.com/index.php/normas-iso-25000>
- Laporte, C., Séguin, N., Villas Boas, G., & Buasung, S. (2013). Aprovechando las ventajas de las normas de ingeniería de software y sistemas. *Revista ISO Focus*.
- Larrondo Petrie, M., Medina García, V., & Méndez Giraldo, G. (2009). Modelo de Registro y Acreditación de Instituciones de Educación Superior basado en el Modelo CMMI. *Seventh LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2009)*.
- Mariño, S., & Alfonzo, P. (2017). Ingeniería de software basado en evidencia: soportes como producto académico. *Revista Venezolana de Información Tecnología y Conocimiento*, 87-99.
- Miguel, B. R. (2017). Mejoramiento de procesos, basa prácticas. Caso: Área de De Informática de la PUCE. *Universidad Andina Simón Bolívar*.
- Miranda Realpe, J. (2017). "HERRAMIENTAS DE GESTIÓN PARA LA TOMA DE DECISIONES DE LA DIRECCIÓN DEL DEPARTAMENTO DE TECNOLOGÍAS DE LA UNIVERSIDAD TÉCNICA DEL NORTE". Ibarra.
- Morales, O. (2016). Departamento de Investigación. *Revista. Fundamentos de la Investigación Documental y la Monografía*.
- MUÑOZ, L. (2000). *GUÍA PRÁCTICA PARA LA SELECCIÓN E IMPLEMENTACIÓN. GESTION.COM*.
- NYCE. (2016). *NYCE SEGURIDAD Y CONFIANZA*. Obtenido de NYCE SEGURIDAD Y CONFIANZA: <https://www.nyce.org.mx/ertificacion-iso-29110/>
- Pairazamán Díaz, J. J., & Escobar Nuñez, G. D. (2016). Definición e Implementación del Servicio de Evaluación de los procesos implicados en el modelo MoProSoft en las empresas virtuales de la Escuela de Ingeniería de Sistemas y Computación de la UPC. *Universidad Peruana de Ciencias Aplicadas (UPC)*.
- Pasini, A., Esponda, S., Boracchia, M., & Pesado, P. (2013). Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110. *Instituto de Investigación en Informática LIDI (III-LIDI)*.
- Pasini, A., Esponda, S., Boracchia, M., & Pesado, P. (s.f.). Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110. *Comisión de Investigaciones Científicas de la Provincia de Bs. As.*
- Pressman, R. (2010). *Ingeniería de Software - Enfoque Practico*. Mexico: McGrawHill.
- Pressman, R. S. (2012). *Ingeniería de Software Enfoque practico*. Mexico DF: McGrawHil.
- Ruiz, F. (s.f.). *Ingeniería de Software I*. Obtenido de Ingeniería de Software I: <https://www.ctr.unican.es/asignaturas/is1/is1-t02-trans.pdf>

- Ruvalcaba, M. (21 de julio de 2017). *Procesos de Software*. Obtenido de <https://sg.com.mx/revista/1/procesos-software#.WWWRdog1-M8>
- Sánchez, S. (2012). Criterios para la Adaptabilidad de Estándares y Modelos de Procesos de software en PYMES Ecuatorianas. *Universidad Politécnica Salesiana de Ecuador*, 6.
- Sanchez., L. G. (2017). *FOSFTENG YOUR COMPETITIVE ADVANTAGE*. Obtenido de <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>
- Satpathy, T. (2016). *CUERPO DE CONOCIMIENTO DE SCRUM*. SCRUMstudy™, una marca de VMedu, Inc.
- SOFTENG. (2015). *Metodología Scrum para desarrollo de software - aplicaciones complejas*. Obtenido de <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>
- SOLARTE, G., MUÑOZ, L., & ARIAS, B. (2009). MODELOS DE CALIDAD PARA PROCESOS DE SOFTWARE . *Scientia Et Technica*.
- SOMMERVILLE, I. (2006). *INGENIERIA DE SOFTWARE*. Madrid: 2005 por PEARSON EDUCACIÓN S.A.
- SUPERINTENDENCIA DE COMPAÑIAS, V. Y. (2017). *SUPERINTENDENCIA DE COMPAÑIAS, VALORES Y Y SEGUROS*. Obtenido de SUPERINTENDENCIA DE COMPAÑIAS, VALORES Y Y SEGUROS: <http://appscvs.supercias.gob.ec/rankingCias/>
- Torres Garibay, R., & Armenta Fonseca, L. (10 de Abril de 2016). *Virtual conference*. Obtenido de <https://sg.com.mx/sgvirtual/sesion/propuestas-mejoras-y-actualizaciones-al-modelo-procesos-moprosoft#.WWdB14g1-Cg>
- Valdebenito Espinosa, C. (2016). *Guía Metodológica para la implementación del estándar ISO/IEC 29110 en Pymes TICs de Colombia*. Medellín: Facultad de Ingeniería en Tecnología de la Información y la Comunicación - Universidad Pontificia Bolivariana.
- Valdebenito Espinosa, C. (2016). *Guía Metodológica para la implementación del estándar ISO/IEC 29110 en Pymes TICs de Colombia*. Medellín.

ANEXOS 1 – METODOLOGÍA SCRUM

Es una metodología de gestión de proyectos que aplica un conjunto de buenas prácticas para trabajar colaborativamente, para obtener los mejores resultados posibles de un proyecto, según la empresa SOFTENG indica que Scrum es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa. Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, autogestión e innovación” (SOFTENG, 2015).

En la metodología se realizan entregas parciales y regulares del producto final, dando prioridad por el beneficio significativo que tengan para el proyecto y el usuario final, Scrum está diseñada para entornos complejos donde es necesario obtener resultados en corto tiempo y donde los requisitos son cambiantes o poco definidos y la innovación complejidad flexibilidad y productividad son prioridad para la organización.

PRINCIPIOS DE SCRUM

Scrum es el modelo más utilizado dentro de las metodologías ágiles. Muchos de los valores y principios del manifiesto ágil tienen su origen en SCRUM. (Alaimo, 2013)

Los valores que menciona el manifiesto son:

- Individuos e iteraciones por sobre procesos y herramientas
- El software debe estar funcionando por sobre la documentación generada
- Colaborar con el cliente por encima de la negociación de contratos.
- Respuesta al cambio encima del seguimiento de un plan

BENEFICIOS

La metodología scrum da los siguientes beneficios:

- **Cumplimiento de expectativas**
Los usuarios establecen sus expectativas al mismo tiempo que aportan con los requerimientos funcionales para el proyecto y con esta información el Product Owner establece sus prioridades.

- **Flexibilidad de cambios**

La flexibilidad que ofrece scrum debido a la capacidad de reacción ante los cambios de los requerimientos generados o por los cambios del mercado; esta metodología se adapta a la variación.

- **Reducción del Time to Market**

La metodología le permite al cliente iniciar el uso del proyecto desde sus primeras fases y sin tener que esperar que finalice.

- **Mayor calidad del software**

El cliente puede empezar a realizar las pruebas en el producto e ir ajustando según la necesidad institucional hasta lograr tener un producto de calidad que cumpla los requerimientos solicitados, con cada iteración se puede tener calidad superior.

- **Mayor productividad**

Gracias a la eliminación de burocracia y a la motivación del equipo la organización puede desarrollar producto de más rápida manera.

- **Maximiza el retorno de la inversión**

Producción de software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión.

- **Predicciones de tiempo**

Mediante esta metodología se conoce la velocidad media del equipo por sprint con lo que consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad que todavía está en el Backlog.

- **Reducción de riesgos**

El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada.

ROLES

Scrum está diseñado para un equipo de trabajo reducido, el equipo se focaliza a construir software de calidad, por ello se centra en definir las características específicas que debe tener el sistema.



Figura 32: Roles de Scrum

Scrum maneja los siguientes roles:

1. SCRUM MASTER

Son las personas que lideran el equipo, guiándolo para que cumplan las reglas y procesos de la metodología, las responsabilidades y aptitudes son:

- Resolver conflictos que entorpezcan el progreso del proyecto
- Conocer la metodología Scrum
- Ampliar vocación de servicio
- Analítico y observador
- Saber incentivar y motivar
- Capacidad docente e instructiva

- Buen carisma para las negociaciones

2. PRODUCT OWNER (PO)

El Dueño de Producto es la única persona autorizada para decidir sobre cuáles funcionalidades y características funcionales tendrá el producto. Es quien representa al cliente, usuarios del software y todas aquellas partes interesadas en el producto.

Las funciones y responsabilidades son:

- Canalizar las necesidades del negocio, sabiendo "escuchar" a las partes interesadas en el producto y transmitir las en "objetivos de valor para el producto", al scrum team.
- Maximizar el valor para el negocio con respecto al Retorno de Inversión (ROI), abogando por los intereses del negocio.
- Revisar el producto e ir adaptándole sus funcionalidades, analizando las mejoras que éstas puedan otorgar un mayor valor para el negocio.

4. TEAM

Es el equipo de desarrolladores multidisciplinario, integrado por programadores, diseñadores, arquitectos, testers y demás, que en forma auto-organizada, será los encargados de desarrollar el producto.

Las funciones y responsabilidades son:

- Llevar el Backlog de producto, a desarrollos potencialmente funcionales y operativos.
- Aptitudes que deben tener los integrantes de un Scrum Team:
- Ser profesionales expertos o avanzados en su disciplina
- Tener "vocación" (la buena predisposición no alcanza) para trabajar en equipo
- Capacidad de autogestión

DOCUMENTOS QUE GENERAR EN SCRUM

La metodología solicita que se generen los siguientes documentos:

Documento	Descripción
Product	Es un documento de alto nivel para todo el proyecto.
Backlog	Contiene descripciones genéricas de todos los requerimientos, funcionales y no funcionales, contiene estimaciones realizadas a grandes rasgos, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido, la prioridad de las diferentes tareas, etc. Es dinámico, nunca está completo, evoluciona junto con el producto
Sprint Backlog.	Documento detallado que contiene las tareas que el Team va a implementar durante el presente sprint.

Tabla 38: Documentos SCRUM

Fuente: (Pasini, Esponda, Boracchia, & Pesado, *Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110*, 2013)

Elaborado por: El Autor

PROCESO

El proceso que utiliza scrum se ejecuta en bloques temporales cortos y fijos denominadas iteraciones, algunos de los equipos realizan iteraciones de 3 a 4 semanas, límite máximo de retroalimentación y análisis. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo cuando lo solicite.

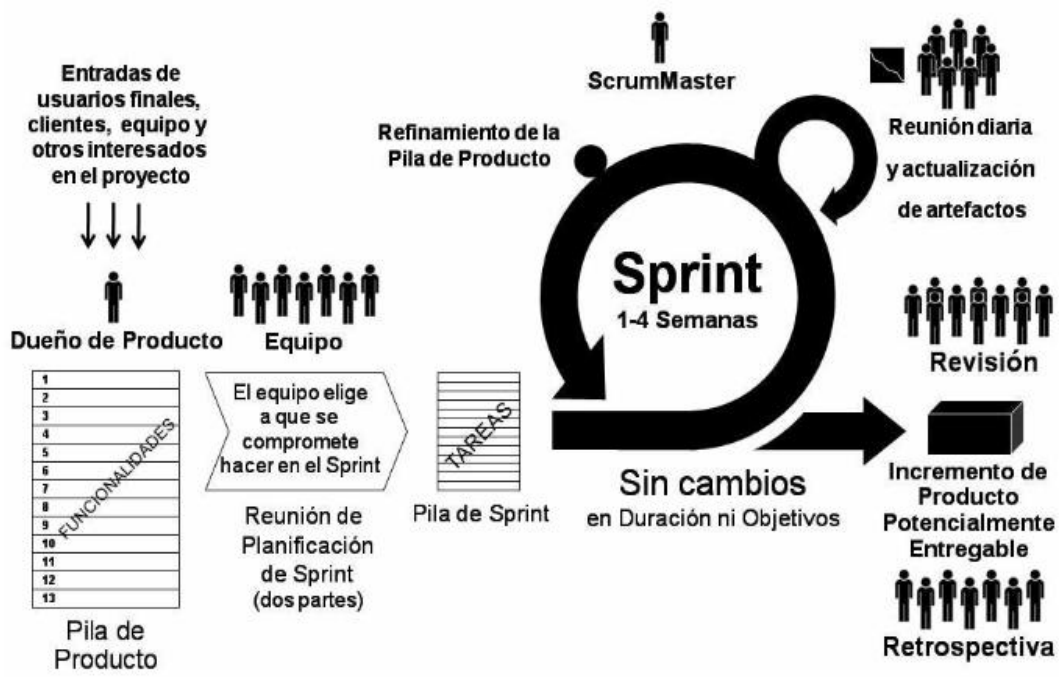


Figura 33 Metodología Scrum

Fuente: (Deemer, Benefield, Larman, & Vodde, 2009).

Elaborado por: (Deemer, Benefield, Larman, & Vodde, 2009).

El proceso parte desde un listado de objetivos/ requisitos priorizados del producto, en esta lista el cliente prioriza los objetivos balanceando el valor que le aportan, las actividades que se llevan a cabo en scrum son las siguientes:

1. SELECCIÓN DE REQUERIMIENTOS

El cliente presenta al equipo de trabajo el listado de requerimientos priorizada del proyecto; después el equipo consulta dudas surgidas y selecciona los requerimientos más prioritarios para completar en la primera iteración de manera que puedan ser entregados si el cliente lo necesitan.

5. PLANIFICACIÓN DE LA ITERACIÓN

El equipo elabora el listado de tareas de la iteración, estas serán las necesarias para poder cumplir las necesidades planificadas en para la primera iteración. La estimación des

esfuerzo se la realizara de manera conjunta y los miembros del equipo se auto asignan tareas.

6. EJECUCIÓN DE LA ITERACIÓN

El equipo realiza una reunión de sincronización, la cual implica que el equipo se mantenga una reunión de trabajo durante 15 minutos, en los cuales exponen sus dudas facilitando la transferencia de información y la colaboración de los miembros del equipo para resolver las inquietudes y promover la productividad. Cada miembro inspecciona el trabajo de resto está ejecutando (dependiendo de las tareas asignadas), y con ello buscar obstáculos que impidan el cumplimiento de los objetivos planteados, estas reuniones buscan contestar las siguientes preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

El facilitador que es el Scrum Master se encarga de que el equipo pueda cumplir con su compromiso y de que no se disminuya la productividad es así que elimina los obstáculos y protege al equipo de interrupciones externas.

7. INSPECCIÓN Y ADAPTACIÓN

El último día de cada iteración se realiza la reunión de revisión de la iteración:

- **Demostración (máximo 4 horas)**

Es una reunión donde el equipo presenta al cliente los requerimientos completados en la iteración; cada reunión de entrega de la iteración es un porcentaje adicional al cumplimiento de la totalidad del proyecto.

Cada entrega e iteración se realizar las pruebas y en caso de existir cambios se hace una reestructuración del proyecto.

8. RETROSPECTIVA (MÁXIMO 4 HORAS)

El equipo analiza cual ha sido su manera de trabajar y cuáles son los productos que podrían impedir el progreso del proyecto de manera adecuada. El facilitador se encarga de ir eliminando los obstáculos encontrados.

ANEXOS 2 – CONTEXTUALIZACIÓN DEL ESTÁNDAR ISO/IEC 29110

Las pequeñas organizaciones (OPs), ya tienen la capacidad para competir, adaptarse y sobrevivir dependiendo del software. Las organizaciones automotrices, alimenticias y educativas dependen mucho de los paquetes de software que manejan y al tener un giro de negocio distinto a lo que es desarrollo de software su área de tecnología es reducida; es por ello que este estándar puede ser aplicado a la gestión del proceso de desarrollo de software.

La industria del software reconoce el valor que tienen las pequeñas organizaciones y Muy Pequeñas Organizaciones (MPO) en la producción de productos y servicios valorables, las pequeñas organizaciones tienen un impacto en la elaboración del producto de software, como se puede observar en la tabla de “**Definición de Pequeña Organización**” nos menciona que el 99.8% de empresas son micro, pequeñas y medianas cumpliendo con una de las características para ser Pequeña organización que es tener hasta 25 empleados.

Tipo de Empresa	Número de empleados	Ventas Anuales(EUR)	Número de Empresas (% del Total)	Número de Empresas
Microempresas	01-sep	≤ 2 millones	92,2 %	19968000
Pequeñas empresas	oct-49	≤ 10 millones	6,5 %	1358000
Medianas empresas	50 –249	≤ 50 millones	1,1 %	228000
Total, micro, Pequeñas y Medianas	87100000		99,8 %	21544 000
Grandes empresas	> 250	> 50 millones		
Total grandes empresas	42900 000		0,2 %	43000

Tabla 39: Definición de pequeña organización

Fuente: Laporte, Claude. (2016). ISO/IEC 29110: Normas y guías de ingeniería de software y sistemas para entidades muy pequeñas. . 10.13140/RG.2.1.2304.9360.

Elaborado por: <https://profs.etsmtl.ca/claporte/english/VSE/>

El estándar ISO/IEC 29110 es un conjunto de normas e informes técnicos que se han desarrollado para entidades muy pequeñas (pequeñas organizaciones). Las cuales se reconoce por desarrollar y mantener software que se utiliza en empresas o que puede ser usado como componente de un gran proyecto de software.

La ISO /IEC 29110 ha sido desarrollada para mejorar la calidad del producto y/o servicio software, también para mejorar el desempeño de las pequeñas organizaciones, el estándar no busca excluir el uso de los ciclos de vida como son casacada, iterativo, incremental, evolutivo o ágil, el estándar cuenta con 5 partes que son:

- **La ISO/IEC 29110 Part 1.** Overview (Parte 1 Descripción general) dirigido a MPO, evaluadores, productores de estándares, vendedores de herramientas y vendedores de metodologías.
- **La ISO/IEC 29110 Part 2.** Framework and taxonomy (Parte 2 Marco de trabajo y taxonomía) dirigido a productores de estándares, vendedores de herramientas y vendedores de metodologías. No está dirigido para la MPO.
- **La ISO/IEC 29110 Part 3** Assessment guide (Parte 3 Guía de evaluación) dirigido a evaluadores y MPO.
- **La ISO/IEC 29110 Part 4** Profile specifications (Parte 4 Especificaciones de perfiles) dirigido a productores de estándares, vendedores de herramientas y vendedores de metodologías. No está dirigido para la MPO.
- La **ISO/IEC 29110 Part 5** Management and engineering guide (Parte 5 Guía de gestión e ingeniería) dirigido a MPO.

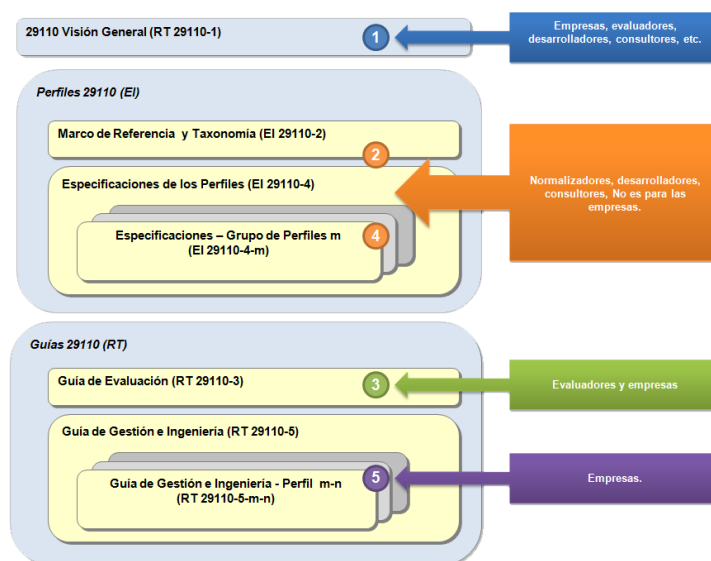


Figura 34: Estructura del estándar ISO /IEC 29110

Fuente: <http://e-processmexico.com/consultoria/iso-29110/>

Elaborado por: <http://e-processmexico.com/consultoria/iso-29110/>

ROLES UTILIZADOS EN LA ISO/IEC 29110

El estándar se fundamenta en dos procesos que son Gestión de proyecto (GP) y la Implementación del software, cada uno de estos procesos están compuestos de actividades, roles y documentos que deben ser contemplados para la ejecución de este.

Rol	Competencias
Ciente	<ul style="list-style-type: none"> • Conoce de los procesos del cliente y tiene la capacidad de proporcionar los requisitos del cliente. • Tiene la facultad de aprobar los requisitos y sus cambios
Líder del proyecto	<ul style="list-style-type: none"> • Capacidad de liderar un equipo • Experiencia en planificación, gestión personal y delegación de funciones • Experiencia en el desarrollo de software
Equipo	<ul style="list-style-type: none"> • Conocimiento y experiencia de acuerdo con sus funciones en el proyecto • Conocimiento de los estándares de programación utilizados por la pequeña empresa y las normativas que utiliza el cliente
Analista	<ul style="list-style-type: none"> • Experiencia en levantamiento de requerimientos • Diseñador de interfaces de usuario y criterios de calidad • Conocimientos de las técnicas de revisión • Experiencia en el desarrollo de software y mantenimiento. • Conocimiento del giro del negocio
Desarrollador	<ul style="list-style-type: none"> • Conocimiento y experiencia en los componentes de software y diseño de las arquitecturas • Conocimiento en técnicas de revisión • Conocimientos en técnicas de pruebas de caja negra y caja blanca. • Conocimiento de modelos y arquitecturas de desarrollo de software • Experiencia en el desarrollo de software y mantenimiento.
Programador	<ul style="list-style-type: none"> • Conocimiento y/u experiencia en la programación, integración y pruebas unitarias • Experiencia en el desarrollo y mantenimiento de software

Tabla 40: Roles del estándar ISO/IEC 29110 (Pasini, Esponda, Boracchia, & Pesado, *Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110*, 2013)

Fuente: (Pasini, Esponda, Boracchia, & Pesado, *Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110*, 2013)

Elaborado por: El Autor.

PERFILES DE LA ISO/IEC 29110

Los perfiles definen el empaquetado de referencias y/o partes de otros documentos de manera formal, con el objetivo de adaptarlos a las necesidades y características de las pequeñas organizaciones, los perfiles deben contener dos tipos de documento formales:

- **Marco de Referencias y Taxonómica**

Especifica los elementos comunes de todos los perfiles (estructura, conformidad, evaluación) e introduce la taxonomía (catálogo) de los perfiles ISO/IEC 29110.

- **Especificación del perfil**

Proporciona la composición definitiva de un perfil, los enlaces normativos al subconjunto normativo de estándares usados en el perfil, y los enlaces informativos (referencias) a documentos de “entrada”. Para cada perfil existe un documento de este tipo.

Un ejemplo de una especificación de perfil es el documento EI 29110-4-1 (Especificaciones de los Perfiles – Perfil Básico). Su objetivo es definir una guía de gestión de proyectos y desarrollo de software, adaptada a las necesidades de las Pequeñas Organizaciones, para un subconjunto de procesos de ISO/IEC 12207.

CARACTERÍSTICAS DE LA ISO/IEC 29110

Las características del estándar son:

- 1) Modelo internacional basado en estudios realizados en los estándares (NMX-I-059-NYCE-2011, MoProSoft, ISO/IEC 12207, ISO/IEC 15289, ISO/IEC 15504, entre otras).
- 2) Modelo Internacional basado en una combinación de estándares
- 3) Es reconocido Internacionalmente.
- 4) Es específico para el desarrollo y mantenimiento de software.
- 5) Dirigido a Pequeñas Entidades de entre 1 a 25 personas.
- 6) Orientado a Proyectos tanto “Internos” como “Externos.”

- 7) Se obtiene un Certificado de Conformidad con vigencia de 3 años y vigilancias anuales.

BENEFICIOS DEL ESTÁNDAR ISO/IEC 29110

- Mejora la calidad del software producido por la organización que adopta el modelo.
- Eleva la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad.
- Las empresas que tienen Implantada la Norma NMX-I-059-NYCE-2011 Nivel 2 “Administrado” pueden migrar fácilmente a la Norma ISO/IEC 29110 “Perfil Básico”.
- Las empresas que lo implementan destacan sobre sus competidores, en beneficio de sus clientes.
- Aparecen en la Lista Nacional de Empresas Certificadas, que sirve como una referencia oficial para clientes, autoridades y competidores.

DESCRIPCIÓN PERFIL DEL ESTÁNDAR ISO/IEC 29110

A continuación, vamos a describir brevemente cada uno de los perfiles existentes para el estándar

1. ISO/IEC 29110 -1

Define los términos de negocio comunes al conjunto de Documentos de perfiles de la Pequeña Organización, el cual introduce conceptos de procesos, ciclo de vida de software y estandarización; así mismo introduce las características que debe tener la organización para la implementación del estándar.

9. ISO/IEC 29110 -2

Introduce conceptos para el perfil estandarizado de la Ingeniería de Software para las Pequeñas Organizaciones, además de reglas comunes para el conjunto de documentos de perfiles de PO. Establece la lógica detrás de la definición y aplicación.

10. ISO/IEC 29110 -3

Determina la guía de evaluación del proceso y determina los requerimientos mínimos que necesita cada perfil definidos por las pequeñas organizaciones; además contienen información útil para desarrolladores de métodos y herramientas de evaluación; esta parte está dirigido para personas que están estrechamente relacionadas con el proceso de evaluación.

11. ISO/IEC 29110 -4-1

Contiene todas las especificaciones del grupo de perfil genéricos. El grupo es aplicable a las pequeñas organizaciones que no desarrollan productos de software críticos. Los perfiles de las pequeñas organizaciones se aplican y están dirigidos a autores o proveedores de guías y herramientas u algún tipo de material que aporta a la utilización del estándar.

12. ISO/IEC 29110 -5-1-1

Es un perfil inicial, creado para pequeñas organizaciones que tienen hasta 3 años de funcionamiento inicial y que sus proyectos tengan una duración de desarrollo inferior a los 6 meses – persona.

13. ISO/IEC 29110 -5-1-2

Perfil Básico, es el primer documento publicado de esta parte que provee una guía de implementación sobre Administración e Ingeniería para del Grupo de Perfiles Genérico descrito en ISO/IEC IS 29110 Parte 4-1. El perfil Básico describe el desarrollo de software de una sola aplicación por un solo equipo de proyecto sin riesgos o factores situacionales especiales.

PROCESOS DEL PERFIL BÁSICO

El perfil base se encuentra constituido por dos procesos: Administración del Proyecto y el proceso de Implementación de Software los cuales tienen el siguiente propósito:

- a. Proporcionar al proceso de administración de proyectos establecer y llevar a cabo de manera sistemática las tareas, actividades del proyecto de implementación que permita el cumplimiento de objetivos planteados en parámetros de tiempo, calidad y eficiencia.
- b. El propósito del proceso de implementación en la aplicación sistemática de las actividades de análisis, diseño, construcción, integración y pruebas del producto de software de un nuevo producto o de algún tipo de mantenimiento.

Los procesos definidos para cada perfil básico son:

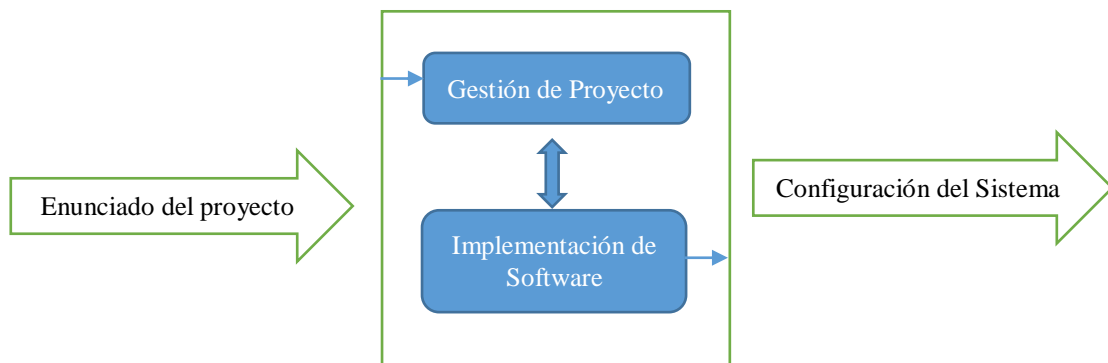


Figura 35: Proceso del proceso del perfil básico

Fuente: (Laporte, Séguin, Villas Boas, & Buasung, 2013)

Elaborado por: El Autor

Ambos procesos se encuentran interrelacionados y pueden trabajar en paralelo, se inicia con la planificación del proyecto originando el plan del proyecto que se utilizara como referencia para monitorear los avances. Durante la ejecución del proyecto se ejecuta las actividades de la implementación del software, una vez que el producto se encuentra probado y validado se realiza la entrega del producto y cierre del proyecto.

GESTIÓN DE PROYECTO

El proceso de gestión de proyecto se define de la siguiente manera:

“El objetivo del proceso de Gestión de Proyectos es establecer y llevar a cabo de manera sistemática las tareas del proyecto de implementación de software, lo que permite cumplir con los objetivos del proyecto en la calidad, tiempo y costo esperados” (Hernadéz, Gonzalez, & Laporte, 2012).

A continuación, se presenta una tabla los objetivos del proceso Gestión de Proyecto

- 1) **Plan de Proyecto:** El plan de proyecto se elabora de acuerdo al enunciado de trabajo, Aquí se define, acuerda y se estima el alcance del proyecto.
- 2) **Avance del Proyecto:** El alcance del proyecto define el control y evaluación del proyecto, identificando los avances y generando análisis y toma de decisiones.
- 3) **Solicitudes de Cambio:** Las solicitudes de cambios son parte del proceso de análisis de requerimientos, donde se evalúan los cambios solicitados en cuanto a tiempo, costo y esfuerzo.
- 4) **Reunión de revisión con el cliente:** Son reuniones que permiten comunicare al cliente los avances, novedades del proyecto, además de poder realizar la toma de decisiones e instancias una comunicación efectiva entre las partes.
- 5) **Riesgos:** Se realiza la gestión y toma de acciones de mitigación queridas para minimizar los riesgos del proyecto.
- 6) **Control de versiones:** Se manejan las versiones de los productos generados.
- 7) **Calidad:** Donde se busca garantizar el cumplimiento del plan de proyecto y el cumplimiento de la especificación de los requisitos. Este es uno de los objetivos que tiene mucho impacto entre las microempresas de Ecuador.

Este proceso tiene roles que participan en el proceso Gestión de Proyecto que son:

- **Cliente:** es quien tiene la necesidad
- **Gestor de Proyecto:** encargado de cumplir el proyecto
- **Líder Técnico:** responsable de proponer alternativas de solución eficientes a las necesidades del cliente.
- **Equipo de Trabajo:** analistas, programadores, arquitectos y tester.

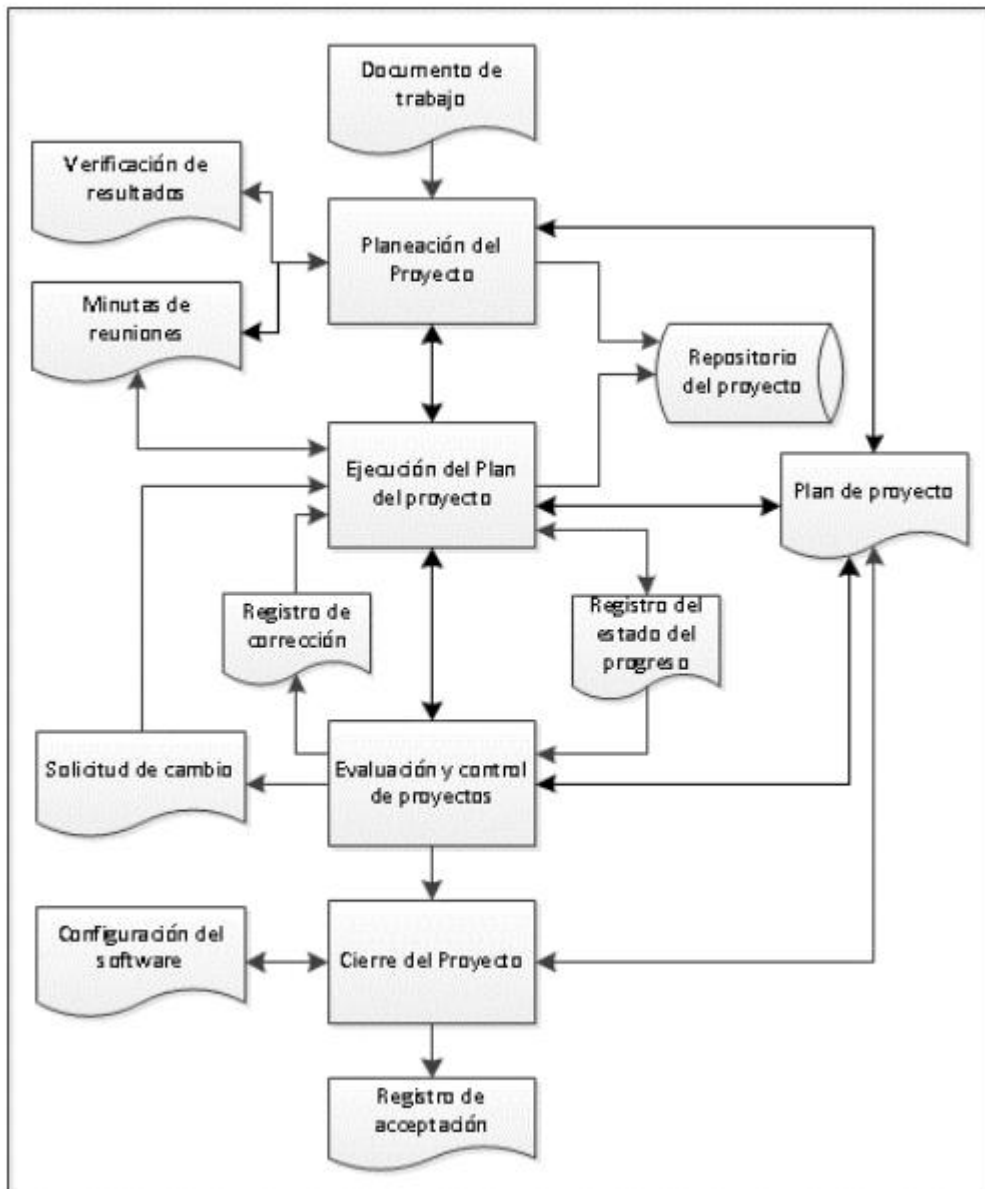


Figura 36: Diagrama del proceso de gestión de proyectos (ISO / IEC 29110)

Fuente: Deployment Package Project Management Entry Profile (September 21, 2012).

Elaborado por: Deployment Package Project Management Entry Profile (September 21, 2012).

IMPLEMENTACIÓN DE SOFTWARE

El proceso de implementación de software según la documentación de Entry profile – DP-Proyect Management_03 (Hernadéz, Gonzalez, & Laporte, 2012), lo define como:

“El propósito del proceso de implementación de software es el desempeño sistemático del análisis, identificación de componentes de software, construcción, integración y pruebas y actividades de entrega de productos para productos de software nuevos o modificados de acuerdo con los requisitos especificados.”

Este proceso define 7 objetivos, que son:

1) Tareas

Las tareas se realizan cumpliendo y ejecutando el plan del proyecto.

2) Definición de los requerimientos de software

Todos los requerimientos son definidos, aprobados y comunicados a las partes interesadas, este es un sub proceso de la definición de requisitos.

3) Arquitectura y diseño detallado

Aquí se describen los Componentes de Software y sus interfaces internas y externas. La consistencia y trazabilidad de los requisitos de Software son establecidos.

4) Componentes de software

Las pruebas unitarias son definidas y ejecutadas para verificar la consistencia de los requisitos y el diseño. La trazabilidad de los requisitos y el diseño son establecidas.

5) Integración de componentes

Subproceso de la integración de software, se genera la solución y se ejecuta de manera integral de cada uno de los componentes donde el producto es verificado utilizando el plan de pruebas definido y aprobado.

6) Configuración del software

Es un subproceso de la gestión de la configuración, se prepara la entrega oficial del producto al cliente final donde se considera la generación de manuales de usuario.

7) Verificación y validación

Subproceso de verificación del software, donde se busca asegurar la calidad del producto en cada uno de los entregables validando los requisitos asociados.

Este proceso define ciertos roles que se deben cumplir como son:

- Cliente: es quien tiene la necesidad.
- Analista: es quien realiza el levantamiento y entendimiento de la solución
- Programadores: son quienes codifican los requerimientos.
- Gestor de proyecto: responsable del cumplimiento del proyecto
- Líder técnico: responsable de proporcionar la solución de la necesidad
- Equipo de trabajo: analistas, programadores, arquitectos, etc.

A continuación, en la siguiente figura se muestra el diagrama de flujo

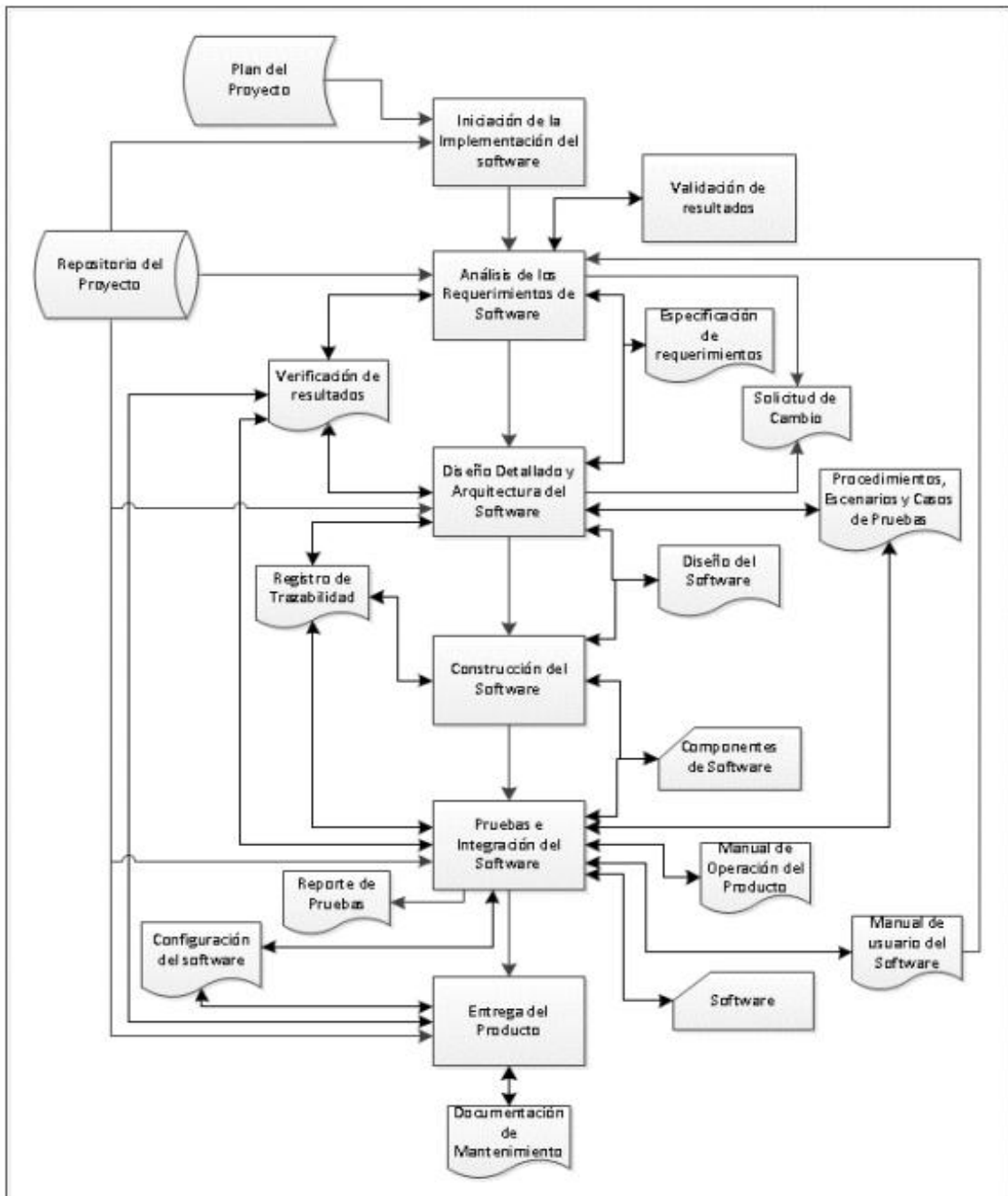


Figura 37:Diagrama del proceso de implementación de Software (ISO / IEC 29110)

Fuente: Deployment Package Project Management Entry Profile (September 21, 2012).

Elaborado por: Deployment Package Project Management Entry Profile (September 21, 2012).

PRODUCTOS O ARTEFACTOS

El estándar en cada uno de los procesos se realiza el levantamiento de artefactos que son creados y llenados de acuerdo al proyecto, a continuación, se detallan los artefactos que son considerador obligatorios en el proceso de administración:

Documentos	Descripción
Declaración de trabajo	Descripción del producto, contiene: Propósito. Requerimientos generales. Alcance. Objetivos. Entregables
Configuración del software	Identificación de conjunto de productos de software que se deben mantener actualizados, contiene: Especificación de requerimientos, Diseño de software, Registro de trazabilidad, Software, Componentes, Casos de prueba, Reportes de pruebas, Manual de usuario, Documentación de mantenimiento.
Solicitud de cambio	Documentación que identifica las solicitudes de cambios, contiene: Propósito, estado de la solicitud, solicitante, impacto.
Plan de proyecto	Descripción de cómo el proyecto y sus actividades serán ejecutadas, contiene: Descripción del producto, Propósito, Requerimientos generales, Alcance, Objetivos, Entregables, Tareas, Estimación de tiempo/costo/duración, Composición del equipo de trabajo, Riesgos.
Registro de aceptación	Documento que establece la aceptación de los entregables por el cliente, contiene: Registro de recepción de entregable, Fecha de recepción, Criterios de aceptación.
Actas de reuniones	Registro de acuerdos establecidos con el cliente y/o equipo de trabajo, contiene: Propósito de la reunión, asistentes, fecha, logros, cuestiones planteadas.

Tabla 41: Documentos de generación obligatoria (*Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110, 2013*)b

Fuente: (Pasini, Esponda, Boracchia, & Pesado, Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110, 2013)b

Elaborado por: El Autor

Adicionalmente a los procesos, subprocesos, actividades, tareas, roles y entregables que se presentan en el perfil básico de la ISO/IEC 29110, se ha trabajado en herramientas que faciliten la implementación del estándar en las pequeñas Organizaciones, lo que ha dado lugar al desarrollo de los denominados “Paquetes de Despliegue” (DP), documentos que definen las directrices y explican con un mayor detalle los procesos, tareas, actividades y roles en los perfiles de la ISO/IEC 29110. El Profesor Claude Laporte (miembro del grupo y precursor del estándar) en su página web (Laporte C. , Public Site of the ISO Working Group Mandated to Develop ISO/IEC 29110 Standards and Guides for Very Small Entities involved in the Development or Maintenance of Systems and/or Software, 2014) tiene publicado los DP que se han construido hasta la fecha, basándose en el cubrimiento de los subprocesos que soportan el estándar para el perfil básico (Valdebenito Espinosa, Guía Metodológica para la implementación del estándar ISO/IEC 29110 en Pymes TICs de Colombia, 2016).



Figura 38: Conjunto de paquetes de implementación para apoyar el perfil básico de ingeniería de sistemas

Fuente: Fuente (Laporte C. , Public Site of the ISO Working Group Mandated to Develop ISO/IEC 29110 Standards and Guides for Very Small Entities involved in the Development or Maintenance of Systems and/or Software, 2014).

Estos paquetes según opinión del profesor Claude Laporte, permiten a una pequeña organización aplicar el contenido del estándar con mayor facilidad y utilizar solo lo que la organización necesite:

- Descripción del proceso
- Actividades
- Tareas
- Roles y productos
- Plantillas
- Lista de verificación
- Ejemplos
- Referencias y trazabilidad de las normas y modelos
- Lista de herramientas.

Adicionalmente se dispone de paquetes oficiales, publicados en el sitio oficial de países como: Canadá, Brasil, Colombia y Perú; en este trabajo se utiliza la traducción realizada por la Universidad Peruana de Ciencias Aplicadas (UPC), la cual se encuentra publicada en el sitio oficial del estándar.

PAQUETE DE DESPLIEGUE DEL PERFIL BÁSICO DE LA NORMA ISO/IEC20110

Los paquetes de despliegue son el resultado de una profunda investigación realizada en conjunto, liderada por el profesor Claude Y. Laporte de la universidad de Quebec – École de technologie supérieure es el principal exponente sobre la realización de la norma, que se encuentra publicada en el URL <http://profs.etsmtl.ca/claporte/english/VSE/>, con ayuda de los Miembros del Grupo de Trabajo 24 encargados de elaborar normas y guías para los VSE que participan en el desarrollo de sistemas y / o programas informáticos.

Universidades como Quebec han proporcionado Paquetes de Despliegue, Japón es parte de los países que ha colaborado, ellos tienen publicado un sitio web; países como Finlandia, Colombia, Perú, Argentina, Brasil, México, Bélgica, Francia.

En Ecuador, los paquetes de despliegue aún no se encuentran en crecimiento, existe estudios realizados en la Escuela Universidad Politécnica del Ejército – ESPE. Para la implementación que se realizara en la Universidad Técnica del Norte se tomara como base el Paquete de Despliegue PD B-52, el cual fue traducido por la Universidad Peruana de Ciencias Aplicadas; este paquete está avalado por el profesor Laporte y mencionado en el sitio oficial de publicación como una de las versiones disponibles para utilizar en una implementación.

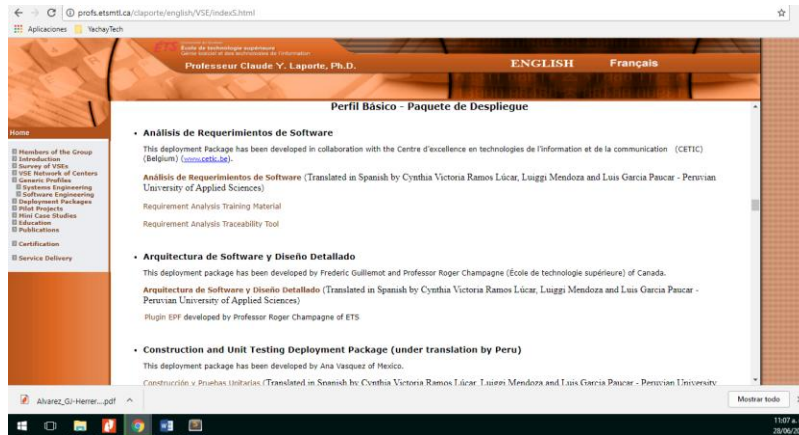


Figura 39: Sitio de descarga del Paquete de Despliegue

La descripción de cada uno de los paquetes se adjunta como anexo

- 1) Análisis de requerimientos
- 2) Arquitectura de software y diseño
- 3) Construcción y pruebas
- 4) Integración y pruebas
- 5) Entrega de producto
- 6) Control de versiones
- 7) Gestión de Proyecto
- 8) Validación y verificación

GENERACIÓN DE ARTEFACTOS DE ESTÁNDAR

La generación de los artefactos para facilitar la implementación del estándar se desarrolló tomando en cuenta los paquetes de despliegue que el estándar nos menciona los cuales son:

PAQUETES DE DESPLIEGUE	DESCRIPCIÓN
Gestión de Proyectos	<p>Este paquete de despliegue describe el proceso de gestión de un proyecto de software, establece el proceso que se debe tomar en cuenta e indica los controles de gestionar un proyecto. Además, nos indica los artefactos que son necesarios para gestionar un proyecto como son Plan de proyecto, reporte de avance y solicitud de cambios, cabe indicar que incluye las plantilladle estos tres documentos.</p> <p>Los roles definidos en el paquete son:</p> <ul style="list-style-type: none"> - Gestor de Proyecto - Analista - Desarrollador - Cliente
Análisis de requerimientos	<p>El propósito es ayudar a la gestión, levantamiento y seguimiento de los requerimientos del proyecto de software, el proceso incluye la producción y</p>

	<p>mantenimiento de las especificaciones de requerimientos de software, además de incluir una base para realizar la estimación de los costos.</p> <p>Los documentos que genera este paquete es la especificación de requerimientos que es basada en el estándar ISO830, los roles que intervienen en el paquete son: Analista, Gestor de Proyecto, Cliente</p>
Arquitectura y diseño de software	<p>El propósito del paquete es facilitar a las pequeñas empresas de tener una buena arquitectura, tomando en cuenta los componentes de Software, Hardware, Interfaces para establecer un entorno de trabajo, este documento se encuentra basado en el estándar ISO / IEC 24765</p> <p>El paquete de despliegue utiliza los siguientes roles:</p> <ul style="list-style-type: none"> - Analista - Diseño - Líder Técnico <p>Los productos que apoya el paquete para su generación son:</p> <ul style="list-style-type: none"> - Diseño de Software - Especificación de Requerimientos - Registro de Trazabilidad - Caso de pruebas y procedimiento - Configuración de Software - Resultados de Validación
Construcción y pruebas	<p>El propósito de este paquete es planificar las actividades de construcción del proyecto, además de apoyar a la definición de las pruebas de caja blanca y de la caja negra. Los productos que se obtiene de este paquete son: el plan de proyecto de software y el rol que interviene es el Líder Técnico.</p>
Integración y pruebas	<p>El propósito del paquete es documentar las pruebas y el objetivo de las pruebas de integración es verificar que los componentes y subsistemas interactúen correctamente.</p> <p>El rol utilizado es el diseñador, analista, líder técnico, cliente y tester y los productos obtenidos son Reporte de pruebas, matriz de requerimientos de pruebas.</p>
Verificación y validación	<p>El propósito del paquete es planificar la verificación y validación del producto de software con la finalidad de promover los atributos de calidad; ayuda a verificar el plan de proyecto, entregables y artefactos generados es.</p> <p>Los roles que intervienen en este paquete es Gestor de Proyecto, Líder técnico y cliente</p>
Entrega del producto	<p>En este paquete se establecen los lineamientos a tener en cuenta para la entrega del software</p> <p>Los roles que intervienen son: Gestor de Proyecto, Cliente y los artefactos generados son: Enunciado del trabajo, plan de proyecto, formulario de instrucción de entrega y formulario de acta de aceptación.</p>

Control de Versiones	<p>Tener un control de las versiones de los productos de software y de los artefactos es muy importante para tener resguardado la información, para la gestión de solicitudes de cambios y sobre todo conocer cuál ha sido la evolución del producto de software y los documentos.</p> <p>Los roles que intervienen son: Líder Técnico y Gestor de Proyecto y se genera las estrategias de control de versiones y el repositorio del proyecto.</p>
----------------------	--

Tabla 42: Paquetes de despliegue de la ISO 29110

Fuente: (Laporte, Séguin, Villas Boas, & Buasung, 2013)

Elaborado por: El autor

ANEXO 3 NORMAS DE CONTROL INTERNO

410 TECNOLOGIA DE LA INFORMACION

410-01 Organización informática

Las entidades y organismos del sector público deben estar acopladas en un marco de trabajo para procesos de tecnología de información que aseguren la transparencia y el control, así como el involucramiento de la alta dirección, por lo que las actividades y procesos de tecnología de información de la organización deben estar bajo la responsabilidad de una unidad que se encargue de regular y estandarizar los temas tecnológicos a nivel institucional.

La Unidad de Tecnología de Información, estará posicionada dentro de la estructura organizacional de la entidad en un nivel que le permita efectuar las actividades de asesoría y apoyo a la alta dirección y unidades usuarias; así como participar en la toma de decisiones de la organización y generar cambios de mejora tecnológica. Además debe garantizar su independencia respecto de las áreas usuarias y asegurar la cobertura de servicios a todas las unidades de la entidad u organismo.

Las entidades u organismos del sector público, establecerán una estructura organizacional de tecnología de información que refleje las necesidades institucionales, la cual debe ser revisada de forma periódica para ajustar las estrategias internas que permitan satisfacer los objetivos planteados y soporten los avances tecnológicos. Bajo este esquema se dispondrá como mínimo de áreas que cubran proyectos tecnológicos, infraestructura tecnológica y soporte interno y externo de ser el caso, considerando el tamaño de la entidad y de la unidad de tecnología.

410-02 Segregación de funciones

Las funciones y responsabilidades del personal de tecnología de información y de los usuarios de los sistemas de información serán claramente definidas y formalmente comunicadas para permitir que los roles y responsabilidades asignados se ejerzan con suficiente autoridad y respaldo.

La asignación de funciones y sus respectivas responsabilidades garantizarán una

.....

410-03 Plan informático estratégico de tecnología

La Unidad de Tecnología de la Información elaborará e implementará un plan informático estratégico para administrar y dirigir todos los recursos tecnológicos, el mismo que estará alineado con el plan estratégico institucional y éste con el Plan Nacional de Desarrollo y las políticas públicas de gobierno.

El plan informático estratégico tendrá un nivel de detalle suficiente para permitir la definición de planes operativos de tecnología de Información y especificará como ésta contribuirá a los objetivos estratégicos de la organización; incluirá un análisis de la situación actual y las propuestas de mejora con la participación de todas las unidades de la organización, se considerará la estructura interna, procesos, infraestructura, comunicaciones, aplicaciones y servicios a brindar, así como la definición de estrategias, riesgos, cronogramas, presupuesto de la inversión y operativo, fuentes de financiamiento y los requerimientos legales y regulatorios de ser necesario.

La Unidad de Tecnología de Información elaborará planes operativos de tecnología de la información alineados con el plan estratégico informático y los objetivos estratégicos de la institución, estos planes incluirán los portafolios de proyectos y de servicios, la arquitectura y dirección tecnológicas, las estrategias de migración, los aspectos de contingencia de los componentes de la infraestructura y consideraciones relacionadas con la incorporación de nuevas tecnologías de información vigentes a fin de evitar la obsolescencia. Dichos planes asegurarán que se asignen los recursos apropiados de la función de servicios de tecnología de información a base de lo establecido en su plan estratégico.

El plan estratégico y los planes operativos de tecnología de información, así como el presupuesto asociado a éstos serán analizados y aprobados por la máxima autoridad de la organización e incorporados al presupuesto anual de la organización; se actualizarán de manera permanente, además de ser monitoreados y evaluados en forma trimestral para determinar su grado de ejecución y tomar las medidas necesarias en caso de desviaciones.

410-04 Políticas y procedimientos

La máxima autoridad de la entidad aprobará las políticas y procedimientos que permitan

410-06 Administración de proyectos tecnológicos

La Unidad de Tecnología de Información definirá mecanismos que faciliten la administración de todos los proyectos informáticos que ejecuten las diferentes áreas que conformen dicha unidad. Los aspectos a considerar son:

1. Descripción de la naturaleza, objetivos y alcance del proyecto, su relación con otros proyectos institucionales, sobre la base del compromiso, participación y aceptación de los usuarios interesados.
2. Cronograma de actividades que facilite la ejecución y monitoreo del proyecto que incluirá el talento humano (responsables), tecnológicos y financieros además de los planes de pruebas y de capacitación correspondientes.
3. La formulación de los proyectos considerará el Costo Total de Propiedad CTP; que incluya no sólo el costo de la compra, sino los costos directos e indirectos, los beneficios relacionados con la compra de equipos o programas informáticos, aspectos del uso y mantenimiento, formación para el personal de soporte y usuarios, así como el costo de operación y de los equipos o trabajos de consultoría necesarios.
4. Para asegurar la ejecución del proyecto se definirá una estructura en la que se nombre un servidor responsable con capacidad de decisión y autoridad y administradores o líderes funcionales y tecnológicos con la descripción de sus funciones y responsabilidades.
5. Se cubrirá, como mínimo las etapas de: inicio, planeación, ejecución, control, monitoreo y cierre de proyectos, así como los entregables, aprobaciones y compromisos formales mediante el uso de actas o documentos electrónicos legalizados.
6. El inicio de las etapas importantes del proyecto será aprobado de manera formal y comunicado a todos los interesados.
7. Se incorporará el análisis de riesgos. Los riesgos identificados serán permanentemente evaluados para retroalimentar el desarrollo del proyecto, además de ser registrados y considerados para la planificación de proyectos futuros.
8. Se deberá monitorear y ejercer el control permanente de los avances del proyecto.
9. Se establecerá un plan de control de cambios y un plan de aseguramiento de calidad que será aprobado por las partes interesadas.
10. El proceso de cierre incluirá la aceptación formal y pruebas que certifiquen la calidad y el cumplimiento de los objetivos planteados junto con los beneficios obtenidos.

410-07 Desarrollo y adquisición de software aplicativo

La Unidad de Tecnología de Información regulará los procesos de desarrollo y adquisición de software aplicativo con lineamientos, metodologías y procedimientos. Los aspectos a considerar son:

1. La adquisición de software o soluciones tecnológicas se realizarán sobre la base del portafolio de proyectos y servicios priorizados en los planes estratégico y operativo

Condiciones de entrada	Sustento
Existe un contrato del proyecto con el Enunciado de Trabajo	Dirección de Desarrollo Tecnológico e Informático cuenta con un formato para establecer un contrato que en este caso se denomina acta de Inicio del proyecto donde se describe el alcance del proyecto o del sistema, costos, herramientas, técnicas, cronograma.
La factibilidad del contrato, que incluye costos, cronograma y aspectos técnicos, fue realizada antes de iniciar el proyecto.	
Los recursos humanos, incluyendo al gestor de proyecto, están asignados y entrenados.	
Los bienes, servicios e infraestructura están disponibles.	

Tabla 43: Revisión de Condiciones de entrada

Fuente: Universidad Técnica del Norte

Elaborado por: El Autor

ANEXO 4: ENUNCIADO DEL TRABAJO - ÁGIL



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ENUNCIADO DEL TRABAJO - ÁGIL

ENUNCIADO DEL TRABAJO - ÁGIL

[Nombre de Proyecto]

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madrueno	Creación de Plantilla		05/11/2017
Versión 1.0	Ing. Juan Carlos García	Revisión	Director de la Dirección de Desarrollo Tecnológico e Informático	06/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ENUNCIADO DEL TRABAJO - AGÍL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

Este documento tiene como objetivo realizar un levantamiento de información del proyecto a ser ejecutado, indicando los entregables dependiendo de los requerimientos de alto nivel mencionados por el cliente.

1.2. GLOSARIO DE TERMINOS

- **Identificador:** Código único generado para un registro
- **Entregables:** Son los documentos físicos y/o electrónicos que se elaboran el desarrollo del proyecto.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Información General

[Realizar una breve descripción de la organización, indicando el nombre, ubicación, así como también el nombre del proyecto y propósito]

2.1. Definir listado de Necesidades del Cliente

[Se lista de los requisitos del cliente en alto nivel, esto lo solicita el cliente].

3. Alcance del proyecto

[Se debe detallar el alcance del proyecto. Expresar claramente todo lo que incluye y las restricciones]

3.1. Proyecto incluye

[Se detalla las características que el proyecto debe cumplir]

3.2. Proyecto excluye

[Las condiciones que el proyecto excluirá]

4. Objetivos del proyecto

[Listar las metas y objetivos del proyecto, con cada objetivo se explica la finalidad que necesitan los stakeholders]



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ENUNCIADO DEL TRABAJO - AGIL

5. Requisitos Generales del Proyecto

[Una lista de requerimientos de alto nivel que solicita el cliente a satisfacer a través de la ejecución de proyecto, recomienda utilizar lenguaje de negocio para que el cliente se sienta cómodo.]

Id ¹	Descripción ²	Estimación ³	Prioridad ⁴
<Identificador>	<descripción del requisito>	<Valor del ítem en puntos>	<Importancia del ítem para el cliente>

Para llenar la tabla, tomar en cuenta el anexo 001 denominado Requisitos Generales del Proyecto

6. Entregables

[Es un listado de elementos como documentos, código fuente a ser entregados, la lista debe contener fechas de entrega, roles y nombres de las personas que reciben]

Identificador	Entregable	Tipo	Fecha estimada	Receptor	Observación
ENT-DDTI-001	Enunciado del trabajo	Artefactos			
ENT-DDTI-002	Plan de Proyecto	Artefactos			
ENT-DDTI-003	Especificación de Requerimientos	Artefactos			
ENT-DDTI-004	Diseño de Arquitectura	Artefactos			
ENT-DDTI-005	Verificación	Artefactos			
ENT-DDTI-006	Validación	Artefactos			
... .. ■					

Tabla 7 Entregables

Guía con el Anexo 002 - Entregables

¹ Identificador

² Descripción del requerimiento

³ Estimación del esfuerzo

⁴ Prioridad de trabajo



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ENUNCIADO DEL TRABAJO - AGÍL

Anexo 001 - Requisitos Generales del Cliente

Para llenar el contenido de la tabla de Requisitos Generales del proyecto se toma en cuenta lo siguiente:

- ✓ **IDENTIFICADOR (Id)**
Cada uno de los requerimientos generales se deben de codificar de manera que podamos dar el seguimiento y su trazabilidad, para la codificación se sugiere el siguiente estandar:
 - REQ-<UNIDAD REQUIRIENTE>-<NUMERACIÓN>Ejemplo: REQ-ACA-001
- ✓ **DESCRIPCIÓN**
En la descripción indicamos la funcionalidad que necesita el cliente, tomando en cuenta la siguiente estructura:
 - < Como rol que utiliza, necesito: funcionalidad para beneficio.>Ejemplo:
Como Analista de Talento Humano necesito: Ser notificado por el sistema con al menos un mes de anticipación los contratos que estén por terminarse.
- ✓ **ESTIMACIÓN**
Es la valoración de cada item en punto, una de las técnicas es el ~~Planning poker~~
- ✓ **PRIORIDAD**
Cada una de los requerimientos o historias de usuario necesita ser priorizada y proporcionar el orden de desarrollo.
Para la estimación se tiene la siguiente tabla:

IDENTIFICADOR	DESCRIPCIÓN
Alto	Cuando el requerimiento necesita ser tratado inmediatamente
Medio	Cuando el requerimiento puede esperar, porque no es tan necesario
Bajo	Cuando el requerimiento puede ser desarrollado con calma

Tabla 12: Opciones de prioridad



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ENUNCIADO DEL TRABAJO - AGÍL

Anexo 002 – Entregables

En la sección de entregables se listan los entregables del proyecto y la fecha estimada de entrega donde la persona que se indica como responsable es la encargada de validar.

Las columnas de la tabla son:

- **IDENTIFICADOR**
Es el código de cada entregables, el cual nos ayudará a identificarlo y darle seguimiento, el id tendrá el siguiente formato:
 - o ENT-<UNIDAD REQUIRENTE>-<SECUENCIAL>Ejemplo: ENT-ACA-001
- **ENTREGABLE**
Se detalla el nombre de los entregables que se realizaran en el proyecto
- **TIPO**
Se debe definir los tipos de entregables que se tiene, que pueden ser:

TIPO	DESCRIPCION
Artefactos	Se conoce como artefactos a la documentación generada en el desarrollo del proyecto como, por ejemplo: <ul style="list-style-type: none">- Manuales de usuario- Gestión de la Configuración- Especificación de Requerimientos- Modelos, diagramas, etc.
Fuentes	Se conoce como fuentes al código fuente del proyecto

Tabla 15: Tipo de entregables

- **FECHA ESTIMADA DE ENTREGA**
Se asigna la fecha estimada de entrega
- **RECEPTOR**
Se menciona la persona responsable de validar el entregable
- **OBSERVACIÓN**
En el caso de existir alguna observación se la indica en la casilla indicada

ANEXO 5: PLAN DE PROYECTO - ÁGIL



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

|

PLAN DE PROYECTO - ÁGIL

[Nombre de Proyecto]

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madruero	Creación de Plantilla	Creador de plantilla	04/11/2017
Versión 1.0	Ing. Juan Carlos Garcia	Director de Desarrollo Tecnológico E Informático	Revisión	07/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

El presente documento tiene como objetivo plantear el proyecto de manera específica y enfocarse a factores como la planificación, levantamiento de requerimientos, gestión de la configuración de manera rápida y accesible para la metodología ágil.

1.2. GLOSARIO DE TERMINOS

- **Identificador:** Código único generado para un registro
- **Sprint:** Termino que se utiliza para nombras a las iteraciones realiza en SCRUM.
- **Riesgos:** Es la posibilidad que un evento ocurra y afecte negativamente a la consecución de los objetivos planteados
- **Entregables:** Son los documentos fisicos y/o electrónicos que se elaboran el desarrollo del proyecto.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Información General

2.1. Visión General del Proyecto

[Una visión general de alto nivel de proyecto que incluye la descripción del producto, el propósito y alcance del proyecto. En la descripción del alcance del proyecto, debería incluirse lo que se incluye y lo que se excluye del proyecto.]

2.2. Objetivo del Proyecto

[Es el fin al que se desea llegar, es decir la meta que se desea cumplir para satisfacer la o las necesidades que tiene el cliente.]

2.3. Requisitos Generales del Proyecto

[Se describe un listado de requisitos de alto nivel que solicita el cliente y se debe satisfacer a través de la ejecución del proyecto]

Id ¹	Descripción ²	Estimación ³	Prioridad ⁴
<Identificador>	<descripción del requisito>	<Valor del ítem en puntos>	<Importancia del ítem para el cliente>

Tabla 1: Requisitos Generales del Proyecto

Para llenar la tabla, tomar en cuenta el anexo 001 denominado Requisitos Generales del Proyecto

¹ Identificador

² Descripción del requerimiento

³ Estimación del esfuerzo

⁴ Prioridad de trabajo



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

3. Responsabilidades del proyecto

[En esta sección se detallan los responsables del proyecto y su información básica para contactos]

Identificador	Nombres	Apellidos	Dirección	Correo Electrónico	RoF ⁵

Tabla 2: Responsabilidades del proyecto

Los roles y la descripción de las funcionalidades de cada uno se detallan en el Anexo 002- Roles.

4. Plan de Entrega

[Es un plan realizado a alto nivel que me indica las expectativas de entrega de cada uno de los sprint o iteraciones del ~~proy~~ect, para la elaboración de este plan se debe tener ~~Product~~ Backlog priorizado, además establecido la duración de cada Sprint para determinar el número de entregas en total]

Fecha estimada de entrega	Sprint	Id Reouerimiento	Observación
<dd/mm/yyyy>			

Tabla 3: Plan de Entrega

5. Gestión de Riesgos

[Se debe elabora un listado de riesgos y su plan o estrategia de mitigación juntamente con los responsables de ejecutar el plan o estrategia de mitigación]

ID	Tipo	Descripción	Plan de Mitigación	Responsables

Tabla 4 Identificación de Riesgos

Revisar en el Anexo 003, Identificación de Riesgos como registrar la información en la tabla del plan de proyecto.

6. Cronograma y presupuesto

[Se detalla un cronograma y un presupuesto]

6.1. Cronograma

[Se indica las actividades, tareas e hitos a ser considerados en el proyecto con las fechas estimadas de entrega]

Hitos ⁶	Descripción	Criterio de Éxito	Fecha Estimada	Observación
HI-<UNIDAD REQUERENTE><SECUENCIAL>				
H01-ACA-001				

Tabla 5: Cronograma

⁵ Los roles que pueden ser indicados son: **Cliente, Líder de Proyecto, Desarrollador y Analista**

⁶ La columna de hitos es cal codificación de cada actividad mencionada.



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

6.2. Presupuesto

[Planificación financiera del proyecto. Detallar los recursos, entrenamiento, viáticos, etc.]

Presupuesto por Hito	Unidades organizacionales				
	REC001	REC002	REC003	REC...	SUBTOTAL
HI-<UNIDAD REQUERENTE> <SECUENCIAL>					
Total					

Tabla 6: Presupuesto

Para el registro del presupuesto usted puede revisar la Guía con el Anexo 004

7. Entregables

[Es un listado de elementos como documentos, código fuente a ser entregados, la lista debe contener fechas de entrega, roles y nombres de las personas que reciben]

Identificador	Entregable	Tipo	Fecha estimada	Receptor	Observación

Tabla 7 Entregables

Guía con el Anexo 005 - Entregables

8. Gestión de la Configuración

8.1. Identificación de elementos de configuración

[Se identifica un listado de elementos que se deben tomar como base y asociado a cada hito del proyecto para incluirlo en la gestión de la configuración Ejem: Especificación de Requerimientos, artefactos entregables, etc.]

8.2. Herramientas, Técnicas y Metodologías

[Listar las herramientas utilizadas para las gestiones de la configuración]

8.3. Acceso a repositorios

[Se debe describir el repositorio a utilizar en el ~~project~~, donde se debe indicar el tipo de repositorio, espacio físico, dispositivos electrónicos, ubicación del repositorio.]

Denominación	Valué
Tipo de repositorio	- Electrónico, - Físico, - Dispositivo electrónico
Descripción del repositorio	
Ubicación del repositorio	
Dirección URL	

Tabla 8 Acceso a repositorios



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

[La dirección de URL me indica en qué lugar del disco se encuentra almacenada la información ejem: C:/repositoio/proyecto1/]

9. Almacenamiento y recuperación de Entregables

9.1. Mecanismo de Almacenamiento y Manipulación de Entregables

Identificador	Artefacto / entregable	Técnica de Almacenamiento	Descripción	Responsable

Tabla 9: Mecanismo de Almacenamiento y Manipulación de Entregables

[Revisar el Anexo 006 – Almacenamiento y recuperación del entregable]

9.2. Mecanismo de Respaldo y Recuperación de Información

9.2.1. Mecanismo de Respaldo

[Se describe el tipo de información, configuración utilizada para el respaldo (control de accesos configuración de software y hardware), además se indica las estrategias de respaldo, frecuencia]

9.2.2. Mecanismo de Recuperación

[Se indica los métodos y estrategias para la recuperación de la información en caso de posibles fallos, estas estrategias deben estar alineadas al plan de recuperación y ~~Backup~~ que disponga la organización]

10. Entrega del Producto

10.1. Requisitos de Entrega

[Validar el ~~check list~~ de entrega del producto, en la columna de validación se debe poner SI en el caso de haber cumplido y NO en caso de no cumplir; se cumple con los prerequisites de entrega si la tabla los entregables tiene que sí.]

ENTREGABLE	VALIDACIÓN	RESPONSABLE	OBSERVACIÓN
Supero las pruebas funcionales			
El sistema se encuentra instalado y operativo funcionalmente			
Cuenta con manuales de usuario			
Cuenta con manuales de configuración			

Tabla 10: Requisitos de Entrega



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

10.2. Elementos que conforman la entrega

[Se identifica una lista de elementos obligatorios que son parte de la entrega del producto]

ENTREGABLE	RECEPTOR	OBSERVACIÓN
Supero las pruebas funcionales		
El sistema se encuentra instalado y operativo funcionalmente		
Cuenta con manuales de usuario		
Cuenta con manuales de configuración		

Tabla 11 Elementos que conforman la entrega

10.3. Pasos de Entrega

[Definición de actividades a seguir para completar la entrega.]

ANEXO 6: ESPECIFICACIÓN DE REQUERIMIENTOS – ÁGIL



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

|

ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

[Nombre de Proyecto]

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madruñero	Creación de Plantilla		06/11/2017
Versión 1.0	Ing. Juan Carlos García	Revisión	Director de la Dirección de Desarrollo Tecnológico e Innovación	07/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

La especificación de requerimientos es un documento formal que se utiliza para dejar escrito de manera formal las funcionalidades, característica, procesos y formas de acceso que deberá tener un producto de software solicitado

1.2. GLOSARIO DE TERMINOS

- **Identificador:** Código único generado para un registro
- **Requerimientos:** Comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado, tomando en cuenta los diversos requisitos de las partes interesadas, que pueden entrar en conflicto entre ellos
- **Requerimientos Funcionales:** Define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas.
- **Entregables:** Son los documentos físicos y/o electrónicos que se elaboran el desarrollo del proyecto.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Información General

2.1. Visión General

[Describir brevemente la visión del proyecto de software final].

2.2. Alcance

[Indicar lo que incluye el sistema y también lo que se excluye].

3. Descripción General

3.1. Funciones de producto

[Listar las funcionalidades que debe tener el software las cuales buscan satisfacer las necesidades de los clientes en condiciones específicas]

IDENTIFICADOR	DESCRIPCIÓN
FUN-<AREA REQUIRENTE> <SECUECNIAL>	<Descripción>
[FUN-ACAD-001]	-----

Tabla 1: Listado de Funciones de producto



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

3.2. Perspectiva del producto

3.2.1. Interfaces del Sistema

[Diseñar un diagrama de las interfaces por las que el sistema se comunicara con otros sistemas ejemplo: Servicios web, JMS, ESB.]

3.2.2. Interface de Hardware

[Listar los distintos dispositivos desde donde se va a poder utilizar el producto de software]

3.2.3. Interface de Comunicación

[Diseñar un diagrama de comunicación que el sistema utilizara para comunicarse con los distintos dispositivos]

4. Requerimientos Específicos

4.1. Requerimientos de interfaces Externas

[Diseñar el diagrama o definir un listado de las interfaces externas con las que se comunica el sistema]

4.2. Características del Producto de Software

[Listar las características no funcionales del software, identificado a que requerimiento funcional se relaciona]

IDENTIFICADOR	DESCRIPCIÓN	¹ IDENTIFICADOR FUNCIONAL	OBSERVACIÓN
<NFUN>-<AREA REQUERENTE>-<SECUENCIAL>	<descripción >	FUN-<AREA REQUERENTE>-<SECUECNIAL>	

Tabla 2: Características del Producto de Software

4.3. Requerimientos de Rendimiento

[Especificar el nivel de uso del software en relación con el tiempo de respuesta y el uso de recursos]

4.4. Requerimientos de Diseño

[Listar las restricciones de diseño del sistema]

4.5. Atributos de calidad de software

Atributo de calidad	Especificación
Confiabilidad	[Especifica la ejecución del software, la madurez y tolerancia a fallas]
Disponibilidad	[Especifica la ejecución del software durante un tiempo o periodo de prueba]
Seguridad	[Especifica el nivel de protección de la infraestructura computacional]
Mantenibilidad	[Descripción de los elementos que ayuden a dar mantenimiento al producto]
Portabilidad	[Describir las características que permite al software ser utilizado en otro lugar]

Tabla 3 Atributos de calidad de software

4.6. Otros requerimientos

4.6.1. Reutilización

[Describir si el producto o bus producto puede ser utilizado por un tercer producto]

¹ En la columna del identificador funcional, identificamos con que requerimiento funcional se encuentra relacionado el requerimiento no funcional

ANEXO 7: DISEÑO DE ARQUITECTURA – ÁGIL



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

|

DISEÑO Y ARQUITECTURA - ÁGIL

[Nombre de Proyecto]

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madruñero	Creacion de Plantilla		07/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

El objetivo del documento es definir de manera formal la arquitectura a ser utilizada en el proyecto de software, sus componentes, interfaces, métodos y técnicas de comunicación entre componentes y explicar las razones por las que se optó por la arquitectura descrita.

1.2. GLOSARIO DE TERMINOS

- ~~Stakeholders~~: Es el o los grupos de interesados en el proyecto de software.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Stakeholders de diseño

2.1. Diagrama de Stakeholders

[Diseñar un diagrama de los stakeholders y sus consideraciones en el proyecto de software].

2.2. Vista de diseño y relaciones con las consideraciones

[Describe la lista de vistas de diseño y sus relaciones con las consideraciones de diseño].

3. Descripción de arquitectura de software

3.1. Definición general de la arquitectura de software

[Describe brevemente la aplicación de software y sus requerimientos funcionales y de calidad].

3.2. Vista de Módulos

[Describe brevemente la aplicación de software y sus requerimientos funcionales y de calidad].

3.2.1. Vista de arquitectura de software

3.2.1.1. Revisión

[Una breve descripción de la vista actual].

3.2.1.2. Presentación inicial

[Diseñar un diagrama que muestre los elementos y relaciones en esta vista, si no estás utilizando una notación UML que es un estándar debe indicar la notación clave].

3.2.1.3. Catálogo de Elementos

[Se define un catálogo de elementos como una entrada adicional que provee información Ejemplo Diagramas de secuencia UML o grafico de estados]

3.2.1.4. Diagrama de Contexto

[Diseñar un diagrama de contexto que muestre el alcance de la parte del sistema representada, usualmente muestra la parte del sistema como una única, distinguida caja en el medio rodeada por otras cajas que son las entidades externas]

3.2.1.5. Guía de Variabilidad

[Describir algún mecanismo de variabilidad que utiliza el sistema mostrado Ejemplo: de la variabilidad incluye: componentes opcionales (~~plugins, add-ons~~), replicación configurable de componentes y conectores]

3.2.1.6. Razón Fundamental



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

[Describe restricciones, resultados de análisis y experimentos y requerimientos arquitectónicos significativos que afectan la vista]

3.3. Vista de componentes y conectores

[La vista de componentes muestra el comportamiento del software en tiempo de ejecución, un componente es una de las unidades principales de procesamiento del sistema en ejecución, pueden ser servicios, procesos, hitos y filtros, repositorios, puntos y un conector es un mecanismo de interacción entre los componentes estos incluyen canales, colas, protocolos de petición y respuesta.]

3.3.1. Vista arquitectura de software 2

3.3.1.1. Revisión

[Una breve descripción de la vista actual].

3.3.1.2. Presentación inicial

[Diseñar un diagrama que muestre los elementos y relaciones en esta vista, si no estás utilizando una notación UML que es un estándar debe indicar la notación clave].

3.3.1.3. Catálogo de Elementos

[Se define un catálogo de elementos como una entrada adicional que provee información Ejemplo Diagramas de secuencia UML o grafico de estados]

3.3.1.4. Diagrama de Contexto

[Diseñar un diagrama de contexto que muestre el alcance de la parte del sistema representada, usualmente muestra la parte del sistema como una única, distinguida caja en el medio rodeada por otras cajas que son las entidades externas]

3.3.1.5. Guía de Variabilidad

[Describir algún mecanismo de variabilidad que utiliza el sistema mostrado Ejemplo: de la variabilidad incluye: componentes opcionales (plugins, add-ons), replicación configurable de componentes y conectores]

3.3.1.6. Razón Fundamental

[Describe restricciones, resultados de análisis y experimentos y requerimientos arquitectónicos significativos que afectan la vista]

3.4. Asignación de vistas

3.4.1. Vista arquitectura de software 2

3.4.1.1. Revisión

[Una breve descripción de la vista actual].

3.4.1.2. Presentación inicial

[Diseñar un diagrama que muestre los elementos y relaciones en esta vista, si no estás utilizando una notación UML que es un estándar debe indicar la notación clave].

3.4.1.3. Catálogo de Elementos

[Se define un catálogo de elementos como una entrada adicional que provee información Ejemplo Diagramas de secuencia UML o grafico de estados]

3.4.1.4. Diagrama de Contexto

[Diseñar un diagrama de contexto que muestre el alcance de la parte del sistema representada, usualmente muestra la parte del sistema como una única, distinguida caja en el medio rodeada por otras cajas que son las entidades externas]



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

3.4.1.5. Guía de Variabilidad

[Describir algún mecanismo de variabilidad que utiliza el sistema mostrado

Ejemplo: de la variabilidad incluye: componentes opcionales (~~plugins, add-ons~~), replicación configurable de componentes y conectores]

3.4.1.6. Razón Fundamental

[Describe restricciones, resultados de análisis y experimentos y requerimientos arquitectónicos significativos que afectan la vista]

3.5. Mapeo entre vistas

[Especificar como los elementos de una vista se mapean a los elementos de otra vista]

3.6. Razón Fundamental

[indica información sobre la arquitectura de software, describe antecedentes y razones. Explica las restricciones y las influencias que llevaron a la arquitectura actual, además de mencionar los enfoques arquitectónicos utilizados]

ANEXO 8: RESULTADOS DE VERIFICACIÓN (CHECK LIST)



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: VERIFICACIÓN DE RESULTADOS - ÁGIL

|

VERIFICACIÓN DE RESULTADOS - ÁGIL

[Nombre de Proyecto]

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madruñero	Creación de Plantilla		07/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: VERIFICACIÓN DE RESULTADOS - ÁGIL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

El objetivo del documento es generar un informe sobre la verificación realizada a través del ~~check list~~ que se tiene y con ello determinar qué actividades, pasos o tareas hacen falta cumplir.

1.2. GLOSARIO DE TERMINOS

- ~~Stakeholders~~: Es el o los grupos de interesados en el proyecto de software.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Información General

Fecha de Revisió.	[dd/mm/yyyy]
Lugar de revisión	[Lugar donde se realizó la revisión]
Participantes	[Listado de Participantes]
Duración de la revisión	[Indicar el tiempo de duración de la revisión]
Reglas de aprobación y verificación	[indicar las reglas de aceptación del producto Ejemplo: 100% requerimientos cumplidos]
Observación	

Tabla 1: Información General

3. Listado de comprobación

[Se copia el listado de preguntas validado en el cuestionario de verificación, dependiendo de la metodología que se utilice la lista de verificación puede variar.]

4. Resultados de la Verificación

[Se realiza un cuadro de resultados, donde se indica el resultado los, indicando los componentes aprobados y no aprobados]

Elementos Aprobados	[Identificar las historias de usuario aprobadas]
Elementos no aprobados	[Identificar historia de usuarios con novedades]
Elementos Pendientes	[Identificar los problemas o defectos encontrados en la verificación]
Defectos identificados	[Identificar los problemas o defectos encontrados en la verificación]

VERIFICACIÓN DE RESULTADOS



Nombre del proyecto		
#	Preguntas	Aprobado?
1 Trazabilidad del Requerimiento		
1,1	El requerimiento tiene componentes designados para la implementación	<input type="checkbox"/>
1,2	La trazabilidad entre requerimientos y componentes se encuentra documentada	<input type="checkbox"/>
1,3	Cada componente de software cuenta con casos de pruebas y procedimientos de pruebas asociados	<input type="checkbox"/>
1,4	La trazabilidad entre requerimientos y casos de pruebas se encuentra documentado	<input type="checkbox"/>
2 Implementación de Requerimientos		
2,1	Cada componente tiene interfaces definidas	<input type="checkbox"/>
2,2	El código fuente se encuentra estandarizado de acuerdo al estandar definido por la	<input type="checkbox"/>
2,3	Se realizaron pruebas unitarias relacionadas con las funcionalidades	<input type="checkbox"/>
2,4	Se mantiene un repositorio con las versiones de las pruebas unitarias realizadas	<input type="checkbox"/>
3 Relación con Requerimientos no Funcionales		
3,1	Los componentes cumplen con los requerimientos indicados	<input type="checkbox"/>
3,2	Los componentes cumplen con los requerimientos no funcionales (portabilidad, usabilidad, etc)	<input type="checkbox"/>
3,3	Los componentes cumplen con los estándares de seguridad establecidos	<input type="checkbox"/>
4		

ANEXO 9: RESULTADOS DE VALIDACIÓN (CHECK LIST)



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: VALIDACIÓN DE RESULTADOS - ÁGIL

|

VALIDACIÓN DE RESULTADOS - ÁGIL

[Nombre de Proyecto]

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madruñero	Creación de Plantilla		07/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: VALIDACIÓN DE RESULTADOS - ÁGIL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

El objetivo de este documento es dejar evidencia de la validación realizada, en la validación el resultado final del desarrollo del software se debe ajustar a lo que le usuario necesitaba es decir las necesidades

1.2. GLOSARIO DE TERMINOS

- ~~Stakeholders~~: Es el o los grupos de interesados en el proyecto de software.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Información General

Fecha de Revisión.	[dd/mm/yyyy]
Lugar de revisión	[Lugar donde se realizó la validación]
Participantes	[Listado de Participantes]
Duración de la revisión	[Indicar el tiempo de duración de la validación]
Reglas de aprobación y validación	[indicar las reglas de aceptación del producto Ejemplo: 100% requerimientos cumplidos]
Observación	

Tabla 1: Información General

3. Listado de comprobación

[Se copia el listado de preguntas validado en el cuestionario de verificación, dependiendo de la metodología que se utilice la lista de verificación puede variar.]

4. Resultados de la validación

[Se realiza un cuadro de resultados, donde se indica el resultado los, indicando los componentes aprobados y no aprobados.

Elementos Aprobados	[Identificar las historias de usuario aprobadas]
Elementos no aprobados	[Identificar historia de usuarios con novedades]
Elementos Pendientes	[Identificar los problemas o defectos encontrados en la validación]
Defectos identificados	[Identificar los problemas o defectos encontrados en la validación]

ANEXO 10: SOLICITUD DE CAMBIOS – ÁGIL.

UNIVERSIDAD TECNICA DEL NORTE			
DIRECCIÓN DE DESARROLLO INFORMATICO E INNOVACIÓN			
INFORMACION GENERAL			Fecha solicitud
Nombre del proyecto:			
Módulo(s)			
Prioridad de la Solicitud			Fecha de Recepción
<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
DEFINICION DE SOLICITUD DE CAMBIO			
Descripción			
[Describe el o los cambios propuestos]			
Justificación			
[Describe la justificacion del cambio]			
Impacto			
[Describe el impacto de no realizar el cambio]			
Alternativas de Solución			
[Describe al menos 2 alternativas de solucion de no proceder el cambio]			
ANÁLISIS DE SOLICITU DE CAMBIO			
[Describir el impacto de realiza el cambio en el o los componentes que serian afectados]			
REVISIÓN INICIAL			
Fecha de Revisión	Nombre del inspector	Rol	Recomendación
			<input type="checkbox"/> Aprobar <input type="checkbox"/> Rechazar <input type="checkbox"/> Posponer hasta:
			<input type="checkbox"/> Aprobar <input type="checkbox"/> Rechazar <input type="checkbox"/> Posponer hasta:
APROBACIÓN FINAL			
Fecha de aprobación	Nombre del inspector	Título	Recomendación
			<input type="checkbox"/> Aprobar

ANEXO 11: ACTA DE ACEPTACIÓN – ÁGIL.



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ACTA DE ACEPTACIÓN - ÁGIL

ACTA DE ACEPTACIÓN

Lugar y Fecha: Ibarra, 18 de febrero de 2016

DETALLES DEL CLIENTE	
Nombre del Cliente:	
Nombre del Aplicativo/Proyecto:	

Detalle del Aplicativo/Proyecto	
Fecha:	

NRO	NOMBRE DEL ENTREGABLE	DESCRIPCION DEL ENTREGABLE	FECHA DE ENTREGA

Analista

Cliente

Observaciones: Este documento representa la entrega de los reportes solicitados por la Dirección de Desarrollo Tecnológico e Informático al usuario final

ANEXO 12: ENUNCIADO DEL TRABAJO – ÁGIL



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ENUNCIADO DEL TRABAJO - AGÍL

|

ENUNCIADO DEL TRABAJO - ÁGIL

Portafolio Estudiantil Móvil – Módulo Consulta de Notas

Versión 1.0

08/11/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madruero	Creador de Documento	Analista	08/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ENUNCIADO DEL TRABAJO - AGÍL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

Este documento tiene como objetivo realizar un levantamiento de información del proyecto a ser ejecutado, indicando los entregables dependiendo de los requerimientos de alto nivel mencionados por el cliente.

1.2. GLOSARIO DE TERMINOS

Identificador: Código único generado para un registro

Entregables: Son los documentos físicos y/o electrónicos que se elaboran el desarrollo del proyecto.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Información General

La Universidad Técnica del Norte es una institución de educación superior encargada de formar estudiantes de Tercer Nivel, la universidad está ubicada en la ciudad de Ibarra, provincia de Imbabura en la avenida 17 de julio sector el Olivo, cuando con alrededor de 7000 estudiantes aproximadamente.

Cuenta con una Dirección de Tecnologías e Informático, la cual se ha encarga de la automatización de los procesos académicos.

2.1. Definir Lista de Necesidades

A continuación, se define la siguiente lista de necesidades

1. Ingresar a la aplicación a través de una aplicación móvil
2. Usar las credenciales de autenticación que nos proporcione la Dirección de Desarrollo Tecnológico e Informático
3. Consultar las notas del semestre actúa.

3. Alcance del proyecto

El proyecto tiene como objetivo proporcionar una aplicación móvil que pueda ser descargada desde la [app Store](#) de Google y consulte las notas del periodo actual del estudiante.

3.1. Proyecto incluye

- Consumir un servicio web que proporcione la Dirección de Desarrollo Tecnológico e Informático
- Un módulo de consulta de notas del periodo actual.
- Una interfaz que se adapte al dispositivo en horizontal o vertical.
- La autenticación a través de un servicio web que nos proporcione la dirección.

3.2. Proyecto excluye

- Federación de claves con el correo institucional (Office 365)
- No incluye el funcionamiento de los módulos de:
 - o Perfil Estudiantil
 - o Registros Académicos



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ENUNCIADO DEL TRABAJO - AGÍL

- Horarios
- Matriculas
- Evaluación Docente
- Pagos Realizados

4. Objetivos del proyecto

- Desarrollar un prototipo de aplicación móvil para la Universidad Técnica del Norte, donde la comunidad universitaria pueda consultar las notas y el porcentaje de asistencias obtenidos durante el semestre,

5. Requisitos Generales del Proyecto

Id ¹	Descripción ²	Estimación ³	Prioridad ⁴
REQ-DDTI-001	Como Alumno necesito : Poder consultar mis notas por medio de mi teléfono celular	1	Media
REQ-DDTI-002	Como Administrador necesito : Publicar las notas de los estudiantes y que puedan tener acceso inmediatamente a través de cualquier dispositivo	2	Media
REQ-DDTI-003	Como Alumno necesito : que mi información sea privada a través de la aplicación móvil, que para consultarla tenga que autenticarme.	1	Alta

Para llenar la tabla, tomar en cuenta el anexo 001 denominado Requisitos Generales del Proyecto

6. Entregables

Identificador	Entregable	Tipo	Fecha estimada	Receptor	Observación
ENT-DDTI-001	Enunciado del trabajo	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-002	Plan de Proyecto	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-003	Especificación de Requerimientos	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-004	Diseño de Arquitectura	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-005	Verificación	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-006	Validación	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-007	Aplicativo	Fuentes	23/11/2017	Ing. Juan Cargos García	

Tabla 7 Entregables

Guía con el Anexo 002 - Entregables

¹ Identificador

² Descripción del requerimiento

³ Estimación del esfuerzo

⁴ Prioridad de trabajo

ANEXO 13: PLAN DE PROYECTO – ÁGIL



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - ÁGIL

PLAN DE PROYECTO - ÁGIL

Portafolio Estudiantil Móvil – Módulo Consulta de Notas

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madruñero	Creación de Plantilla	Tesista	04/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

El presente documento tiene como objetivo plantear el proyecto de manera específica y enfocarse a factores como la planificación, levantamiento de requerimientos, gestión de la configuración de manera rápida y accesible para la metodología ágil.

1.2. GLOSARIO DE TERMINOS

- **Identificador:** Código único generado para un registro
- **Sprint:** Termino que se utiliza para nombras a las iteraciones realiza en SCRUM.
- **Riesgos:** Es la posibilidad que un evento ocurra y afecte negativamente a la consecución de los objetivos planteados
- **Entregables:** Son los documentos fisicos y/o electrónicos que se elaboran el desarrollo del proyecto.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Información General

2.1. Visión General del Proyecto

La Universidad Técnica del Norte no cuenta con una aplicación móvil, que gestione procesos académicos desde los dispositivos móviles, la aplicación tiene como objetivo promover y construir una aplicación que agilite el servicio a la comunidad Universitaria.

2.2. Objetivo del Proyecto

Desarrollar un prototipo de aplicación móvil para la Universidad Técnica del Norte, donde la comunidad universitaria pueda consultar las notas y el porcentaje de asistencias obtenidos durante el semestre

2.3. Requisitos Generales del Proyecto

Id ¹	Descripción ²	Estimación ³	Prioridad ⁴
REQ-DDTI-001	Como Alumno necesito: Poder consultar mis notas por medio de mi teléfono celular	1	Media
REQ-DDTI-002	Como Administrador necesito: Publicar las notas de los estudiantes y que puedan tener acceso inmediatamente a través de cualquier dispositivo	2	Media
REQ-DDTI-003	Como Alumno necesito: que mi información sea privada a través de la aplicación móvil, que para consultarla tenga que autenticarme.	1	Alta

Tabla 1: Requisitos Generales del Proyecto

Para llenar la tabla, tomar en cuenta el anexo 001 denominado Requisitos Generales del Proyecto

¹ Identificador

² Descripción del requerimiento

³ Estimación del esfuerzo

⁴ Prioridad de trabajo



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

3. Responsabilidades del proyecto

Identificador	Nombres	Apellidos	Dirección	Correo Electrónico	Rol ⁵
ROL-DDTI-001	Juan Carlos	García		jgarcia@utn.edu.ec	Líder de Proyecto
ROL-DDTI-002	Edwin Ramiro	Madruñero Padilla		edwmadru@hotmail.com	Desarrollador
ROL-DDTI-003	Edwin Ramiro	Madruñero Padilla		edwmadru@hotmail.com	Analista
ROL-DDTI-004	Juan Carlos	García		jgarcia@utn.edu.ec	Cliente

Tabla 2: Responsabilidades del proyecto

Los roles y la descripción de las funcionalidades de cada uno se detallan en el Anexo 002- Roles.

4. Plan de Entrega

Fecha estimada de entrega	Sprint	Id Requerimiento	Observación
23/11/2017	Nro.002	REQ-DDTI-001	
23/11/2017	Nro.003	REQ-DDTI-002	
16/11/2017	Nro.001	REQ-DDTI-003	

Tabla 3: Plan de Entrega

5. Gestión de Riesgos

ID	Tipo	Descripción	Plan de Mitigación	Responsables
RISK-DDTI-001	RISK-TYPE-INT	Integración con el ERP académico a través de servicios web	<ol style="list-style-type: none"> 1. Desarrollar el cliente del servicio web. 2. Realizar las pruebas unitarias 3. Incorporar un control para identificar si tenemos acceso a internet. 	Desarrollador
RISK-DDTI-002	RISK-TYPE-REQ	Baja definición de los requerimientos	<ol style="list-style-type: none"> 1. Realizar una reunión de levantamiento. 2. Liberar desarrollos permanentemente para que el 	Cliente Analista Líder

⁵ Los roles que pueden ser indicados son: Cliente, Líder de Proyecto, Desarrollador y Analista



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

			usuario apoye con la validación	
--	--	--	---------------------------------	--

Tabla 4 Identificación de Riesgos

Revisar en el Anexo 003, Identificación de Riesgos como registrar la información en la tabla del plan de proyecto.

6. Cronograma y presupuesto

6.1. Cronograma

Hitos ⁶	Descripción	Criterio de Éxito	Fecha Estimada	Observación
H01-DDTI-001	Pantalla de Logio	Que ejecute la comprobación de credenciales correctamente	16/11/2017	
H01-DDTI-002	Opción de consulta de notas	Que la pantalla de consulta de notas muestre las notas del semestre	23/11/2017	

Tabla 5: Cronograma

6.2. Presupuesto

Presupuesto por Hito	REC001	REC004	REC007	SUBTOTAL
	H01-DDTI-001	10 USA	10 USA	30 USA
H01-DDTI-002	10 USA	10 USA	30 USA	50 USA
Total	20 USA	20 USA	60 USA	100 USA

Tabla 6: Presupuesto

Para el registro del presupuesto usted puede revisar la Guía con el Anexo 004



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

7. Entregables

Identificador	Entregable	Tipo	Fecha estimada	Receptor	Observación
ENT-DDTI-001	Enunciado del trabajo	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-002	Plan de Proyecto	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-003	Especificación de Requerimientos	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-004	Diseño de Arquitectura	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-005	Verificación	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-006	Validación	Artefactos	23/11/2017	Ing. Juan Cargos García	
ENT-DDTI-007	Aplicativo	Fuentes	23/11/2017	Ing. Juan Cargos García	

Tabla 7 Entregables

Guía con el Anexo 005 - Entregables

8. Gestión de la Configuración

8.1. Identificación de elementos de configuración

Los elementos que se deben tomar en cuenta para la gestión de la configuración son:

- Enunciado del trabajo
- Plan de Proyecto
- Especificación de Requerimientos
- Diseño de Arquitectura
- Verificación
- Validación
- Fuentes del aplicativo

8.2. Herramientas, Técnicas y Metodologías

Las herramientas, Técnicas y Metodologías utilizadas para la gestión de la configuración son:

- **Herramienta**
- **Técnica**
- **Metodología**



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: PLAN DE PROYECTO - AGÍL

8.3. Acceso a repositorios

[Se debe describir el repositorio a utilizar en el ~~proyecto~~, donde se debe indicar el tipo de repositorio, espacio físico, dispositivos electrónicos, ubicación del repositorio.]

Denominación	Valué
Tipo de repositorio	- Electrónico, - Físico, - Dispositivo electrónico
Descripción del repositorio	
Ubicación del repositorio	
Dirección URL	

Tabla 8 Acceso a repositorios

9. Almacenamiento y recuperación de Entregables

9.1. Mecanismo de Almacenamiento y Manipulación de Entregables

Identificador	Artefacto / entregable	Técnica de Almacenamiento	Descripción	Responsable
ENT-DDTI-001	Enunciado del trabajo	Deltas inversos	Artefacto generado	Juan Carlos García
ENT-DDTI-002	Plan de Proyecto	Deltas inversos	Artefacto generado	Juan Carlos García
ENT-DDTI-003	Especificación de Requerimientos	Deltas inversos	Artefacto generado	Juan Carlos García
ENT-DDTI-004	Diseño de Arquitectura	Deltas inversos	Artefacto generado	Juan Carlos García
ENT-DDTI-005	Verificación	Deltas inversos	Artefacto generado	Juan Carlos García
ENT-DDTI-006	Validación	Deltas inversos	Artefacto generado	Juan Carlos García
ENT-DDTI-007	Aplicativo	Deltas inversos	Código fuente	Juan Carlos García

Tabla 9: Mecanismo de Almacenamiento y Manipulación de Entregables

9.2. Mecanismo de Respaldo y Recuperación de Información

9.2.1. Mecanismo de Respaldo

Se realiza un respaldo cada vez que exista una modificación en el sistema, para tener acceso al sistema de versionamiento debe crearse un usuario en el servidor y asignar los permisos de lectura.

ANEXO 14: ESPECIFICACIÓN DE REQUERIMIENTOS – ÁGIL



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

|

ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

Portafolio Estudiantil Móvil – Módulo Consulta de Notas

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

La especificación de requerimientos es un documento formal que se utiliza para dejar escrito de manera formal las funcionalidades, característica, procesos y formas de acceso que deberá tener un producto de software solicitado

1.2. GLOSARIO DE TERMINOS

- **Identificador:** Código único generado para un registro
- **Requerimientos:** Comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado, tomando en cuenta los diversos requisitos de las partes interesadas, que pueden entrar en conflicto entre ellos
- **Requerimientos Funcionales:** Define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas.
- **Entregables:** Son los documentos físicos y/o electrónicos que se elaboran el desarrollo del proyecto.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Información General

2.1. Visión General

La Universidad Técnica del Norte no cuenta con una aplicación móvil, que gestione procesos académicos desde los dispositivos móviles, la aplicación tiene como objetivo promover y construir una aplicación que agilite el servicio a la comunidad Universitaria

2.2. Alcance

El proyecto tiene como objetivo proporcionar una aplicación móvil que permita consulte las notas del periodo actual del estudiante.

2.2.1. Proyecto incluye

- Consumir un servicio web que proporcione la Dirección de Desarrollo Tecnológico e Informático
- Un módulo de consulta de notas del periodo actual.
- Una interfaz que se adapte al dispositivo en horizontal o vertical.
- La autenticación a través de un servicio web que nos proporcione la dirección.

2.2.2. Proyecto excluye

- Gestionar la autenticación con la integración a través de servicios web
- No incluye el funcionamiento de los módulos de:
 - Perfil Estudiantil
 - Registros Académicos
 - Horarios
 - Matriculas
 - Evaluación Docente
 - Pagos Realizados



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

3. Descripción General

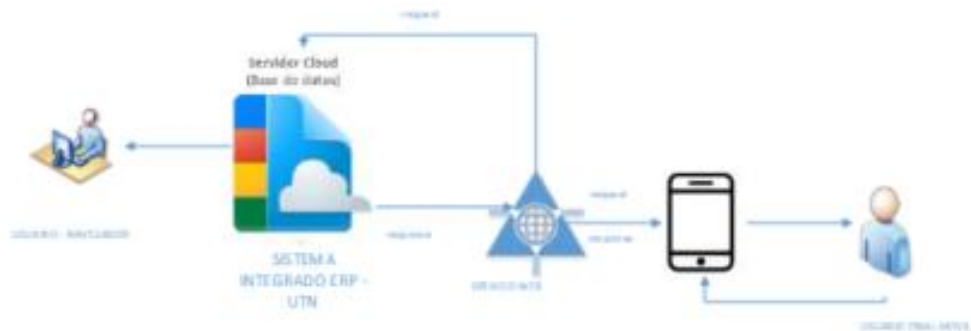
3.1. Funciones de producto

IDENTIFICADOR	DESCRIPCIÓN
FUN-DDTI-001	Consultar las notas de cada estudiante de manera personal.
FUN-DDTI-002	Utilizar el usuario y clave de acceso que se utiliza en el sistema ERP

Tabla 1: Listado de Funciones de producto

3.2. Perspectiva del producto

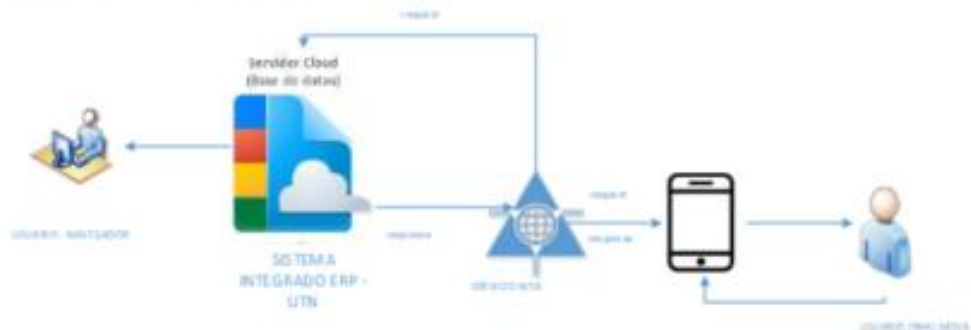
3.2.1. Interfaces del Sistema



3.2.2. Interface de Hardware

Este aplicativo se puede utilizar desde cualesquiera dispositivos móviles que tengan sistema operativo Android desde la versión 5.0

3.2.3. Interface de Comunicación





UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

4. Requerimientos Específicos

4.1. Requerimientos de interfaces Externas

Un requerimiento es que el equipo de Desarrollo de la Universidad Técnica del Norte nos exponga dos métodos en un servicio web

Método	Características
Login	Método que recibe como parámetro el usuario y clave para realizar la validación de credenciales y retorno nombres completos, carrera y una bandera indicando si la validación de credenciales es correcta o no
Consulta de Notas	Método que enviando como parámetro la cedula de identidad y me permite retornar las asignaturas y las notas que tiene registradas.

4.2. Características del Producto de Software

IDENTIFICADOR	DESCRIPCIÓN	¹ IDENTIFICADOR FUNCIONAL	OBSERVACIÓN
NFUN-DDTI-001	Seguridad en acceso al aplicativo	FUN-DDTI-002	
NFUN-DDTI-002	Usabilidad	FUN-DDTI-001	
NFUN-DDTI-003	Portabilidad	FUN-DDTI-002	

Tabla 2: Características del Producto de Software

4.3. Requerimientos de Rendimiento

La aplicación al hacer uso de servicio web, su consulta y respuesta será rápida, por lo que se solicita la exposición de los servicios web

4.4. Requerimientos de Diseño

No se tiene

4.5. Atributos de calidad de software

Atributo de calidad	Especificación
Confiabilidad	La app tendrá un método que me verifica la conectividad con el internet.
Disponibilidad	Se realizarán las pruebas necesarias para que el aplicativo <u>este</u> disponible
Seguridad	Se implementa un servicio para el Login de los usuarios
Mantenibilidad	La estructura y herramienta en la que es desarrollado da una curva de aprendizaje es rápida
Portabilidad	Al ser una aplicación móvil, y desarrollado para dispositivos con sistema operativo Android se puede utilizar para cualquier dispositivo con Android.

Tabla 3 Atributos de calidad de software

4.6. Otros requerimientos

4.6.1. Reutilización

¹ En la columna del identificador funcional, identificamos con que requerimiento funcional se encuentra relacionado el requerimiento no funcional

ANEXO 15: DISEÑO DE ARQUITECTURA – ÁGIL



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

|

DISEÑO Y ARQUITECTURA - ÁGIL

Portafolio Estudiantil Móvil – Módulo Consulta de Notas

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madruñero	Creación de Plantilla		07/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

El objetivo del documento es definir de manera formal la arquitectura a ser utilizada en el proyecto de software, sus componentes, interfaces, métodos y técnicas de comunicación entre componentes y explicar las razones por las que se optó por la arquitectura descrita.

1.2. GLOSARIO DE TERMINOS

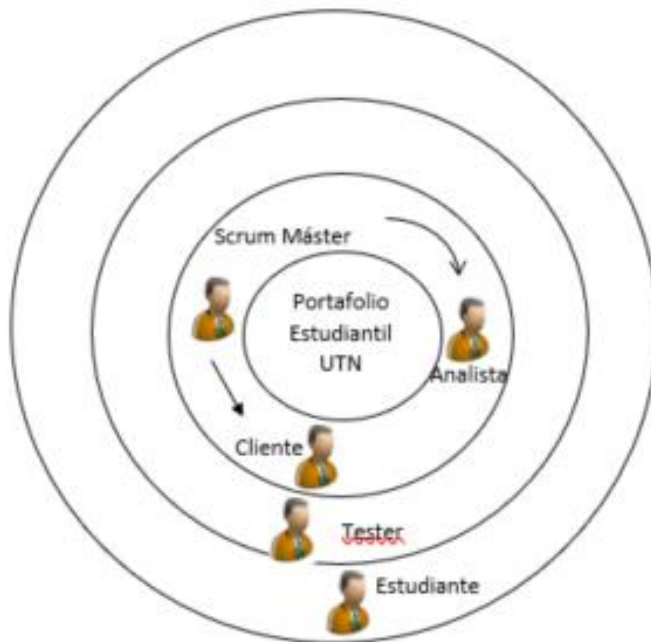
- **Stakeholders:** Es el o los grupos de interesados en el proyecto de software.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Stakeholders de diseño

2.1. Diagrama de Stakeholders





UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

2.2. Vista de diseño y relaciones con las consideraciones

- Vista de servicios
En la capa de servicio se desarrolló construcción del modelo (clase java) para transformar de formato json a objetos java y pueda ser tratado.
- Vista de presentación
En la capa de presentación se crearon actividades las cuales fueron enlazadas a clases java donde se codifico el funcionamiento.
- Vista de negocio
Se creo la estructura de la consulta de calificaciones, estructura se cargará la información con lo entregado por el servicio web.

3. Descripción de arquitectura de software

3.1. Definición general de la arquitectura de software

La aplicación debe contar con las siguientes características:

- Requerimientos funcionales
Los requerimientos funcionales son que el estudiante pueda ingresar a través de su teléfono móvil y consultar las notas del semestre actual.
El sistema debe tener una manera de autenticación.
- Requerimientos no funcionales
Que el estudiante pueda autenticarse de manera segura con las credenciales que tiene para el portafolio estudiantil que esta publicado en el portal web.
- Atributos de calidad
Los atributos de calidad que tiene el aplicativo son:
 - o Usabilidad
 - o Disponibilidad
 - o Seguridad
 - o Confiabilidad.



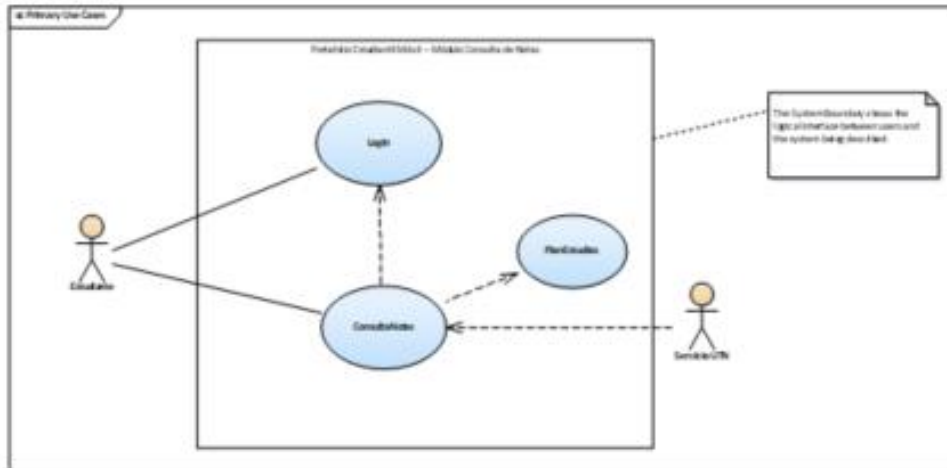
UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL



Para la definición de la arquitectura se toma en cuenta la capa de presentación, capa de servicio y capa de negocio.

3.2. Vista de Módulos

La arquitectura se encuentra definida en tres vistas: presentación, servicios y lógica de negocio; entre las 3 vistas se distribuye los siguientes requerimientos funcionales y no funcionales.

El estudiante debe ingresar a través de un celular y poder consultar las notas de su semestre actual.

- El aplicativo debe ser fácil de usar e intuitivo.
- Debe ser seguro para brindar confiabilidad a los estudiantes.

3.2.1. Vista de arquitectura de software

3.2.1.1. Revisión

La arquitectura cuenta de 3 vistas que son: presentación, lógica y de servicios; las tres vistas se encuentran integradas para lograr el funcionamiento del aplicativo y puesta en funcionamiento.

3.2.1.2. Presentación inicial



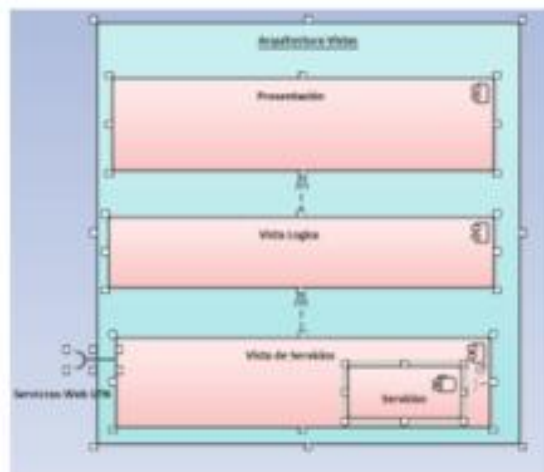
UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

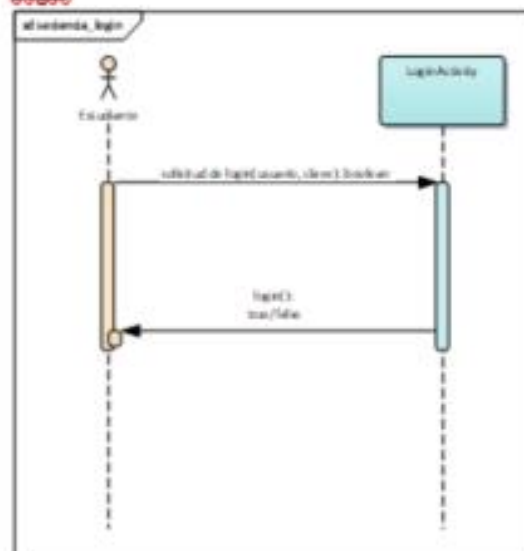
ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL



3.2.1.3. Catálogo de Elementos

- Login



- Consulta de notas



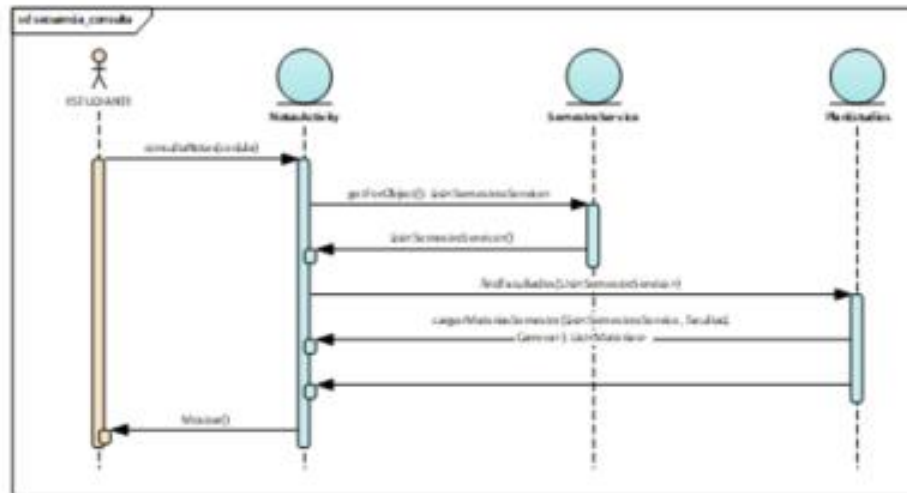
UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

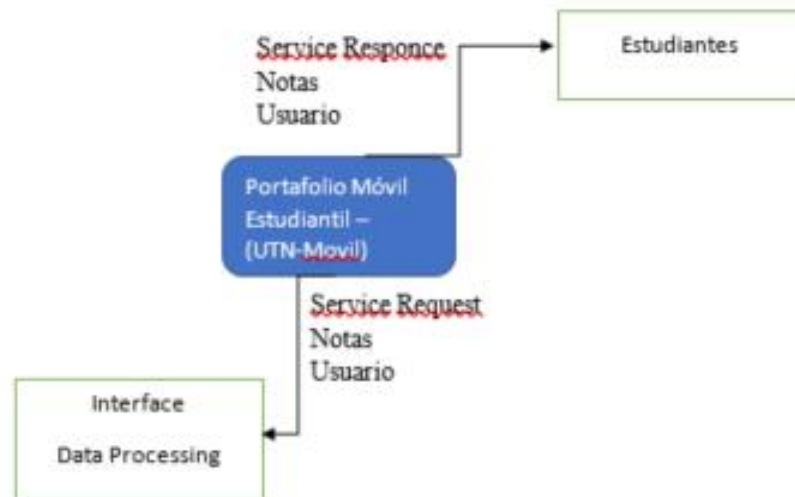
UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL



3.2.1.4. Diagrama de Contexto





UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: ESPECIFICACIÓN DE REQUERIMIENTOS - ÁGIL

El sistema cuenta con varios **plugins** que ayudan a la interpretación del servicio web en formato **json**; los **plugins** son:

```
compile fileTree(dir: ['**/jar'], dir: 'lib')
androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
    exclude group: 'com.android.support', module: 'support-annotations'
})
//compile group: 'org.mozilla.android', name: 'httpclient', version: '4.4.1.1'
compile 'com.android.support:appcompat-v7:25.3.1'
compile 'com.android.support.constraint:constraint-layout:1.0.2'
compile 'com.android.support:design:25.3.1'
compile 'com.android.support:support-v4:25.3.1'
compile 'com.android.support:support-vector-drawable:25.3.1'
compile 'org.springframework.android:spring-android-rest-template:1.0.1.RELEASE'
compile 'com.fasterxml.jackson.core:jackson-databind:2.3.2'
testCompile 'junit:junit:4.11'
```

3.2.1.6. Razón Fundamental

Para el desarrollo de la aplicación móvil se desarrolló un análisis considerando que la Universidad Técnica del Norte no cuenta con una aplicación móvil que ofrezca servicios para los estudiantes, esta aplicación no se encuentra federada con la aplicación de Microsoft office 365, por lo que para realizar la seguridad de usuario se consumió un servicio web.

Por temas de tiempo solo se desarrolló un prototipo que contiene el módulo de consulta de notas actuales de cada estudiante.

ANEXO 16: RESULTADOS DE VERIFICACIÓN – ÁGIL



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: VERIFICACIÓN DE RESULTADOS - ÁGIL

|

VERIFICACIÓN DE RESULTADOS - ÁGIL

Portafolio Estudiantil Móvil – Módulo Consulta de Notas

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madruero	Creación de Plantilla		07/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: VERIFICACIÓN DE RESULTADOS - ÁGIL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

El objetivo del documento es generar un informe sobre la verificación realizada a través del **check list** que se tiene y con ello determinar qué actividades, pasos o tareas hacen falta cumplir.

1.2. GLOSARIO DE TERMINOS

- **Stakeholders:** Es el o los grupos de interesados en el proyecto de software.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Información General

Fecha de Revisión	23/01/2018
Lugar de revisión	Dirección de Desarrollo Tecnológico e Informático
Participantes	Ing. Lenin Chavez Ing. Edwin Madruñero
Duración de la revisión	3 horas
Reglas de aprobación y verificación	<ul style="list-style-type: none">- Validación de credenciales de usuario y clave- Consulta de notas del usuario logueado sin necesidad de ingresar más parámetros- Acceso de la aplicación a través de una interfaz móvil.- Aplicación fácil de manejar y entender
Observación	

Tabla 1: Información General

3. Listado de comprobación

Se verifico los siguientes factores:

- Trazabilidad de Requerimientos
 - El requerimiento tiene componentes definidos para la implementación
 - La trazabilidad entre los requerimientos y los componentes se encuentra documentada
 - Cada componente de software cuenta con casos de pruebas
- Implementación de Requerimientos
 - Cada interfaz tiene una interfaz definida
 - El código se encuentra estandarizado de acuerdo con el estándar de la Universidad Técnica del Norte.
 - Se realizaron pruebas unitarias relacionadas a las funcionalidades
- Relación con requerimientos no Funcionales
 - Los componentes cumplen con los requerimientos indicados
 - Los componentes cumplen con requerimientos no funcionales (portabilidad, usabilidad).
 - Los componentes cumplen los estándares de seguridad establecidos.



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: VERIFICACIÓN DE RESULTADOS - ÁGIL

4. Resultados de la Verificación

Elementos Aprobados	FUN-DDTI-001 FUN-DDTI-002 NFUN-DDTI-001 NFUN-DDTI-002 NFUN-DDTI-003
Elementos no aprobados	No existen requerimientos no aprobados
Elementos Pendientes	No existen requerimientos no aprobados
Defectos identificados	No existen requerimientos no aprobados

ANEXO 17: RESULTADOS DE VALIDACIÓN – ÁGIL



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: VALIDACIÓN DE RESULTADOS - ÁGIL

|

VALIDACIÓN DE RESULTADOS - ÁGIL

Portafolio Estudiantil Móvil – Módulo Consulta de Notas

Versión 1.0

31/10/2017

HISTORIAL DE CAMBIOS				
Versión	Autor	Comentarios	Responsable	Fecha
Versión 1.0	Ing. Edwin Madruñero	Creación de Plantilla		07/11/2017



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: VALIDACIÓN DE RESULTADOS - ÁGIL

1. Introducción

1.1. OBJETIVO DEL DOCUMENTO

El objetivo de este documento es dejar evidencia de la validación realizada, en la validación el resultado final del desarrollo del software se debe ajustar a lo que le usuario necesitaba es decir las necesidades

1.2. GLOSARIO DE TERMINOS

- **Stakeholders:** Es el o los grupos de interesados en el proyecto de software.

1.3. REFERENCIAS

(Ramos & Mendoza, 2014).

2. Información General

Fecha de Revisión	23/01/2018
Lugar de revisión	Dirección de Desarrollo Tecnológico e Informático
Participantes	Ing. Lenin Chavez Ing. Edwin Madruero
Duración de la revisión	3 horas
Reglas de aprobación y validación	<ul style="list-style-type: none">- Validación de credenciales de usuario y clave- Consulta de notas del usuario logueado, sin necesidad de ingresar más parámetros- Acceso de la aplicación a través de una interfaz móvil.- Aplicación fácil de manejar y entender
Observación	

Tabla 1: Información General

3. Listado de comprobación

- Contenido general de las historias de usuario
 - La Historia de usuario posee código único
 - La historia se describe en una ficha sin problemas
 - La historia sigue el formato establecido
 - Posee criterios de aceptación.
 - La historia de usuario es independiente y autónoma
 - Las historias de usuario son negociables
 - Las historias de usuario agregan valor al **stakeholders**.
 - Los requerimientos no tienen ambigüedad.
- Requerimientos
 - El levantamiento de las historias de usuario se realizó por medio de una entrevista
 - Las historias de usuario han sido aprobadas por lo **stakeholders**.
 - Los requerimientos se encuentran relacionados con el listado de funcionalidades
- Diseño y Arquitectura
 - Se encuentra diseñada la arquitectura utilizada en el proyecto
 - Se encuentra fundamentada la arquitectura utilizada.



UNIVERSIDAD TÉCNICA DEL NORTE

IBARRA – ECUADOR

UNIVERSIDAD: UNIVERSIDAD TÉCNICA DEL NORTE

ÁREA: DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

DOCUMENTO: VALIDACIÓN DE RESULTADOS - ÁGIL

4. Resultados de la validación

Elementos Aprobados	FUN-DDTI-001 FUN-DDTI-002 NFUN-DDTI-001 NFUN-DDTI-002 NFUN-DDTI-003
Elementos no aprobados	No existen requerimientos no aprobados
Elementos Pendientes	No existen requerimientos no aprobados
Defectos identificados	No existen requerimientos no aprobados
Elementos Aprobados	[Identificar las historias de usuario aprobadas]
Elementos no aprobados	[Identificar historia de usuarios con novedades]
Elementos Pendientes	[Identificar los problemas o defectos encontrados en la validación]
Defectos identificados	[Identificar los problemas o defectos encontrados en la validación]

ANEXO 18: ANÁLISIS DE METODOLOGÍAS RUP Y SCRUM UTILIZADA EN LA DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO DE LA UNIVERSIDAD TÉCNICA DEL NORTE

INTRODUCCIÓN

Para realizar el análisis del proceso de desarrollo de software que tiene la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte se tomó como base las metodologías utilizadas en los diferentes proyectos de software como son: RUP, SCRUM.

METODOLOGÍA RUP

Se elaboró una matriz de análisis tomando en cuenta los procesos, actividades y tareas que se realizan con RUP y se establece una comparativa con las tareas que se realizan en la dirección de desarrollo para los proyectos CRM.

A) FASE DE INICIO

A continuación, se muestra la siguiente tabla de análisis:

RUP - FASE INICIAL			
ACTIVIDAD	SUBACTIVIDAD	TAREA	DD TI
Concebir un nuevo proyecto	Concebir un nuevo proyecto	Identificar y valorar los riesgos	N
		Desarrollar el caso de negocio	N
		Iniciar el proyecto	S
		Revisión de aprobación del proyecto	S
Preparar los entornos del proyecto	Preparar el entorno para el proyecto	Personalizar el proceso de desarrollo para el proyecto	S
		Desarrollar el guion de desarrollo	N
		Preparar plantillas para el proyecto	N
		Preparar directrices para el proyecto	N
		Seleccionar y adquirir herramientas	N
	Crear entornos de gestión de la configuración (CM) de proyectos	Configurar el entorno de gestión de la configuración (CM)	N
		Crear espacios de trabajo de integración	S

Assess Business Status	Assess Business Status	Evaluar el objetivo de la organización	N
		Establecer y ajustar objetivos	N
		Identificar objetivos comerciales y KPI	N
		Análisis Arquitectónico Empresarial	S
		Capturar un vocabulario general	N
		Establecer reglas del negocio	N
Preparar el entorno para una iteración	Preparar el entorno para una iteración	Desarrollar el guion de desarrollo	N
		Preparar directrices para el proyecto	N
		Desarrollar la guía de estilo del manual	N
		Preparar plantillas para el proyecto	N
		Iniciar el proceso de desarrollo	S
		Configurar las herramientas	S
		Verificar la instalación y configuración de herramientas	S
Definir los planes del proyecto	Planificar el proyecto	Desarrollar el plan de medidas	N
		Desarrollar el plan de gestión de riesgos	N
		Desarrollar el plan de aceptación del producto	N
		Desarrollar el plan de resolución de problemas	N
		Desarrollar el plan de garantía de calidad	N
		Definir la organización y el personal del proyecto	S
		Definir los procesos de supervisión y control	N
		Planificar fases e iteraciones	N
		Compilar el plan de desarrollo de software	N
		Revisión de planificación del proyecto	S
		Supervisar y controlar el proyecto	Supervisar y controlar el proyecto
Supervisar el estado del proyecto	S		
Informe de estado	S		
Manejar las excepciones y los problemas	N		
Revisión del proyecto con la autoridad de revisión de proyectos (PRA)	S		
Organizar la revisión	N		
Dirigir la revisión	N		
Desarrollar la visión inicial	Analizar el problema	Crear un vocabulario común	N
		Buscar actores y guiones de uso	N
		Desarrollar la visión	S

		Desarrollar el plan de gestión de requisitos	N
	Conocer las necesidades del interesado	Crear un vocabulario común	N
		Obtener las solicitudes del interesado	S
		Desarrollar la visión	S
		Buscar actores y guiones de uso	N
		Desarrollar especificaciones suplementarias	S
		Gestionar las dependencias	S
Develop Domain Model		Develop Domain Model	Capture un vocabulario comercial común
	Mantener las reglas de negocios		N
	Análisis Arquitectónico Empresarial		N
	Detalle una entidad comercial		N
	Revise el Modelo de Análisis de Negocios		N
Gestionar el ámbito del sistema	Gestionar el ámbito del sistema	Desarrollar la visión	S
		Gestionar las dependencias	S
		Priorizar los guiones de uso	N
Definir el sistema	Definir el sistema	Desarrollar la visión	S
		Crear un vocabulario común	N
		Buscar actores y guiones de uso	N
		Desarrollar especificaciones suplementarias	S
		Gestionar las dependencias	S
Realizar la síntesis arquitectónica	Realizar la síntesis arquitectónica	Definir el contexto del sistema	S
		Análisis de la arquitectura	S
		Construir arquitectura de prueba de concepto	N
		Valorar la viabilidad de la arquitectura de prueba de concepto	N
Definir la misión de evaluación	Definir la misión de evaluación	Identificar motivadores de prueba	N
		Acordar la misión	N
		Identificar destinos de prueba	N
		Definir necesidades de valoración y rastreabilidad	N
		Identificar ideas de prueba	N
		Definir el enfoque de prueba	N
Supervisar la iteración	Supervisar la iteración	Adquirir personal	N
		Iniciar la iteración	N
		Revisión de los criterios de evaluación de la iteración	S
		Identificar y valorar los riesgos	N
		Valorar la iteración	N
		Revisión de la aceptación de la iteración	S

Planear la siguiente iteración	Planear la siguiente iteración	Desarrollar el plan de iteración	N
		Desarrollar el caso de negocio	N
		Revisión del plan de la iteración	N

Tabla 44: Tabla comparativa fase de inicio- RUP

Fuente: (Contraloría General de la República de Costa Rica, 2017)

Elaborado por: El autor

De este análisis realizado se obtuvo los siguientes resultados:

Total, de Tareas	83
Tareas Realizadas	28
Tareas no realizadas	55

Tabla 45: Resumen de Inicio

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

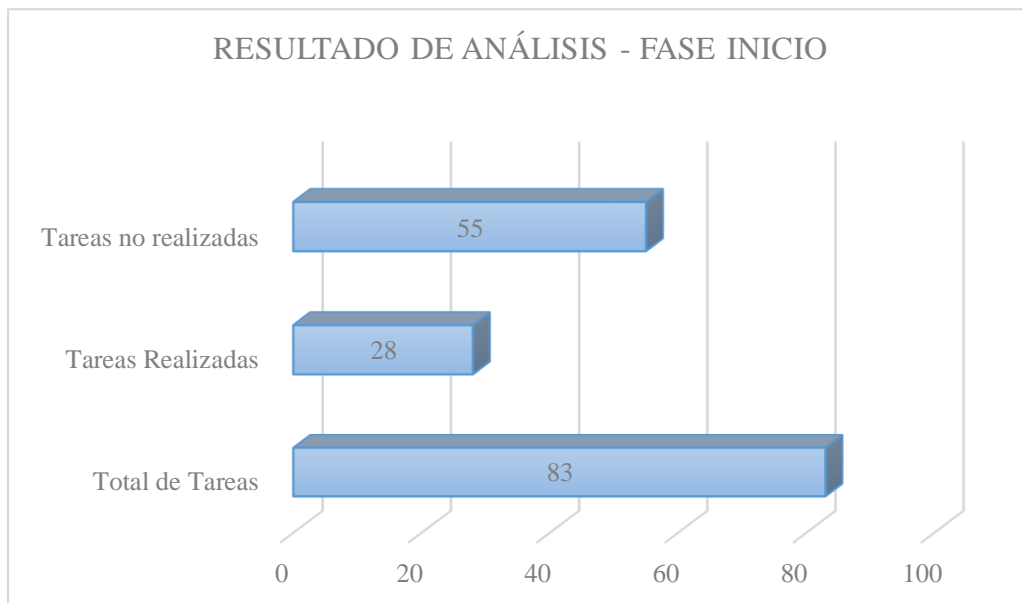


Figura 40 Gráfica de resultados de análisis - Fase inicio

Fuente: La Investigación

Elaborado por: El autor

Conclusión:

En la gráfica resumen de la fase de inicio identificamos que del total de tareas que se deben desarrollar siguiendo la metodología RUP, el área de desarrollo solo ejecuta 28 actividades correspondiente al 33.73% y las 55 actividades restantes que corresponde al 66.26% no las ejecuta, esto porque los encargados del proceso de desarrollo toman como

referencia la metodología RUP para documentar el proyecto, pero los documentos generados son a criterio de los encargados.

B) FASE DE ELABORACIÓN

A continuación, se presenta la tabla de análisis de la fase de Elaboración

RUP - FASE ELABORACIÓN			
ACTIVIDAD	SUBACTIVIDAD	TAREA	DDTI
Preparar el entorno para una iteración	Preparar el entorno para una iteración	Desarrollar el guion de desarrollo	N
		Preparar directrices para el proyecto	S
		Desarrollar la guía de estilo del manual	N
		Preparar plantillas para el proyecto	N
		Iniciar el proceso de desarrollo	S
		Configurar las herramientas	S
		Verificar la instalación y configuración de herramientas	S
Revisar y completar los planes del proyecto	Planificar el proyecto	Desarrollar el plan de medidas	N
		Desarrollar el plan de gestión de riesgos	N
		Desarrollar el plan de aceptación del producto	N
		Desarrollar el plan de resolución de problemas	N
		Desarrollar el plan de garantía de calidad	N
		Definir la organización y el personal del proyecto	S
		Definir los procesos de supervisión y control	N
		Planificar fases e iteraciones	N
		Compilar el plan de desarrollo de software	N
		Revisión de planificación del proyecto	N
	Planificar la configuración del proyecto y el control de cambios	Establecer políticas de gestión de la configuración (CM)	N
		Escribir el plan de gestión de la configuración (CM)	N
		Establecer el proceso de control de cambios	N
	Planificar la integración	Planificar la integración	N
	Planear el despliegue	Desarrollar el plan de despliegue	N
Definir la lista de materiales		N	
Gestión y soporte continuos	Supervisar la iteración	Adquirir personal	N
		Iniciar la iteración	N

		Revisión de los criterios de evaluación de la iteración	N
		Identificar y valorar los riesgos	N
		Valorar la iteración	N
		Revisión de la aceptación de la iteración	N
	Supervisar y controlar el proyecto	Planificar y asignar trabajo	S
		Supervisar el estado del proyecto	S
		Informe de estado	N
		Manejar las excepciones y los problemas	N
		Revisión del proyecto con la autoridad de revisión de proyectos (PRA)	N
		Organizar la revisión	N
		Dirigir la revisión	N
	Gestionar cambios de requisitos	Estructurar el modelo de guion de uso	N
		Gestionar las dependencias	S
		Revisar requisitos	S
	Gestionar solicitudes de cambio	Enviar una solicitud de cambio	N
		Actualizar solicitud de cambio	N
		Revisar solicitudes de cambio	N
		Confirmar las CR duplicadas o rechazadas	N
		Planificar y asignar trabajo	S
		Verificar cambios en la compilación	N
	Dar soporte al entorno durante una iteración	Dar soporte al desarrollo	S
	Cambiar y entregar elementos de configuración	Crear espacios de trabajo de desarrollo	S
		Realizar cambios	S
		Entregar los cambios	S
		Actualizar el espacio de trabajo	S
		Crear las líneas base	N
		Promocionar las líneas base	N
	Gestionar líneas base y releases	Crear la unidad de despliegue	N
		Crear las líneas base	N
		Promocionar las líneas base	N
	Supervisar y notificar el estado de la configuración	Informar del estado de la configuración	N
		Realizar la auditoría de configuración	N
Perfeccionar la definición del sistema	Perfeccionar la definición del sistema	Detallar un guion de uso	N
		Desarrollar especificaciones suplementarias	N
		Detallar los requisitos de software	N
Definir una arquitectura candidata	Definir una arquitectura candidata	Definir el contexto del sistema	S
		Análisis de la arquitectura	S
		Análisis de caso de uso	S

		Análisis de la operación	N
		Identificar patrones de seguridad	N
Perfeccionar la arquitectura	Perfeccionar la arquitectura	Identificar mecanismo de diseño	S
		Identificar elementos de diseño	S
		Análisis de la operación	N
		Incorporar elementos de diseño existentes	N
		Estructurar el modelo de implementación	N
		Describir la arquitectura de tiempo de ejecución	S
		Describir la distribución	S
		Revisar la arquitectura	S
		Desarrollar componentes [dentro del ámbito]	Analizar el comportamiento
Análisis de caso de uso	S		
Análisis de la operación	N		
Diseñar la interfaz de usuario	S		
Prototipo de interfaz de usuario	S		
Revisar el diseño	S		
Identificación de servicio	Descomposición de dominios		N
	Modelado de servicio de objetivos		N
	Análisis de activos existentes		N
Componentes de diseño	Diseño de caso de uso		S
	Diseño del subsistema		N
	Diseño de la operación		N
	Diseño de clase		N
	Definir los elementos de comprobabilidad		N
	Diseñar los elementos de comprobabilidad		N
	Diseño de la cápsula		N
	Revisar el diseño		N
Diseñar la base de datos	Diseño de clase		N
	Especificar la migración de datos		S
	Diseño de base de datos		S
	Revisar el diseño		S
Especificación de servicios	Realizar especificación de servicio		N
	Realizar análisis de subsistema		N
	Realizar especificación de componentes		N
Realización de servicios	Decisiones de realización de servicio de documentos		N
	Especificación de componentes (SOA)		N
	Construir prueba de concepto arquitectónica (SOA)		N
	Valorar la viabilidad de la arquitectura de prueba de concepto		N
Implementar componentes	Planificar la integración del subsistema		N

		Implementar elementos de diseño	N
		Implementar elementos de diseño	N
		Implementar los elementos de comprobabilidad	N
		Implementar la prueba de desarrollador	N
		Ejecutar pruebas de desarrollador	S
		Revisar el código	S
Integrar y probar	Verificar el enfoque de prueba	Definir las configuraciones del entorno de prueba	N
		Identificar los mecanismos de comprobabilidad	S
		Definir los elementos de comprobabilidad	N
		Definir detalles de prueba	N
		Implementar la prueba	S
		Implementar el conjunto de aplicaciones de prueba	N
		Obtener confirmación de comprobabilidad	N
	Integrar y validar la compilación	Integrar los subsistemas	S
		Integrar el sistema	S
		Validar la estabilidad de la compilación	S
	Prueba y evaluación [dentro del ámbito]	Probar y evaluar	S
		Alcanzar una misión aceptable	N
		Mejorar los activos de prueba	N
Desarrollar material de soporte [dentro del ámbito]	Desarrollar materiales de soporte	Desarrollar materiales de formación	N
		Desarrollar materiales de soporte	N
		Crear material gráfico del producto	N
		Crear material gráfico del producto	N
	Probar y evaluar	Definir detalles de prueba	N
		Implementar la prueba	N
		Implementar el conjunto de aplicaciones de prueba	N
		Ejecutar el conjunto de aplicaciones de prueba	N
		Analizar las anomalías en la prueba	N
		Estructurar la implementación de la prueba	N
		Identificar ideas de prueba	N
		Determinar resultados de prueba	N
	Planear la siguiente iteración	Desarrollar el plan de iteración	N
		Desarrollar el caso de negocio	N
		Revisión del plan de la iteración	N

Tabla 46: Análisis fase de elaboración

Fuente: (Contraloría General de la República de Costa Rica, 2017)

Elaborado por: El autor

De la siguiente tabla de análisis se presenta el siguiente mapa de resumen

Tareas Realizadas	39
Tareas no realizadas	97
Tareas Totales	136

Tabla 47: Resumen fase de Elaboración

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

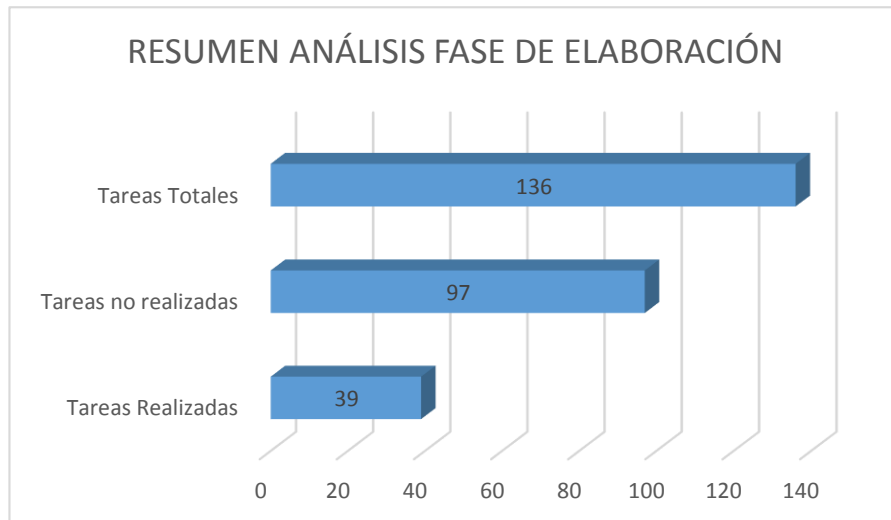


Figura 41: Resumen análisis de fase de elaboración

Fuente: La investigación

Elaborado por: El autor

Conclusión:

En la gráfica resumen de la fase de elaboración, identificamos que del total de tareas que se deben desarrollar siguiendo la metodología RUP, el área de desarrollo solo ejecuta el 39 de actividades el 28.67% y que 97 actividades correspondiente al 71.32% no las ejecuta, esto porque los encargados del proceso de desarrollo toman como referencia la metodología RUP para documentar el proyecto, pero los documentos generados son a criterio de los encargados

C) FASE DE CONSTRUCCIÓN

A continuación, se presenta la tabla de análisis de la fase de Construcción

RUP - FASE CONSTRUCCIÓN			
ACTIVIDAD	SUBACTIVIDAD	TAREA	DDTI
Preparar el entorno para una iteración	Preparar el entorno para una iteración	Desarrollar el guion de desarrollo	N
		Preparar directrices para el proyecto	N
		Desarrollar la guía de estilo del manual	N
		Preparar plantillas para el proyecto	N
		Iniciar el proceso de desarrollo	S
		Configurar las herramientas	S
		Verificar la instalación y configuración de herramientas	S
Gestión y soporte continuos	Supervisar la iteración	Adquirir personal	N
		Iniciar la iteración	N
		Revisión de los criterios de evaluación de la iteración	N
		Identificar y valorar los riesgos	N
		Valorar la iteración	N
		Revisión de la aceptación de la iteración	N
	Supervisar y controlar el proyecto	Planificar y asignar trabajo	N
		Supervisar el estado del proyecto	S
		Informe de estado	N
		Manejar las excepciones y los problemas	N
		Revisión del proyecto con la autoridad de revisión de proyectos (PRA)	N
		Organizar la revisión	N
		Dirigir la revisión	N
	Gestionar cambios de requisitos	Gestionar cambios de requisitos	N
		Gestionar las dependencias	S
		Revisar requisitos	S
	Gestionar solicitudes de cambio	Enviar una solicitud de cambio	N
		Actualizar solicitud de cambio	N
		Revisar solicitudes de cambio	N
		Confirmar las CR duplicadas o rechazadas	N
		Planificar y asignar trabajo	S
		Verificar cambios en la compilación	S
	Dar soporte al entorno durante una iteración	Dar soporte al desarrollo	N
	Cambiar y entregar elementos de configuración	Cambiar y entregar elementos de configuración	N
		Realizar cambios	S
		Entregar los cambios	N

		Actualizar el espacio de trabajo	N	
		Crear las líneas base	N	
		Promocionar las líneas base	N	
	Gestionar líneas base y releases	Crear la unidad de despliegue	N	
		Crear las líneas base	N	
		Promocionar las líneas base	N	
	Supervisar y notificar el estado de la configuración	Informar del estado de la configuración	N	
		Realizar la auditoría de configuración	N	
	Desarrollar componentes [dentro del ámbito]	Analizar el comportamiento	Identificar elementos de diseño	N
			Análisis de caso de uso	S
			Análisis de la operación	N
			Diseñar la interfaz de usuario	S
Prototipo de interfaz de usuario			S	
Revisar el diseño			S	
Identificación de servicio		Descomposición de dominios	N	
		Modelado de servicio de objetivos	N	
		Análisis de activos existentes	N	
Análisis de activos existentes		Diseño de caso de uso	S	
		Diseño del subsistema	S	
		Diseño de la operación	N	
		Diseño de clase	N	
		Definir los elementos de comprobabilidad	N	
		Diseñar los elementos de comprobabilidad	N	
		Diseño de la cápsula	N	
		Revisar el diseño	S	
Diseñar la base de datos		Diseño de clase	N	
		Especificar la migración de datos	S	
		Diseño de base de datos	S	
		Revisar el diseño	S	
Especificación de servicios		Realizar especificación de servicio	N	
		Realizar análisis de subsistema	N	
		Realizar especificación de componentes	S	
Realización de servicios		Decisiones de realización de servicio de documentos	S	
		Especificación de componentes (SOA)	N	
		Construir prueba de concepto arquitectónica (SOA)	N	
		Valorar la viabilidad de la arquitectura de prueba de concepto	N	
Implementar componentes		Planificar la integración del subsistema	S	
		Implementar elementos de diseño	N	
		Analizar el comportamiento en tiempo de ejecución	N	
		Implementar los elementos de comprobabilidad	N	
		Implementar la prueba de desarrollador	S	
		Ejecutar pruebas de desarrollador	S	
		Revisar el código	S	
		Definir las configuraciones del entorno de prueba	S	

Integrar y probar	Verificar el enfoque de prueba	Identificar los mecanismos de comprobabilidad	N		
		Definir los elementos de comprobabilidad	N		
		Definir detalles de prueba	N		
		Implementar la prueba	S		
		Implementar el conjunto de aplicaciones de prueba	N		
		Obtener confirmación de comprobabilidad	N		
	Integrar y validar la compilación	Integrar los subsistemas	N		
		Integrar el sistema	N		
		Validar la estabilidad de la compilación	N		
	Prueba y evaluación [dentro del ámbito]	Probar y evaluar	N		
		Alcanzar una misión aceptable	N		
		Mejorar los activos de prueba	N		
Desarrollar material de soporte [dentro del ámbito]	Desarrollar materiales de soporte	Desarrollar materiales de formación	N		
		Desarrollar materiales de soporte	N		
		Crear material gráfico del producto	N		
		Desarrollar productos de trabajo de instalación	S		
	Probar y evaluar	Definir detalles de prueba	N		
		Implementar la prueba	N		
		Implementar el conjunto de aplicaciones de prueba	N		
		Ejecutar el conjunto de aplicaciones de prueba	N		
		Analizar las anomalías en la prueba	N		
		Estructurar la implementación de la prueba	N		
		Identificar ideas de prueba	S		
		Identificar ideas de prueba	N		
		Planear la siguiente iteración	Planear la siguiente iteración	Planear la siguiente iteración	N
				Desarrollar el caso de negocio	N
Revisión del plan de la iteración	N				
Producir la unidad de despliegue	Producir la unidad de despliegue	Escribir las notas del release	N		
		Escribir las notas del release	N		

Tabla 48: Análisis de la fase de construcción

Fuente: (Contraloría General de la República de Costa Rica, 2017)

Elaborado por: El autor

A continuación, se muestra la tabla de resumen de análisis de la fase de construcción

Tareas Realizadas	29
Tareas no realizadas	77
Tareas Totales	106

Tabla 49: Resumen fase de Construcción

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

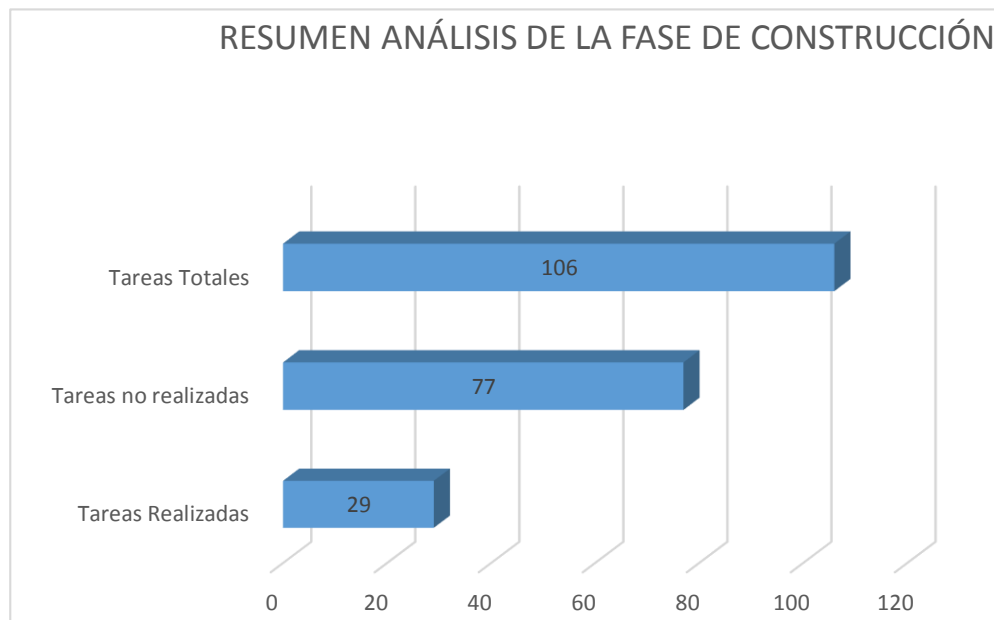


Figura 42: Resumen análisis de la fase de construcción

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

Conclusión:

En la gráfica resumen de la fase de construcción que del total de tareas que se deben desarrollar siguiendo la metodología RUP, el área de desarrollo solo ejecuta 29 actividades correspondiente al 27.35% y las 77 actividades restantes que corresponde al 72.64% no las ejecuta, esto porque los encargados del proceso de desarrollo toman como referencia la metodología RUP para documentar el proyecto, pero los documentos generados son a criterio de los encargados.

D) FASE DE TRANSICIÓN

A continuación, se muestra la tabla de análisis de la fase de transición

RUP - FASE TRANSICIÓN			
ACTIVIDAD	SUBACTIVIDAD	TAREA	DDTI
Preparar el entorno para una iteración	Preparar el entorno para una iteración	Desarrollar el guion de desarrollo	N
		Preparar directrices para el proyecto	N
		Desarrollar la guía de estilo del manual	N
		Preparar plantillas para el proyecto	N
		Iniciar el proceso de desarrollo	S
		Configurar las herramientas	S
		Verificar la instalación y configuración de herramientas	S
Gestión y soporte continuos	Gestión y soporte continuos	Adquirir personal	N
		Iniciar la iteración	N
		Revisión de los criterios de evaluación de la iteración	N
		Identificar y valorar los riesgos	N
		Valorar la iteración	N
		Revisión de la aceptación de la iteración	N
	Revisión de la aceptación de la iteración	Planificar y asignar trabajo	N
		Supervisar el estado del proyecto	N
		Informe de estado	N
		Manejar las excepciones y los problemas	N
		Revisión del proyecto con la autoridad de revisión de proyectos (PRA)	S
		Organizar la revisión	S
		Dirigir la revisión	S
	Gestionar cambios de requisitos	Gestionar cambios de requisitos	N
		Gestionar las dependencias	N
		Gestionar las dependencias	N
	Gestionar solicitudes de cambio	Enviar una solicitud de cambio	N
		Enviar una solicitud de cambio	N
		Revisar solicitudes de cambio	N
		Confirmar las CR duplicadas o rechazadas	N
		Planificar y asignar trabajo	N
		Verificar cambios en la compilación	N
	Dar soporte al entorno durante una iteración	Dar soporte al desarrollo	S
		Cambiar y entregar elementos de configuración	N
	Cambiar y entregar elementos de configuración	Realizar cambios	N
		Entregar los cambios	N
		Actualizar el espacio de trabajo	N

		Crear las líneas base	N
		Promocionar las líneas base	N
	Gestionar líneas base y release	Gestionar líneas base y release	N
		Crear las líneas base	N
		Promocionar las líneas base	N
	Supervisar y notificar el estado de la configuración	Informar del estado de la configuración	N
		Realizar la auditoría de configuración	N
Planear el despliegue	Planear el despliegue	Desarrollar el plan de despliegue	N
		Definir la lista de materiales	N
Arreglar los defectos de los componentes	Arreglar los defectos de los componentes	Planificar la integración del subsistema	S
		Implementar elementos de diseño	N
		Analizar el comportamiento en tiempo de ejecución	N
		Implementar los elementos de comprobabilidad	N
		Implementar la prueba de desarrollador	S
		Ejecutar pruebas de desarrollador	S
		Revisar el código	S
Ejecutar pruebas de desarrollador	Analizar el comportamiento	Identificar elementos de diseño	N
		Análisis de caso de uso	S
		Análisis de la operación	N
		Diseñar la interfaz de usuario	S
		Prototipo de interfaz de usuario	S
		Revisar el diseño	S
	Identificación de servicio	Descomposición de dominios	N
		Modelado de servicio de objetivos	N
		Análisis de activos existentes	N
	Componentes de diseño	Diseño de caso de uso	S
		Diseño del subsistema	N
		Diseño de la operación	N
		Diseño de clase	N
		Definir los elementos de comprobabilidad	N
		Diseñar los elementos de comprobabilidad	N
		Diseño de la cápsula	N
		Revisar el diseño	N
	Diseñar la base de datos	Diseño de clase	N
		Especificar la migración de datos	S
		Diseño de base de datos	S
		Revisar el diseño	S
	Especificación de servicios	Realizar especificación de servicio	N
		Realizar análisis de subsistema	N
		Realizar especificación de componentes	N
	Realización de servicios	Decisiones de realización de servicio de documentos	N
		Especificación de componentes (SOA)	N
		Construir prueba de concepto arquitectónica (SOA)	N

		Valorar la viabilidad de la arquitectura de prueba de concepto	N
	Implementar componentes	Planificar la integración del subsistema	N
		Implementar elementos de diseño	N
		Analizar el comportamiento en tiempo de ejecución	N
		Implementar los elementos de comprobabilidad	N
		Implementar la prueba de desarrollador	N
		Ejecutar pruebas de desarrollador	S
		Revisar el código	S
Integrar y probar		Verificar el enfoque de prueba	Definir las configuraciones del entorno de prueba
	Identificar los mecanismos de comprobabilidad		N
	Definir los elementos de comprobabilidad		N
	Definir detalles de prueba		S
	Implementar la prueba		S
	Implementar el conjunto de aplicaciones de prueba		S
	Obtener confirmación de comprobabilidad		N
			Integrar y validar la compilación
	Integrar el sistema	S	
	Validar la estabilidad de la compilación	N	
	Prueba y evaluación [dentro del ámbito]	Probar y evaluar	S
		Alcanzar una misión aceptable	N
		Mejorar los activos de prueba	S
Desarrollar el material de soporte restante [dentro del ámbito]	Desarrollar materiales de soporte	Desarrollar materiales de formación	N
		Desarrollar materiales de soporte	N
		Crear material gráfico del producto	N
		Desarrollar productos de trabajo de instalación	N
	Probar y evaluar	Definir detalles de prueba	N
		Implementar la prueba	N
		Implementar el conjunto de aplicaciones de prueba	N
		Ejecutar el conjunto de aplicaciones de prueba	N
		Analizar las anomalías en la prueba	N
		Estructurar la implementación de la prueba	N
		Identificar ideas de prueba	N
		Determinar resultados de prueba	N
Gestionar la prueba de aceptación	Gestionar la prueba de aceptación	Gestionar la prueba de aceptación	S
		Dar soporte al desarrollo	S
		Ejecutar el conjunto de aplicaciones de prueba	S
		Determinar resultados de prueba	S
Producto de prueba de versión beta	Producto de prueba de versión beta	Gestionar la prueba de versión beta	S
Crear el producto para el release	Crear el producto para el release	Escribir las notas del release	N
		Crear la unidad de despliegue	N
	Empaquetar el producto	Verificar el producto manufacturado	N

		Enviar a fábrica	N
	Proporcionar acceso a un sitio de descargas	Proporcionar acceso a un sitio de descargas	S
Planear la siguiente iteración	Planear la siguiente iteración	Desarrollar el plan de iteración	N
		Desarrollar el caso de negocio	N
		Revisión del plan de la iteración	N
Concluir el proyecto	Concluir el proyecto	Prepararse para el cierre del proyecto	N
		Revisión de la aceptación del proyecto	N

Tabla 50: Análisis fase de transición

Fuente: (Contraloría General de la República de Costa Rica, 2017)

Elaborado por: El autor

Además, se presenta la siguiente tabla resumen de la fase de transición

Tareas Realizadas	34
Tareas no realizadas	91
Tareas Totales	125

Tabla 51: Resumen análisis fase de transición

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

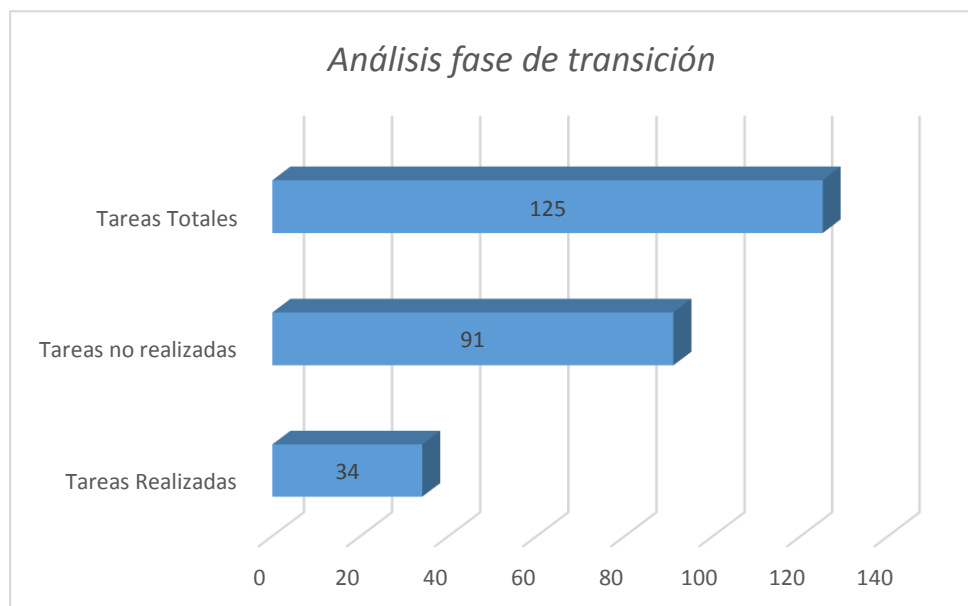


Figura 43: Resumen análisis fase de transición

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

Conclusión:

En la gráfica resumen de la fase de elaboración, identificamos que del total de tareas que se deben desarrollar siguiendo la metodología RUP, el área de desarrollo solo ejecuta el 34 de actividades el 27.2% y que 91 actividades correspondiente al 72.8% no las ejecuta, esto porque los encargados del proceso de desarrollo toman como referencia la metodología RUP para documentar el proyecto, pero los documentos generados son a criterio de los encargados

- **Resumen de análisis**

A continuación, se muestra un resumen del análisis realizado

	Total	Actividades realizadas	%	Actividades no realizadas	%
Fase de Inicio	83	28	33.73%	55	66.27%
Fase de Elaboración	136	39	28.68%	97	71.32%
Fase de construcción	106	29	27.36%	77	72.64%
Fase de Transición	125	34	27.20%	91	72.80%
Total	450	130	28.89%	320	71.11%

Tabla 52 Consolidado análisis de situación actual

Fuente: Análisis y tabulación realizada

Elaborado por: El autor



Figura 44: Consolidado análisis de situación actual

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

Con la finalidad de profundizar en el conocimiento y el grado de aplicación de un proceso de desarrollo se realizó el análisis de los roles que la Dirección de Desarrollo Tecnológico e Informático utiliza para los proyectos.

ANÁLISIS DE ROLES SCRUM

Según análisis realizado basado en la metodología RUP, se obtuvo los roles y se estableció una comparación entre las actividades que se desarrollan en la Dirección de Desarrollo Tecnológico e Informático

ROLES DEL RUP			DD TI
Analista	Analista de sistemas	Buscar actores y guiones de uso	N
		Crear un vocabulario común	N
		Definir el contexto del sistema	S
		Desarrollar el plan de gestión de requisitos	N
		Desarrollar especificaciones suplementarias	N
		Desarrollar la visión	S
		Estructurar el modelo de guión de uso	N
		Gestionar las dependencias	S
		Obtener las solicitudes del interesado	N
	Business Architect	Business Architectural Analysis	S
		Construct Business Architectural Proof-of-Concept	S
		Functional Area Analysis	N
		Prioritize Business Use Cases	N
	Business Designer	Business Operation Analysis	N
		Business Operation Design	S
		Business Use-Case Analysis	S
		Define Automation Requirements	S
		Detail a Business Entity	S
		Detail a Business Use Case	S
	Business-Process Analyst	Detail a Business Worker	S
		Assess Target Organization	N
		Capture a Common Business Vocabulary	N
		Define Business System Context	N
		Find Business Actors and Use Cases	N
		Identify Business Goals and KPIs	N
		Maintain Business Rules	N
		Refine a Business Use-Case	S
		Set and Adjust Objectives	N
	Structure the Business Use-Case Model	N	

Desarrolladores	Especificador de requisitos	Detallar los requisitos de software	S
		Detallar un guión de uso	N
	Arquitecto de seguridad	Identificar patrones de seguridad	N
		Arquitecto de software	Análisis de activos existentes
	Análisis de guiones de uso empresariales (SOA)		N
	Análisis de la arquitectura		S
	Análisis de reglas empresariales		S
	Análisis del modelo de datos		S
	Análisis del proceso empresarial		N
	Aplicar pruebas decisivas de servicio		N
	Construir arquitectura de prueba de concepto		N
	Construir prueba de concepto arquitectónica (SOA)		N
	Construir prueba de concepto arquitectónica (SOA)		N
	Describir la distribución		N
	Estructurar el modelo de implementación		N
	Identificar elementos de diseño		N
	Identificar mecanismo de diseño		N
	Incorporar elementos de diseño existentes		N
	Priorizar los guiones de uso		N
	Valorar la viabilidad de la arquitectura de prueba de concepto		N
	Diseñador		Análisis de caso de uso
		Análisis de la operación	N
		Decisiones de realización de servicio de documentos	N
		Diseñar los elementos de comprobabilidad	N
		Diseño de caso de uso	S
		Diseño de clase	S
		Diseño de la operación	N
		Diseño de mensajes	N
		Diseño de subsistema (SOA)	N
		Diseño del subsistema	N
Especificación de componentes (SOA)		N	
Especificación de servicio		N	
Identificar factores comunes y variabilidad		N	
Identificar y asociar servicios con objetivos		N	
Diseñador de base de datos	Diseño de base de datos	S	
	Specify Data Migration	S	
Diseñador de cápsulas	Diseño de la cápsula	N	
Diseñador de interfaz de usuario	Diseñar la interfaz de usuario	S	
	Prototipo de interfaz de usuario	S	
Prototipo de interfaz de usuario	Analizar el comportamiento en tiempo de ejecución	N	

		Desarrollar productos de trabajo de instalación	N
		Ejecutar pruebas de desarrollador	S
		Implementar elementos de diseño	N
		Implementar la prueba de desarrollador	S
		Implementar los elementos de comprobabilidad	N
	Integrador	Crear espacios de trabajo de integración	S
		Crear las líneas base	S
		Integrar el sistema	S
		Integrar los subsistemas	S
		Planificar la integración del sistema	N
		Planificar la integración del subsistema	N
		Promocionar las líneas base	N
	Gestores	Administrador del sistema	Dar soporte al desarrollo
Gestor de configuración		Configurar el entorno de gestión de la configuración (CM)	N
		Crear la unidad de despliegue	N
		Escribir el plan de gestión de la configuración (CM)	N
		Establecer políticas de gestión de la configuración (CM)	N
		Informar del estado de la configuración	N
		Realizar la auditoría de configuración	N
Gestor de control de cambios		Confirmar las CR duplicadas o rechazadas	S
		Establecer el proceso de control de cambios	N
		Revisar solicitudes de cambio	N
Gestor de despliegue		Definir la lista de materiales	N
		Desarrollar el plan de despliegue	N
		Enviar a fábrica	N
		Escribir las notas del release	N
		Gestionar la prueba de aceptación	N
		Gestionar la prueba de versión beta	N
		Proporcionar acceso a un sitio de descargas	N
		Verificar el producto manufacturado	N
Gestor de proyectos		Adquirir personal	N
		Compilar el plan de desarrollo de software	N
		Definir la organización y el personal del proyecto	S
		Definir los procesos de supervisión y control	S
		Desarrollar el caso de negocio	N
		Desarrollar el plan de aceptación del producto	N
		Desarrollar el plan de garantía de calidad	N
		Desarrollar el plan de gestión de riesgos	N
		Desarrollar el plan de iteración	N
		Desarrollar el plan de medidas	N
		Desarrollar el plan de resolución de problemas	N

		Identificar y valorar los riesgos	N
		Informe de estado	N
		Iniciar el proyecto	N
		Iniciar la iteración	N
		Manejar las excepciones y los problemas	N
		Planificar fases e iteraciones	N
		Planificar y asignar trabajo	N
		Prepararse para el cierre de la fase	N
		Prepararse para el cierre del proyecto	N
		Supervisar el estado del proyecto	N
		Valorar la iteración	N
	Gestor de pruebas	Acordar la misión	N
		Identificar motivadores de prueba	N
		Obtener confirmación de comprobabilidad	N
		Valorar y defender la calidad	N
		Valorar y mejorar el esfuerzo de prueba	N
	Revisor de gestión	Revisión de aprobación del proyecto	N
		Revisión de la aceptación de la iteración	N
		Revisión de la aceptación del proyecto	N
		Revisión de los criterios de evaluación de la iteración	N
		Revisión de los objetivos del ciclo de vida	N
		Revisión de planificación del proyecto	N
		Revisión del plan de la iteración	N
		Revisión del proyecto con la autoridad de revisión de proyectos (PRA)	N
Producción y soporte	Administrador del sistema	Dar soporte al desarrollo	S
	Artista gráfico	Crear material gráfico del producto	N
	Desarrollador de cursos	Desarrollar materiales de formación	N
	Escritor técnico	Desarrollar la guía de estilo del manual	N
		Desarrollar materiales de soporte	N
	Especialista en herramientas	Configurar las herramientas	S
		Seleccionar y adquirir herramientas	S
		Verificar la instalación y configuración de herramientas	S
	Ingeniero de proceso	Desarrollar el guión de desarrollo	N
		Iniciar el proceso de desarrollo	S
		Personalizar el proceso de desarrollo para el proyecto	S
Preparar directrices para el proyecto		N	
Preparar plantillas para el proyecto		N	
Roles generales	Coordinador de revisión	Organizar la revisión	N
	Cualquier rol	Actualizar el espacio de trabajo	N
		Actualizar solicitud de cambio	N

		Crear espacios de trabajo de desarrollo	S
		Entregar los cambios	N
		Enviar una solicitud de cambio	N
		Realizar cambios	N
	Interesado	Interesado	N
	Revisor	Dirigir la revisión	N
	Revisor técnico	Review the Business Analysis Model	S
		Review the Business Use-Case Model	S
		Revisar el código	S
		Revisar el diseño	S
Revisar la arquitectura		S	
	Revisar requisitos	S	
Verificadores	Analista de pruebas	Definir detalles de prueba	S
		Definir necesidades de valoración y rastreabilidad	N
		Determinar resultados de prueba	S
		Identificar destinos de prueba	N
		Identificar ideas de prueba	N
		Verificar cambios en la compilación	N
	Diseñador de pruebas	Definir el enfoque de prueba	N
		Definir las configuraciones del entorno de prueba	S
		Definir los elementos de comprobabilidad	N
		Estructurar la implementación de la prueba	N
		Identificar los mecanismos de comprobabilidad	N
	Gestor de pruebas	Acordar la misión	N
Identificar motivadores de prueba		N	
Obtener confirmación de comprobabilidad		N	
Valorar y defender la calidad		N	
Valorar y mejorar el esfuerzo de prueba		N	
Verificador	Analizar las anomalías en la prueba	N	
	Ejecutar el conjunto de aplicaciones de prueba	S	
	Implementar el conjunto de aplicaciones de prueba	N	
	Implementar la prueba	S	

Tabla 53: Tabla de comparación RUP-Roles DDTI

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

A continuación, se presenta una tabla donde se resume el análisis de la tabla anterior sobre los roles que se definen en el proceso de desarrollo de la organización

TOTAL, ACTIVIDADES RUP	182
PROCESO DDTI - UTN	50
PROCESO RUP SIN USAR	132

Tabla 54: Resumen de Roles entre RUP - Proceso DDTI

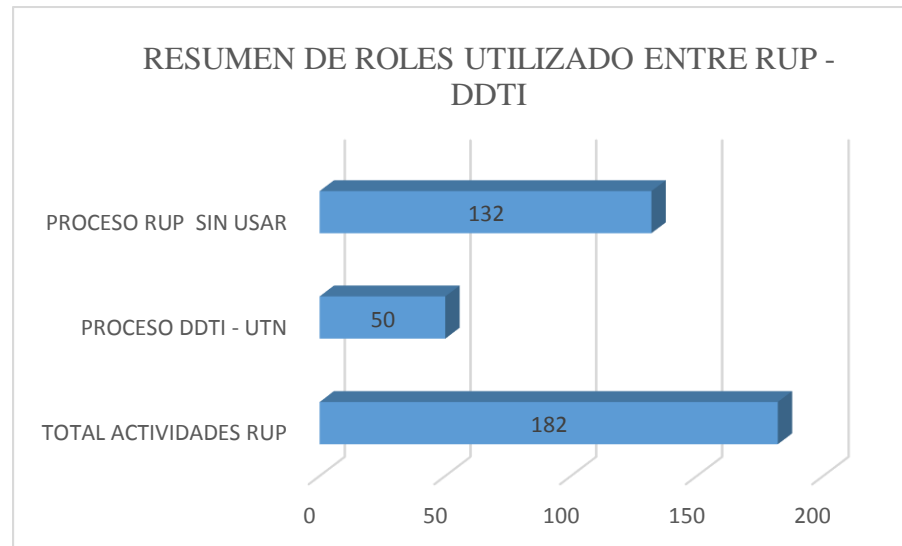


Figura 45: Resumen de uso de Roles

Fuente: Análisis y tabulación realizada

Elaborado por: El autor

Conclusión:

- El proceso de desarrollo de la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte cumple parcialmente las actividades que ejecuta la metodología RUP para el proceso de desarrollo de los proyectos CRM; con la metodología SCRUM se está iniciando su aplicación en proyectos por lo que la documentación que se genera estaba referenciada a los artefactos que solicita la metodología RUP.

METODOLOGÍA SCRUM

La Metodología SCRUM aplicada en la Dirección de Desarrollo Tecnológico e Informático apoya a la gestión del proyecto de software, las interacciones que realiza el desarrollador en la construcción del producto son las mismas que utiliza RUP, con la diferencia de la magnitud del producto.

El análisis se realizó de la metodología SCRUM define 6 actividades a ser desarrolladas en el proceso de desarrollo de Software y estas son:

- **Planificación la Iteración**

Se realiza la planificación de las iteraciones y esta se divide en dos partes que es la planificación del cliente priorizando los requerimientos y la segunda parte que es la planificación del equipo de desarrollo para las actividades a desarrollar en cada sprint.

- **Ejecución de la Iteración**

Se ejecuta en bloques cortos, cada ejecución de un bloque debe dar como resultado un entregable potencial.

- **Reunión diaria de sincronización del equipo**

El objetivo de esta reunión es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad, al poner de manifiesto puntos en que se pueden ayudar unos a otros (ÁGILES, 2017)

- **Demostración de los requerimientos completados**

Se realizan reuniones informales con el cliente para hacer una demostración del avance de los requerimientos.

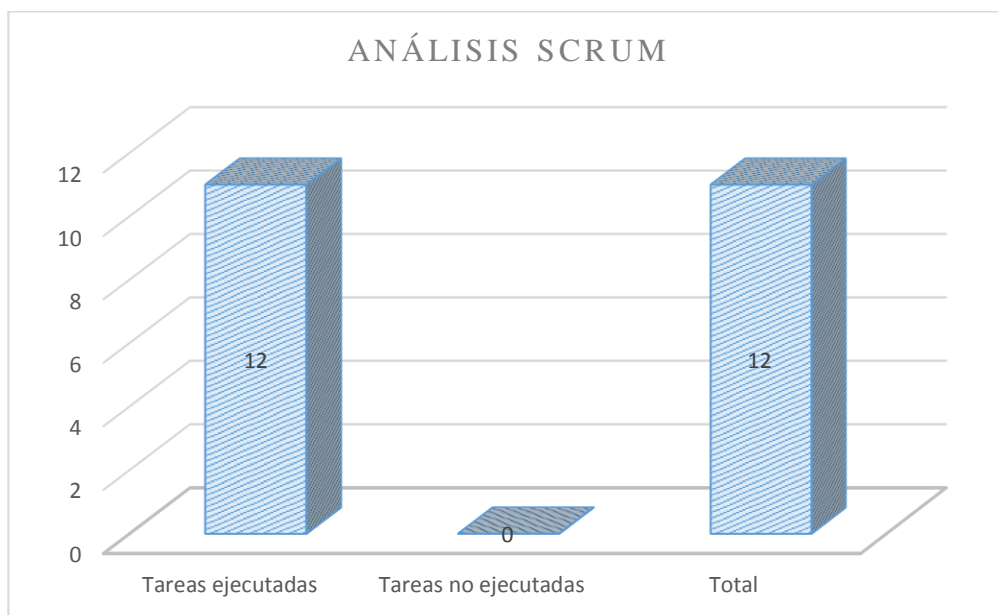
- **Retrospectiva**

De cada reunión se obtiene un feedback para llevar a cabo las mejoras continuas en el proyecto y en los futuros proyecto.

- **Refinamiento de lista de requerimientos y cambios de proyecto**

Se realizan reuniones con el cliente para realizar el refinamiento de los requerimientos e ir trabajando con los objetivos.

Actividad	Tarea	Si	No
Planificación de Iteración	El equipo conoce todos los requerimientos	X	
	El equipo interviene en la planificación de cada iteración	X	
	El equipo realiza la priorización de las actividades	X	
Ejecución de Iteración	Las actividades son ejecutadas de manera independiente	X	
	Las actividades se ejecutan en bloques cortos	X	
	Codificación de Requerimientos	X	
Reuniones diarias	Se realizan reuniones diarias con el equipo	X	
	Las reuniones tienen una duración de entre 15 a 20 min	X	
	El equipo comparte novedades del desarrollo	X	
	Se plantean posibles soluciones	X	
	Se ejecuta un tablero de cumplimiento	X	
Retrospectiva	Se realiza una retroalimentación del proyecto	X	



ANEXO 19: ACTA DE ACEPTACIÓN DE PROYECTO DE INVESTIGACIÓN