

UNIVERSIDAD TÉCNICA DE AMBATO



FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL / DIRECCIÓN DE POSGRADO

MAESTRÍA EN GERENCIA DE SISTEMAS DE INFORMACIÓN

Tema: “ESTANDARIZACIÓN DE LOS PROCESOS DE DESARROLLO DE SOFTWARE UTILIZANDO BUENAS PRÁCTICAS DE PROGRAMACIÓN Y SCRUM COMO MARCO DE TRABAJO ÁGIL EN DEPARTAMENTOS DE TI”

Trabajo de Investigación, previo a la obtención del Grado Académico de
Magister en Gerencia de Sistemas de Información

Autor: Ing. Jorge Vladimir Chávez Andrade

Director: Ing. Darío Javier Robayo Jácome, Mg.

Ambato – Ecuador

2019

A la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

El Tribunal receptor del Trabajo de Investigación presidido por la Ingeniera Elsa Pilar Urrutia Urrutia Magister, e integrado por los señores Ingeniero Víctor Hugo Guachimbosa Villalba Ph.D., Ingeniero Franklin Oswaldo Mayorga Mayorga Magister. designados por la Unidad Académica de Titulación de Posgrado de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, para receptor el Trabajo de Investigación con el tema: “Estandarización de los Procesos de Desarrollo de Software utilizando Buenas Prácticas de Programación y Scrum como Marco de Trabajo Ágil en Departamentos de TI”, elaborado y presentado por el señor Ingeniero Jorge Vladimir Chávez Andrade, para optar por el Grado Académico de Magister en Gerencia de Sistemas de Información; una vez escuchada la defensa oral del Trabajo de Investigación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la UTA.



Ing. Elsa Pilar Urrutia Urrutia, Mg.
Presidenta del Tribunal



Ing. Víctor Hugo Guachimbosa Villalba, PhD.
Miembro del Tribunal



Ing. Franklin Oswaldo Mayorga Mayorga, Mg.
Miembro del Tribunal

AUTORÍA DEL TRABAJO DE INVESTIGACIÓN

La responsabilidad de las opiniones, comentarios y críticas emitidas en el trabajo de investigación presentado con el tema: “Estandarización de los Procesos de Desarrollo de Software utilizando Buenas Prácticas de Programación y Scrum como Marco de Trabajo Ágil en Departamentos de TP”, le corresponde exclusivamente a: Ingeniero Jorge Vladimir Chávez Andrade, autor bajo la dirección del Ingeniero Darío Javier Robayo Jácome Magister, Director del Trabajo de Investigación; y el patrimonio intelectual a la Universidad Técnica de Ambato.



Ing. Jorge Vladimir Chávez Andrade
c.c.: 1803541125
AUTOR



Ing. Darío Javier Robayo Jácome, Mg.
c.c.: 1802842268
DIRECTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que el Trabajo de Investigación, sirva como un documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución.

Cedo los Derechos de mi trabajo, con fines de difusión pública, además apruebo la reproducción de este, dentro de las regulaciones de la Universidad.



Ing. Jorge Vladimir Chávez Andrade
c.c.: 1803541125

ÍNDICE GENERAL

Portada.....	i
A la Unidad Académica de Titulación de la Facultad.....	ii
Autoría del Trabajo de Investigación.....	iii
Derechos de Autor.....	iv
Índice General	v
Índice de Figuras	xi
Índice de Tablas	xii
Agradecimiento.....	xviii
Dedicatoria	xix
Resumen Ejecutivo.....	xx
Executive Summary	xxii
Introducción	1
CAPÍTULO I	3
1 El problema de investigación.....	3
1.1 Tema de Investigación	3
1.2 Planteamiento del Problema.....	3
1.2.1 Contextualización.....	3
1.2.2 Análisis Crítico	5
1.2.3 Prognosis	6
1.2.4 Formulación del Problema	6
1.2.5 Interrogantes (Subproblemas).....	6
1.2.6 Delimitación del objeto de investigación.....	7
1.2.7 Delimitación Espacial:	7
1.2.8 Delimitación Temporal:	7
1.2.9 Unidades de Observación:	7
1.3 Justificación.....	7
1.4 Objetivos	8
1.4.1 Objetivo General	8
1.4.2 Objetivos Específicos:.....	8
CAPITULO II	9

2	Marco Teórico.....	9
2.1	Antecedentes Investigativos.....	9
2.2	Fundamentación Filosófica.....	11
2.3	Fundamentación Legal.....	12
2.4	Categorías Fundamentales.....	19
2.4.1	Categorías de la Variable Independiente.....	20
2.4.1.1	Ingeniería de Software.....	20
2.4.1.2	Proceso de Desarrollo de Software.....	20
2.4.1.3	Buenas prácticas de programación.....	21
2.4.1.3.1	Estándares de documentación.....	21
2.4.1.3.2	Documentación de código.....	22
2.4.1.3.3	Tabulación.....	22
2.4.1.3.4	Control de versiones.....	22
2.4.1.4	Metodologías de desarrollo de software.....	22
2.4.1.5	Metodologías tradicionales.....	23
2.4.1.5.1	Modelo en Cascada o Secuencial.....	23
2.4.1.5.2	Modelo Iterativo e Incremental.....	24
2.4.1.5.3	Modelo en Espiral.....	25
2.4.1.6	Metodologías Ágiles.....	25
2.4.1.6.1	Scrum.....	28
2.4.1.6.2	XP.....	31
2.4.2	Categorías de la Variable Dependiente.....	32
2.4.2.1	Modelos de Gestión de Tecnologías y Sistemas de Información (TSI).....	32
2.4.2.2	Modelos de Calidad del producto software.....	33
2.4.2.3	Funcionalidad.....	35
2.5	Hipótesis.....	38
2.6	Señalamiento de Variables.....	38
	CAPITULO III.....	39
3	Metodología.....	39
3.1	Enfoque.....	39
3.2	Modalidad básica de la investigación.....	39
3.3	Nivel o tipo de investigación.....	39

3.4	Población y Muestra.....	40
3.5	Operacionalización de Variables.....	41
3.5.1	Variable Independiente	41
3.5.2	Variable Dependiente.....	42
3.6	Recolección de Información	43
3.7	Procesamiento y Análisis	44
CAPITULO IV		45
4	Análisis e Interpretación de Resultados	45
4.1	Análisis de resultados.....	45
4.1.1	Formulario de evaluación y medición.....	45
4.2	Validación de las respuestas obtenidas	50
4.2.1	Validación del formulario de evaluación y medición	50
4.2.2	Interpretación del resultado del formulario de evaluación y medición	50
CAPITULO V.....		52
5	Conclusiones y Recomendaciones	52
5.1	Conclusiones	52
5.2	Recomendaciones.....	52
CAPITULO VI		54
6	Propuesta.....	54
6.1	Datos Informativos.....	54
6.1.1	Título.....	54
6.1.2	Institución ejecutara	54
6.1.3	Beneficiarios	54
6.1.4	Ubicación	54
6.1.5	Responsable	54
6.1.6	Director	54
6.2	Antecedentes de la propuesta.....	55
6.3	Justificación.....	55
6.4	Objetivos	55
6.4.1	Objetivo general	55
6.4.2	Objetivos específicos	55
6.5	Análisis de factibilidad.....	56

6.5.1	Factibilidad operativa.....	56
6.5.2	Factibilidad económica	56
6.5.3	Factibilidad técnica	56
6.5.4	Factibilidad legal.....	58
6.6	Fundamentación	58
6.6.1	Entornos de desarrollo integrado IDE's.....	58
6.6.2	Métricas de calidad del producto software para Funcionalidad.....	59
6.7	Elaboración de la propuesta	60
6.7.1	Ficha del proceso.....	60
6.7.2	Alcance del proceso	60
6.7.3	Políticas del proceso.....	61
6.7.4	Diagrama de flujo del proceso	63
6.7.5	Descripción de actividades del proceso	64
6.7.6	Indicadores del proceso.....	66
6.8	Estándares de Desarrollo de Software.....	68
6.8.1	Roles dentro del Equipo Scrum (Scrum Team)	68
6.8.1.1	Interesados (Stakeholders)	68
6.8.1.2	Dueño del producto (Product Owner)	68
6.8.1.3	Scrum Master	68
6.8.1.4	Miembros del equipo de desarrollo (Development Team Members) ...	68
6.8.2	Etapas del desarrollo (Sprint).....	68
6.8.2.1	Planeación del Sprint (Sprint Planning).....	69
6.8.2.2	Reunión de Equipo de Scrum (Scrum team meeting).....	69
6.8.2.3	Refinamiento del Backlog (Backlog Refinement)	69
6.8.2.4	Revisión del Sprint (Sprint Review)	69
6.8.2.5	Retrospectiva del Sprint (Retrospective)	70
6.8.3	Herramientas de Scrum	70
6.8.3.1	Backlog de Producto (Product Backlog).....	70
6.8.3.2	Historias de Usuario (User Stories)	70
6.8.3.3	Backlog del Sprint (Sprint Backlog)	70
6.8.3.4	El panel de Tareas (The Taskboard)	70
6.8.4	Buenas prácticas de programación.....	71

6.8.4.1	Estándares de documentación	71
6.8.4.1.1	Documentación técnica.....	71
6.8.4.1.1.1	Formato	71
6.8.4.1.1.2	Ubicación	71
6.8.4.1.2	Estructura directorios.....	71
6.8.4.1.3	Nombres de objetos	72
6.8.4.1.4	Archivos.....	72
6.8.4.2	Documentación de Código.....	72
6.8.4.3	Tabulación.....	73
6.8.4.4	Control de versiones.....	73
6.8.5	Anexos propuesta.....	74
6.8.5.1	Anexo A	74
6.8.5.2	Anexo B	74
6.8.5.3	Anexo C	75
6.8.5.4	Anexo D	75
6.9	Modelo Operativo	76
6.9.1	Capacitación.....	76
6.9.2	Roles dentro del Equipo Scrum.....	76
6.9.3	Análisis del sistema.....	77
6.9.3.1	Funcionalidades.....	77
6.9.4	Backlog del Producto	77
6.9.4.1	Historias de Usuario.....	77
6.9.4.2	Tareas de Ingeniería	80
6.9.5	Planeación del Sprint.....	88
6.9.6	Reunión de Equipo de Scrum.....	89
6.9.7	Refinamiento del Backlog.....	89
6.9.8	Revisión del Sprint.....	90
6.9.9	Retrospectiva del Sprint	90
6.9.10	Buenas prácticas de programación aplicadas al proyecto	92
6.9.10.1	Documentación técnica.....	92
6.9.10.2	Documentación de código	92
6.9.10.3	Tabulación	93

6.9.10.4	Control de versiones	94
6.9.11	Diseño de la Base de Datos	95
6.9.12	Pruebas	95
6.9.13	Indicadores de calidad.....	96
CAPITULO VII.....		97
7	Análisis de Impactos	97
7.1	Introducción	97
7.2	Análisis de resultados.....	97
7.2.1	Encuesta a usuarios, personal técnico y funcionarios	97
7.2.2	Formulario de evaluación y medición.....	106
7.3	Validación de las respuestas obtenidas	109
7.3.1	Validación a la encuesta a usuarios, personal técnico y funcionarios.	109
7.3.2	Interpretación del resultado del formulario de evaluación y medición	110
7.4	Verificación de la hipótesis.....	110
7.5	Verificación estadística de la hipótesis	110
7.5.1	Planteamiento de la hipótesis	111
7.5.2	Cálculo de Chi Cuadrado de Pearson.....	111
7.6	Datos estadísticos de funcionalidad como indicador de calidad.....	112
7.6.1	Resultados	114
Bibliografía		115
Anexos		120
Anexo 1		120
Anexo 2		122
Anexo 3		124
Anexo 4.....		140

ÍNDICE DE FIGURAS

Figura 1. Inclusiones Conceptuales.....	19
Figura 2. Constelación de Ideas de la Variable Independiente	19
Figura 3. Constelación de Ideas de la Variable Dependiente.....	20
Figura 4. Modelo de desarrollo de software en Cascada o Secuencial	23
Figura 5. Modelo de desarrollo de software Incremental.	24
Figura 6. Modelo de desarrollo de software en Espiral	25
Figura 7. Modelo de desarrollo de software Ágil	25
Figura 8. Modelo de gobierno y gestión de las TIC propuesto por AENOR.....	32
Figura 9. Proceso de Gestión de Desarrollo de Software utilizando Scrum	63
Figura 10. Estructura de directorios desarrollo interno.....	72
Figura 11. TFS. Panel de tareas: Trabajo pendiente del Sprint 1.....	91
Figura 12. TFS. Panel de tareas: Detalle del Sprint 2	91
Figura 13. Documentación en la definición de variables	92
Figura 14. Documentación en una función	92
Figura 15. Documentación en la base de datos	93
Figura 16. Tabulación de código.....	93
Figura 17. Control de versiones TFVC	94
Figura 18. Diagrama de la base de datos	95
Figura 19. Aseguramiento del cumplimiento de las necesidades del cliente.....	98
Figura 20. Reducción de errores en el sistema en producción.....	99
Figura 21. Mejorar la confianza y satisfacción del cliente	100
Figura 22. Aprendizaje y operabilidad.....	101
Figura 23. Protección contra errores del usuario	102
Figura 24. Estética de la interfaz de usuario	103
Figura 25. Utilidad del sistema	104
Figura 26. Mejores prácticas	105

ÍNDICE DE TABLAS

Tabla 1. Tiempos planificados en proyectos de desarrollo DITIC-UTA.....	5
Tabla 2. Constitución de la República del Ecuador	12
Tabla 3. Ley Orgánica de Educación Superior	12
Tabla 4. Normativas de Control Interno de la Contraloría General del Estado	13
Tabla 5. Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación	14
Tabla 6. Estatuto de la Universidad Técnica de Ambato	15
Tabla 7. Reglamento Orgánico de Gestión Organizacional por Procesos de la Universidad Técnica de Ambato	16
Tabla 8. Reglamento de la Dirección de Tecnología de Información y Comunicación -DITIC- de la Universidad Técnica de Ambato.....	16
Tabla 9. Reglamento del Sistema Integrado de Información SI-UTA.....	18
Tabla 10. Etapas del proceso de desarrollo de software	21
Tabla 11. Manifiesto para el desarrollo ágil de software	26
Tabla 12. Valores de Scrum	29
Tabla 13. Roles de Scrum	29
Tabla 14. Eventos del Sprint	30
Tabla 15. Artefactos de Scrum.....	31
Tabla 16. Valores y principios de XP	31
Tabla 17. Modelos relacionados con la evaluación del producto software.....	33
Tabla 18. Características y subcaracterísticas de calidad según ISO/IEC 25010 .	34
Tabla 19. Modelos y estándares de calidad del producto software.....	35
Tabla 20. Muestra de estudio	40
Tabla 21. Variable Independiente: Metodologías de desarrollo de software.....	41
Tabla 22. Variable Dependiente: Funcionalidad.....	42
Tabla 23. Recolección de la información.....	43
Tabla 24. Metodología de desarrollo	45
Tabla 25. Documentación de código.....	46
Tabla 26. Control de cambios sobre el software	46
Tabla 27. Documento de especificación de requerimientos.....	46

Tabla 28. Tabulación de métricas de calidad para adecuación funcional	49
Tabla 29. Resultado del formulario de evaluación y medición.....	50
Tabla 30. Especificaciones técnicas Equipos de desarrollo	56
Tabla 31. Especificaciones técnicas Servidor de pruebas y versionamiento	56
Tabla 32. Especificaciones técnicas Servidor de base de datos de pruebas.....	57
Tabla 33. Especificaciones técnicas Servidor de base de datos	57
Tabla 34. Especificaciones técnicas Servidor espejo.....	57
Tabla 35. Características de los Entornos de desarrollo integrado (IDE's).....	58
Tabla 36. Métricas de calidad para adecuación funcional	59
Tabla 37. Ficha del Proceso de Gestión de Desarrollo de Software utilizando Scrum	60
Tabla 38. Descripción de actividades del proceso	64
Tabla 39. Indicadores del proceso.....	66
Tabla 40. Entornos de desarrollo integrado (IDE's) considerados para el desarrollo	73
Tabla 41. Gestores de versionamiento	73
Tabla 42. Formato Retrospectiva del Sprint	74
Tabla 43. Formato Historias de usuario	75
Tabla 44. Formato Tarea de ingeniería	75
Tabla 45. Roles dentro del Equipo Scrum	76
Tabla 46. Historia de usuario: Acceso al Sistema de Acciones de Personal.....	77
Tabla 47. Historia de usuario: Plantilla.....	78
Tabla 48. Historia de usuario: Asignación a los perfiles de usuarios	78
Tabla 49. Historia de usuario: Parametrización de datos iniciales para Acciones de personal	78
Tabla 50. Historia de usuario: Gestionar las Acciones de Personal.....	79
Tabla 51. Historia de usuario: Revisión y Aprobación	79
Tabla 52. Historia de usuario: Reportes	79
Tabla 53. Historia de usuario: Integración del Sistema de acción de personal con el SIUTA.....	79
Tabla 54. Tarea de ingeniería. Ingreso Exitoso.....	80
Tabla 55. Tarea de ingeniería. Ingreso fallido	80

Tabla 56. Tarea de ingeniería. Bloqueo	80
Tabla 57. Tarea de ingeniería. Control URL Autorizado	80
Tabla 58. Tarea de ingeniería. Sesión Expirada.....	81
Tabla 59. Tarea de ingeniería. Creación de hojas de estilo.....	81
Tabla 60. Tarea de ingeniería. Creación de Imágenes	81
Tabla 61. Tarea de ingeniería. Creación Plantilla HTML.....	81
Tabla 62. Tarea de ingeniería. Integrar plantilla al aplicativo y verificación del diseño	82
Tabla 63. Tarea de ingeniería. Creación de usuarios	82
Tabla 64. Tarea de ingeniería. Eliminación de usuarios	82
Tabla 65. Tarea de ingeniería. Asignación de Perfiles.....	82
Tabla 66. Tarea de ingeniería. Acceso a gestionar diferentes tipos de acción de personal	82
Tabla 67. Tarea de ingeniería. Asignación de carreras y departamentos.....	83
Tabla 68. Tarea de ingeniería. Gestión del estado de la acción de personal.....	83
Tabla 69. Tarea de ingeniería. Denegar accesos a usuarios - roles.....	83
Tabla 70. Tarea de ingeniería. Verificación de perfiles de usuario	83
Tabla 71. Tarea de ingeniería. Tipo de acciones de personal	84
Tabla 72. Tarea de ingeniería. Autorizado por	84
Tabla 73. Tarea de ingeniería. Solicitado por	84
Tabla 74. Tarea de ingeniería. Grado Académico.....	84
Tabla 75. Tarea de ingeniería. Firmas.....	84
Tabla 76. Tarea de ingeniería. Gestión automática de campos.....	85
Tabla 77. Tarea de ingeniería. Creación de acciones personal	85
Tabla 78. Tarea de ingeniería. Actualización y Validación de la Acción Personal	85
Tabla 79. Tarea de ingeniería. Carga masiva del distributivo de trabajo.....	85
Tabla 80. Tarea de ingeniería. Revisión y aprobación.....	86
Tabla 81. Tarea de ingeniería. Firma y Digitalización.....	86
Tabla 82. Tarea de ingeniería. Anulación	86
Tabla 83. Tarea de ingeniería. Reporte de distributivo.....	86
Tabla 84. Tarea de ingeniería. Reporte de Contratos - Horas.....	87

Tabla 85. Tarea de ingeniería. Reporte de Contratación.....	87
Tabla 86. Tarea de ingeniería. Reporte Formación Docente Terminado.....	87
Tabla 87. Tarea de ingeniería. Reporte Formación Docente en Curso	87
Tabla 88. Tarea de ingeniería. Reporte de Funcionarios	87
Tabla 89. Tarea de ingeniería. Creación de roles y asignación de Active Directory	88
Tabla 90. Funcionalidad: Seguridades y Parametrización	88
Tabla 91. Funcionalidad: Gestión de las Acciones de Personal	88
Tabla 92. Funcionalidad: Reportería e Integración con el SI-UTA	89
Tabla 93. Retrospectiva del Sprint 1	90
Tabla 94. Retrospectiva del Sprint 2	91
Tabla 95. Adecuación funcional del sistema de Acciones de Personal	96
Tabla 96. Aseguramiento del cumplimiento de las necesidades del cliente	98
Tabla 97. Reducción de errores en el sistema en producción	99
Tabla 98. Mejorar la confianza y satisfacción del cliente.....	100
Tabla 99. Aprendizaje y operabilidad	101
Tabla 100. Protección contra errores del usuario.....	102
Tabla 101. Estética de la interfaz de usuario.....	103
Tabla 102. Utilidad del sistema.....	104
Tabla 103. Mejores prácticas	105
Tabla 104. Resumen de casos de la encuesta a usuarios, personal técnico y funcionarios.....	109
Tabla 105. Estadística de fiabilidad de la encuesta a usuarios, personal técnico y funcionarios.....	109
Tabla 106. Chi-cuadrado de Pearson para la pregunta 8.....	112
Tabla 107. Resumen de procesamiento de casos	113
Tabla 108: Tabulación cruzada: Tipo de metodología * Funcionalidad.....	113
Tabla 109. Prueba del Chi Cuadrado de Pearson para Tipo de metodología * Funcionalidad.....	113
Tabla 110. Adecuación funcional del sistema de Estructura Académica Universitaria.....	124
Tabla 111. Adecuación funcional del sistema de Distributivos.....	124

Tabla 112. Adecuación funcional del sistema de UTAm@tico Docentes	125
Tabla 113. Adecuación funcional del sistema de Estudiantes Matriculas	125
Tabla 114. Adecuación funcional del sistema de Estudiantes Calificaciones	126
Tabla 115. Adecuación funcional del sistema de Adecuación funcional del sistema de	126
Tabla 116. Adecuación funcional del sistema de Tutorías docentes	127
Tabla 117. Adecuación funcional del sistema de Seguimiento al sílabo	127
Tabla 118. Adecuación funcional del sistema de Titulados.....	128
Tabla 119. Adecuación funcional del sistema de Seguimiento a estudios de posgrado	128
Tabla 120. Adecuación funcional del sistema de Posgrados	129
Tabla 121. Adecuación funcional del sistema de Becas	129
Tabla 122. Adecuación funcional del sistema de Matriculas Estudiantes Cultura Física e Idiomas.....	130
Tabla 123. Adecuación funcional del sistema de Información Académica.	130
Tabla 124. Adecuación funcional del sistema de Hoja de Vida de Funcionarios	131
Tabla 125. Adecuación funcional del sistema de Plan Estratégico de desarrollo institucional.....	131
Tabla 126. Adecuación funcional del sistema de Infraestructura	132
Tabla 127. Adecuación funcional del sistema de Vacaciones	132
Tabla 128. Adecuación funcional del sistema de Convenios.....	133
Tabla 129. Adecuación funcional del sistema de Reportes Información estratégica	133
Tabla 130. Adecuación funcional del sistema Business Intelligence	134
Tabla 131. Adecuación funcional del sistema de Control Docente	134
Tabla 132. Adecuación funcional del sistema Evalúate.....	135
Tabla 133. Adecuación funcional del sistema de Roles de pago	135
Tabla 134. Adecuación funcional del sistema de Facturación electrónica	136
Tabla 135. Adecuación funcional del sistema de Simulador de acreditación.....	136
Tabla 136. Adecuación funcional del sistema de Seguimiento a graduados	137
Tabla 137. Adecuación funcional del sistema de Bolsa de empleo	137
Tabla 138. Adecuación funcional del sistema de Practicas preprofesionales.....	138

Tabla 139. Adecuación funcional del sistema de Proyectos de vinculación	138
Tabla 140. Adecuación funcional del sistema de Seguimiento de proyectos de Investigación.	139
Tabla 141. Resumen formulario de evaluación y medición.....	140

AGRADECIMIENTO

Indispensable agradecer a mi amada esposa y mis hermosas hijas por su comprensión y apoyo incondicional.

Mi gratitud a funcionarios, directores y profesores de la universidad quienes ayudaron a concluir el presente trabajo.

Al Ing. Darío Robayo, director del proyecto de investigación, por su guía y conocimiento para el desarrollo del mismo.

Jorge

DEDICATORIA

Este trabajo esta respetuosamente dedicado a mi querida esposa e hijas. Quienes motivaron y crearon un amor incondicional en mi vida, sin ellas nada hubiese sido igual.

Jorge

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
MAESTRÍA EN GERENCIA DE SISTEMAS DE INFORMACIÓN

TEMA:

“Estandarización de los Procesos de Desarrollo de Software utilizando Buenas Prácticas de Programación y Scrum como Marco de Trabajo Ágil en Departamentos de TI”

AUTOR: Ing. Jorge Vladimir Chávez Andrade
DIRECTOR: Ing. Darío Javier Robayo Jácome, Mg.
FECHA: 17 de noviembre de 2018

RESUMEN EJECUTIVO

El presente proyecto de investigación busca estandarizar el proceso de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil en departamentos de TI como la Dirección de Tecnología de Información y Comunicación (DITIC) de la Universidad Técnica de Ambato, con el objetivo de incrementar el grado de funcionalidad del producto software con la aplicación de la propuesta elaborada.

La carente aplicación metodológica o marco de trabajo colaborativo en el desarrollo de sistemas de información, conlleva a obtener un producto software de baja calidad, software no utilizado o que no satisfacen las necesidades y requerimientos de los usuarios.

En primera instancia, la aplicación de un marco de trabajo ágil como Scrum logra la integración del usuario al proceso de desarrollo, así como el comprometimiento del equipo de desarrollo para obtener un producto software de calidad que cumpla las aspiraciones del usuario que, mediante la utilización de buenas prácticas de

programación permitirá un mejor entendimiento de las funcionalidades codificadas, integración de los programadores, manejo del código fuente en el tiempo y control ante eventualidades o actualizaciones.

Las normas y estándares internacionales establecidos dentro de la familia ISO/IEC 25000 brindan un conjunto de características y subcaracterísticas de calidad para evaluar el grado de funcionalidad del producto software.

La propuesta de investigación se ha validado mediante la utilización de métricas de calidad pertenecientes a la norma internacional ISO/IEC 25023 relacionadas a funcionalidad, evaluando productos software desarrollados por la DITIC-UTA que utilizaron metodologías de desarrollo tradicionales en comparación con un desarrollo de un producto software aplicado en la propuesta. De igual manera, se identificó la percepción del incremento de funcionalidad como atributo de calidad por parte de los involucrados e interesados del proyecto.

Descriptor: Desarrollo de software, buenas prácticas de programación, Scrum, metodologías de desarrollo tradicionales, metodologías de desarrollo ágiles, producto software, ISO/IEC 25000, ISO/IEC 25023, funcionalidad, calidad.

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
MAESTRÍA EN GERENCIA DE SISTEMAS DE INFORMACIÓN

THEME:

“Software Development Processes Standardization using Good Programming Practices and Scrum as an Agile Framework for IT Departments”

AUTHOR: Ing. Jorge Vladimir Chávez Andrade
DIRECTED BY: Ing. Dario Javier Robayo Jácome, Mg.
DATE: Nov 17st, 2018

EXECUTIVE SUMMARY

This project research seeks to standardize software development process through the use of good programming practices and Scrum as an agile framework for IT departments such as Information Technology and Communication Department (DITIC) belonging to “Universidad Técnica de Ambato” with the aim of increase the software product functionality extent by means of the elaborated proposal application.

The lacking methodological application or collaborative framework in the information systems development leads to acquire a low quality software product, that isn't used or that doesn't meet the needs and requirements of users.

Firstly, the agile framework application such as Scrum achieves user integration to the development process, as well as the development staff implication to obtain a quality software product that meets the user aspirations, through the use of good programming practices will allow a better understanding of coded functionalities, programmers integration, source code management over time and control due to eventualities or updates.

International norms and standards established within ISO/IEC 25000 family provides a set of characteristics and sub-characteristics related to quality to evaluate the software product functionality extent.

The research proposal has been validated through the use of quality metrics belonging to the international standard ISO/IEC 25023 related to functionality evaluating software products developed by DITIC-UTA that in the first instance used traditional development methodologies compared to a development of a software product applied in the proposal. Equally, the increase functionality perception as quality attribute was identified by the involved and interested parts of this project.

Keywords: Software development, good programming practices, Scrum, traditional development methodologies, agile development methodologies, software product, ISO/IEC 25000, ISO/IEC 25023, functionality, quality.

INTRODUCCIÓN

Con el objetivo de encaminar los productos y servicios a un nivel de excelencia, en las instituciones ágiles los mismos empleados se auto organizan y construyen conocimiento (Monte Galiano, 2016).

Actualmente, la Dirección de Tecnología de Información y Comunicación (DITIC) de la Universidad Técnica de Ambato realiza un trabajo empírico, utilizando metodologías tradicionales, pero carece de estándares de programación, guías o procedimientos para el desarrollo de software.

Para garantizar un producto software de calidad desarrollados por los departamentos de TI como la DITIC-UTA, es necesario identificar los problemas que se presentan actualmente; con ello, plantear una alternativa que proponga la solución en el ámbito de las metodologías o marcos de trabajo ágiles.

La presente investigación plantea determinar políticas y prácticas adecuadas para el área de desarrollo de software, apalancados en metodologías o marcos de trabajo que estén enmarcados en el Manifiesto Ágil. Para ello es necesario definir normas, precisar estilos de escritura de código, establecer técnicas de planificación y seguimiento a proyectos de desarrollo de software, entre otros aspectos.

Al finalizar, el marco de trabajo propuesto será implementado con un proyecto de desarrollo de software, en contraparte de los proyectos que aplicaron una metodología tradicional. De esta forma, al determinar políticas y procedimientos adecuados para el área de desarrollo de software, utilizando buenas prácticas de programación con metodologías ágiles, se evaluará qué marco de trabajo es el adecuado para instituciones como la estudiada, determinando si debería cambiarse o no la forma de desarrollo que se lleva actualmente.

Con estos antecedentes, el presente documento se encuentra estructurado como se indica a continuación:

El CAPÍTULO I, EL PROBLEMA contiene: el tema de investigación, el planteamiento del problema, su contexto, análisis crítico, prognosis, formulación del problema, interrogantes, delimitación, justificación y objetivos.

El CAPÍTULO II MARCO TEÓRICO contiene: antecedentes de la investigación, fundamentación filosófica, fundamentación legal, categorías fundamentales, hipótesis y señalamiento de variables.

El CAPÍTULO III METODOLOGÍA contiene: el enfoque de investigación, modalidad básica de la investigación, nivel o tipo de investigación, población y muestra, operacionalización de variables, plan de recolección de información y plan de procesamiento de la información.

El CAPÍTULO IV ANÁLISIS E INTERPRETACIÓN DE RESULTADOS contiene: análisis de resultados, formulario de evaluación y medición, Validación de las respuestas obtenidas, interpretación del resultado del formulario de evaluación y medición.

El CAPÍTULO V contiene: CONCLUSIONES Y RECOMENDACIONES.

El CAPÍTULO VI PROPUESTA contiene: datos informativos, antecedentes de la propuesta, justificación, objetivos, análisis de factibilidad, elaboración de la propuesta, modelo operativo.

El CAPÍTULO VII ANÁLISIS DE IMPACTOS contiene: Análisis de resultados, validación de las respuestas obtenidas, verificación de la hipótesis, cálculo de Chi Cuadrado de Pearson, datos estadísticos de funcionalidad como indicador de calidad.

CAPÍTULO I

1 EL PROBLEMA DE INVESTIGACIÓN

1.1 Tema de Investigación

Estandarización de los procesos de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil en departamentos de TI.

1.2 Planteamiento del Problema

1.2.1 Contextualización

Los procesos o metodologías de desarrollo de software han cambiado adaptándose a la constante evolución de las tecnologías de información, a la forma de trabajar y planificar los proyectos y, a la eficiencia y calidad del desarrollo.

Las metodologías tradicionales brindan al desarrollo de software un ciclo de vida secuencial bien definido y un marco de trabajo para planificar actividades y responsables. Sin embargo, se generan extensas documentaciones y, por lo general, se prolonga el tiempo de desarrollo. Si no se tienen implementadas buenas prácticas de programación, la falta o cambio de un desarrollador en el proyecto, puede influir en el cumplimiento de la planificación y calidad del producto resultante.

A principios del siglo XXI se publica el Manifiesto Ágil (Beck, et al., 2001), fortaleciendo el enfoque de metodologías como XP (eXtreme Programming) y Scrum como marco de trabajo (Schwaber, 2004). Las metodologías ágiles priorizan la calidad del software sobre la documentación excesiva obteniendo, por lo general, entregables de calidad, con una planificación adaptativa a los cambios que pueden surgir durante el desarrollo.

A nivel internacional el informe Chaos Report, publicado por Standish Group desde 1994, publica periódicamente los resultados del grado de éxito de gestión de proyectos informáticos. El informe del año 2015 ha estudiado unos 50.000 proyectos de todo el mundo concluyendo que el uso de metodologías ágiles dio

como resultado proyectos más exitosos en un 39% frente a un 11% que utilizó la metodología tradicional en cascada (The Standish Group, 2015).

En el Ecuador las empresas de software han asumido modelos, normas y certificaciones para lograr competitividad a nivel internacional, estas adopciones son: tener una estructura organizacional adecuada, sistemas de gestión de calidad y certificaciones internacionales (Palacios & Merchán, 2014).

Por lo general, los centros de datos; llámense también direcciones, jefaturas o departamentos de Tecnologías de Información (TI), son las encargadas de: planificar, elaborar y ejecutar programas informáticos que satisfacen requerimientos específicos institucionales. Estos requerimientos están en constante cambio, por ello es necesario que las direcciones de TI adopten un marco de trabajo que facilite el desarrollo, que permita escribir código limpio y sostenible en el tiempo, que exista un ambiente colaborativo de alto nivel, todo ello para garantizar la calidad e incrementar la productividad en el desarrollo de software.

El Área de Gestión de Desarrollo se encuentra dentro de la Estructura Interna de la Dirección de Tecnología de Información y Comunicación (DITIC) de la Universidad Técnica de Ambato (UTA) (H. Consejo Universitario, 2012), y entre sus atribuciones y responsabilidades está el de análisis, diseño, programación e instalación del sistema integrado de información universitario (H. Consejo Universitario, 2013). Se utiliza una metodología tradicional en cascada, de forma empírica, pero carece de estándares de programación, guías o procedimientos para el desarrollo de software.

1.2.2 Análisis Crítico

Al utilizar metodologías tradicionales de forma empírica, la fecha fin planificada no se cumple, el ciclo de desarrollo se extiende con nuevas especificaciones de requerimientos y muchas veces el proyecto fracasa al no satisfacer las necesidades del usuario o no utilizar el software desarrollado. En el caso de la UTA, la tabla 1 muestra los tiempos planificados y nuevas solicitudes de requerimientos posteriores a la fecha de entrega de los últimos proyectos de desarrollo de la DITIC-UTA.

Tabla 1. Tiempos planificados en proyectos de desarrollo DITIC-UTA

Nombre del Proyecto	Fecha Inicio	Fecha Fin (Planificada)	Observaciones
Sistema de Contratos	07/02/17	11/05/17	Hasta la presente fecha se solicitan varios nuevos requerimientos.
Sistema de Registro y Seguimiento al Sílabo	06/10/16	27/01/17	Varios problemas presentados hasta mediados de 2018, se recomienda 2 o 3 revisiones adicionales
Sistema de Nombramientos	02/09/16	30/11/16	Se plantean nuevos requerimientos
Sistema de Planificación	24/03/16	29/06/16	No se utiliza
Sistema de Tutorías	01/03/16	06/06/16	En marzo de 2017 se encuentran un cambio de normativa y en abril de 2017 se solicitan actualizaciones

Elaborado por: Investigador

A pesar de elaborar una extensa documentación técnica, esta no garantiza que el software desarrollado cumpla todas las expectativas del usuario. Así por ejemplo, el Sistema de Registro y Seguimiento al Sílabo, hasta mediados de 2018, requirió modificaciones que han sido solicitadas por la Comisión de Expertos designados por Consejo Académico Universitario.

Cuando no se aplican buenas prácticas de programación, como no comentar el código, no llevar un control de versiones, no existir una hoja de ruta documentada; al final del desarrollo se obtendrá un código fuente inmanejable, no se podrá regresar a una versión anterior del código, y al efectuarse cambios en el grupo de desarrollo, un nuevo programador no podrá hacerse cargo del desarrollo elaborado.

A pesar de que los proyectos de desarrollo mencionados en la Tabla 1 poseen documentación técnica, ninguno de ellos tiene su código fuente documentado y no llevan ningún sistema de versionamiento.

La falta de capacitación especializada genera desconocimiento de nuevas tecnologías o metodologías que agilicen el proceso de desarrollo y que se observe una mejora en la evaluación de resultados. Tal es el caso del Sistema de Garantías Internas de Calidad (SEGIC) no tiene un soporte integral por la DITIC-UTA por no contar con el personal capacitado en las herramientas utilizadas en el desarrollo del mismo.

1.2.3 Prognosis

De no estandarizar el proceso de desarrollo de software utilizando buenas prácticas de programación, la DITIC-UTA no logrará incrementar el grado de funcionalidad del producto software.

1.2.4 Formulación del Problema

¿Cómo incide la estandarización del proceso de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil en el grado de funcionalidad del producto software desarrollado por la Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato?

1.2.5 Interrogantes (Subproblemas)

- ¿Cuáles son los procesos, políticas, técnicas, procedimientos y buenas prácticas de programación para estandarizar el proceso de desarrollo de software utilizando Scrum como marco de trabajo ágil?
- ¿Cuáles son las métricas de calidad para cumplir la funcionalidad como característica de calidad del producto software?
- ¿Se puede utilizar una metodología ágil en el proceso de desarrollo de software en departamentos de TI como la DITIC-UTA para incrementar el grado de funcionalidad del producto software?

1.2.6 Delimitación del objeto de investigación

Campo: Tecnologías y Sistemas de Información

Área: Ingeniería de Software

Aspecto: Estandarización de Procesos de Desarrollo de Software

1.2.7 Delimitación Espacial:

Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato.

1.2.8 Delimitación Temporal:

Desde mayo a octubre de 2018

1.2.9 Unidades de Observación:

Área de Gestión de Desarrollo

1.3 Justificación

Estudios internacionales demuestran que, anualmente, los resultados de la gestión de proyectos con la utilización de metodologías o marcos de trabajo ágiles tienen un mayor porcentaje de éxito frente a proyectos que utilizan metodologías tradicionales (The Standish Group, 2015).

Un marco de trabajo ágil busca la colaboración de todos los involucrados en el proceso de desarrollo, usuarios y programadores, con el objetivo de brindar un producto software de calidad que satisfice las necesidades y requerimientos del usuario. Así mismo, la utilización de buenas prácticas de programación es fundamental ya que brindaría sostenibilidad al código fuente, así como su reutilización.

Por lo expuesto se observa que el presente proyecto de investigación es de interés para los departamentos de TI como la DITIC-UTA que buscan incrementar la calidad del producto software resultante al proceso de desarrollo.

Factibilidad Técnica

Decimos que técnicamente es factible realizar porque se cuenta con los recursos tecnológicos requeridos, haciendo referencia a la infraestructura, herramientas tecnológicas o software, acceso a datos e información requerida.

Factibilidad Operativa

El presente proyecto es factible operativamente porque cuenta con el apoyo de quienes están al frente de la DITIC-UTA, lo cual permite tener la apertura necesaria con el personal de la universidad para proporcionar la información necesaria y asegura que los resultados del presente proyecto por su beneficio y utilidad sean aplicados.

Factibilidad Económica

Podemos mencionar que económicamente el presente proyecto es factible ya que los costos que implican el análisis, estudio, tiempo empleado en estos temas son asumidos por el investigador.

1.4 Objetivos

1.4.1 Objetivo General

Estandarizar el proceso de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil para incrementar el grado de funcionalidad del producto software en departamentos de TI como la DITIC-UTA.

1.4.2 Objetivos Específicos:

- Identificar los procesos de desarrollo de software y un marco de trabajo ágil utilizando buenas prácticas de programación que mejor se adapte a departamentos de TI como la DITIC-UTA.
- Establecer el grado de funcionalidad del producto software desarrollados.
- Proponer la estandarización del proceso de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil para incrementar el grado de funcionalidad del producto software en departamentos de TI como la DITIC-UTA.

CAPITULO II

2 MARCO TEÓRICO

2.1 Antecedentes Investigativos

Luego de efectuar una revisión y análisis bibliográfico, en el repositorio de la UTA, se encontró que en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial existe un trabajo relacionado, pero desde un enfoque diferente al planteado, el cual citamos a continuación:

Según Pedro David Granda Gudiño (Granda Gudiño, 2016) en la investigación “Análisis de las metodologías ágiles y su incidencia en la creación del portafolio de servicio para la unidad de extensión universitaria de la Universidad Técnica del Norte de la ciudad de Ibarra”, en el año 2016, en la que analiza la incidencia de la implementación de metodologías ágiles en el desarrollo de software; en dicha investigación se menciona:

- Plantear una alternativa a las metodologías tradicionales en la creación de portafolios de servicio, estableciendo al cliente como fuente principal en el desarrollo del mismo.
- Al diagnosticar al personal del Departamento de Desarrollo Informático de la Universidad Técnica del Norte, se logró encontrar una metodología apegada a la realidad de la institución.
- Una de las conclusiones del estudio fue determinar a Scrum y eXtreme Programming como metodología especialmente utilizada en grupos pequeños de desarrollo de software, permitiendo obtener productos de calidad a satisfacción del cliente.

Al realizar la investigación bibliografía en otras universidades ecuatorianas, se encontraron varios trabajos relacionados que servirán de apoyo a la investigación.

Según Cristhian Fabián Cabezas Gutiérrez (Cabezas Gutiérrez, 2016) en la investigación “Estudio cualitativo del impacto de aplicación de buenas prácticas para la administración de proyectos de software en 11 organizaciones del sector público”, trabajo realizado como tesis de la Facultad de Ciencias Administrativas y Contables en la Pontificia Universidad Católica del Ecuador, en el año 2016 plantea:

- Conocer la situación actual de las organizaciones del estado y la gestión de los proyectos de desarrollo de software, identificando si se aplica o no buenas prácticas.
- Realizar una propuesta de integración entre las buenas prácticas y normas de control interno correspondientes a la Dirección de Proyectos Tecnológicos de la Contraloría General del Estado.

Según Paula Natalia Cadena Loayza (Cadena Loayza, 2015) en la investigación “Adaptación de marcos de referencia de calidad a la industria de desarrollo de software en Ecuador”, trabajo de titulación de Magister en Gerencia de Sistemas y Tecnologías de Información de la Universidad de las Américas (Udla), en el año 2015 concluye:

- Para lograr la calidad del software se debe cumplir estándares establecidos que brinden una guía efectiva.
- Marcos de referencia como ISO y CMMI establecen normativas enfocadas a aspectos técnicos de desarrollo de software.

También se citan referencias bibliográficas de trabajos de investigación de universidades de américa latina.

Según Daynel Díaz Polo (Díaz Polo, 2011) en la investigación “Definición de un proceso de desarrollo de software en un entorno universitario”, trabajo de titulación de Título de Máster en Informática Aplicada del Instituto Superior Politécnico José Antonio Echeverría, en el año 2011 señala:

- Para el proceso de desarrollo de software se dispone actualmente de una diversidad de procesos, técnicas y paradigmas, por ello se debe realizar una selección adecuada de la metodología de desarrollo a ser utilizada.
- Según CMMI cuando se define un proceso de desarrollo de software se debe garantizar una herramienta que le de soporte al proceso, agilizando y aumentando la calidad del producto final.

Según Jaime Andrés Britto Montoya (Britto Montoya, 2014) en la investigación “Adaptación de un proceso de desarrollo de software basado en buenas prácticas”, trabajo de titulación de Maestría en Gestión y Desarrollo de Proyectos de Software de la Facultad de Ingeniería de la Universidad Autónoma de Manizales, en el año 2014 concluye:

- Es necesario que las mejores prácticas se adapten a la cultura y necesidades de cada institución.
- El éxito de un proyecto depende de la adaptación a las buenas practicas por las personas, y no solo que el proceso diseñado cumpla todas las especificaciones y requerimientos de CMMI.

2.2 Fundamentación Filosófica

La presente investigación se enmarca en el paradigma Crítico Propositivo (Ramos, 2015); es crítico por que realiza un análisis crítico del problema actual que surge cuando se usa metodologías de desarrollo de software tradicional, y es propositivo porque busca proponer una solución factible al problema, misma que se apalanca en el uso de buenas prácticas de programación y un Scrum como marco de trabajo ágil.

2.3 Fundamentación Legal

El presente trabajo de investigación se sustenta en las siguientes leyes:

CONSTITUCION DE LA REPUBLICA DEL ECUADOR

Tabla 2. Constitución de la República del Ecuador

CONSTITUCION DE LA REPUBLICA DEL ECUADOR	
Artículo	Texto
Art. 355	<p>El Estado reconocerá a las universidades y escuelas politécnicas autonomía académica, administrativa, financiera y orgánica, acorde con los objetivos del régimen de desarrollo y los principios establecidos en la Constitución.</p> <p>Se reconoce a las universidades y escuelas politécnicas el derecho a la autonomía, ejercida y comprendida de manera solidaria y responsable. Dicha autonomía garantiza el ejercicio de la libertad académica y el derecho a la búsqueda de la verdad, sin restricciones; el gobierno y gestión de sí mismas, en consonancia con los principios de alternancia, transparencia y los derechos políticos; y la producción de ciencia, tecnología, cultura y arte.</p>

Fuente: (Asamblea Nacional, 2008)

LEY ORGANICA DE EDUCACION SUPERIOR

Tabla 3. Ley Orgánica de Educación Superior

LEY ORGANICA DE EDUCACION SUPERIOR	
Artículo	Texto
Art. 12.-	<p>Principios del Sistema. - El Sistema de Educación Superior se rige por los principios de autonomía responsable, cogobierno, igualdad de oportunidades, calidad, pertinencia, integralidad, autodeterminación para la producción del pensamiento y conocimiento, en el marco del diálogo de saberes, pensamiento universal y producción científica y tecnológica global.</p> <p>El Sistema de Educación Superior, al ser parte del Sistema Nacional de Inclusión y Equidad Social, se rige por los principios de universalidad, igualdad, equidad, progresividad, interculturalidad, solidaridad y no discriminación; y funcionará bajo los criterios de calidad, eficiencia, eficacia, transparencia, responsabilidad y participación</p> <p>Estos principios rigen de manera integral a las instituciones, actores, procesos, normas, recursos, y demás componentes del sistema, en los términos que establece esta Ley.</p>
Art. 13.-	<p>Funciones del Sistema de Educación Superior. - Son funciones del Sistema de Educación Superior:</p> <p>a) Garantizar el derecho a la educación superior mediante la docencia, la investigación y su vinculación con la sociedad, y asegurar crecientes niveles de calidad, excelencia académica y pertinencia;</p> <p>b) Promover la creación, desarrollo, transmisión y difusión de la ciencia, la técnica, la tecnología y la cultura;</p>

LEY ORGANICA DE EDUCACION SUPERIOR	
Art. 17.-	<p>Reconocimiento de la autonomía responsable. - El Estado reconoce a las universidades y escuelas politécnicas autonomía académica, administrativa, financiera y orgánica, acorde con los principios establecidos en la Constitución de la República.</p> <p>En el ejercicio de autonomía responsable, las universidades y escuelas politécnicas mantendrán relaciones de reciprocidad y cooperación entre ellas y de estas con el Estado y la sociedad: además observarán los principios de justicia, equidad, solidaridad, participación ciudadana, responsabilidad social y rendición de cuentas.</p> <p>Se reconoce y garantiza la naturaleza jurídica propia y la especificidad de todas las universidades y escuelas politécnicas.</p>
Art. 18.-	<p>Ejercicio de la autonomía responsable. - La autonomía responsable que ejercen las universidades y escuelas politécnicas consiste en:</p> <p>e) La libertad para gestionar sus procesos internos;</p> <p>El ejercicio de la autonomía responsable permitirá la ampliación de sus capacidades en función de la mejora y aseguramiento de la calidad de las universidades y escuelas politécnicas. El reglamento de la presente ley establecerá los mecanismos para la aplicación de este principio.</p>

Fuente: (Asamblea Nacional, 2010), (Asamblea Nacional, 2018)

NORMAS DE CONTROL INTERNO DE LA CONTRALORIA GENERAL DEL ESTADO

Tabla 4. Normativas de Control Interno de la Contraloría General del Estado

NORMAS DE CONTROL INTERNO DE LA CONTRALORIA GENERAL DEL ESTADO	
Artículo	Texto
410-06	<p>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:</p> <ol style="list-style-type: none"> 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. 6. El inicio de las etapas importantes del proyecto será aprobado de manera formal y comunicado a todos los interesados. 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.

NORMAS DE CONTROL INTERNO DE LA CONTRALORIA GENERAL DEL ESTADO	
410-07	<p>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:</p> <p>2. Adopción, mantenimiento y aplicación de políticas públicas y estándares internacionales para: codificación de software, nomenclaturas, interfaz de usuario, interoperabilidad, eficiencia de desempeño de sistemas, escalabilidad, validación contra requerimientos, planes de pruebas unitarias y de integración.</p> <p>3. Identificación, priorización, especificación y acuerdos de los requerimientos funcionales y técnicos institucionales con la participación y aprobación formal de las unidades usuarias. Esto incluye, tipos de usuarios, requerimientos de: entrada, definición de interfaces, archivo, procesamiento, salida, control, seguridad, plan de pruebas y trazabilidad o pistas de auditoría de las transacciones en donde aplique.</p> <p>5. En los procesos de desarrollo, mantenimiento o adquisición de software aplicativo se considerarán: estándares de desarrollo, de documentación y de calidad, el diseño lógico y físico de las aplicaciones, la inclusión apropiada de controles de aplicación diseñados para prevenir, detectar y corregir errores e irregularidades de procesamiento, de modo que éste, sea exacto, completo, oportuno, aprobado y auditable. Se considerarán mecanismos de autorización, integridad de la información, control de acceso, respaldos, diseño e implementación de pistas de auditoría y requerimientos de seguridad. La especificación del diseño considerará las arquitecturas tecnológicas y de información definidas dentro de la organización.</p> <p>9. Los derechos de autor del software desarrollado a la medida pertenecerán a la entidad y serán registrados en el organismo competente. Para el caso de software adquirido se obtendrá las respectivas licencias de uso.</p> <p>10. Formalización con actas de aceptación por parte de los usuarios, del paso de los sistemas probados y aprobados desde el ambiente de desarrollo/prueba al de producción y su revisión en la post-implantación.</p> <p>11. Elaboración de manuales técnicos, de instalación y configuración; así como de usuario, los cuales serán difundidos, publicados y actualizados de forma permanente.</p>

Fuente: (Contraloría General del Estado, 2009)

CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN

Tabla 5. Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación

CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN	
Artículo	Texto
Art. 131.-	<p>Protección de software. - El software se protege como obra literaria. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea como código fuente; es decir, en forma legible por el ser humano; o como código objeto; es decir,</p>

CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN	
	en forma legible por máquina, ya sea sistemas operativos o sistemas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa.
Art. 133.-	Titulares de derechos. - Es titular de los derechos sobre un software el productor, esto es, la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se presumirá titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual.

Fuente: (Asamblea Nacional, 2016)

ESTATUTO DE LA UNIVERSIDAD TÉCNICA DE AMBATO

Tabla 6. Estatuto de la Universidad Técnica de Ambato

ESTATUTO DE LA UNIVERSIDAD TÉCNICA DE AMBATO	
Artículo	Texto
Art. 116.-	El Vicerrectorado Administrativo es el encargado de gestionar y administrar los procesos de talento humano, financieros, servicios administrativos, recursos de infraestructura y tecnologías de la información y comunicaciones; así como coordinar la consecución de los planes, programas y proyectos necesarios para su desarrollo con todos los procesos institucionales. Para su accionar contará con las siguientes dependencias: e) Dirección de Tecnologías de Información y Comunicación
CAPÍTULO 6. DE LA DIRECCIÓN DE TECNOLOGÍA DE INFORMACIÓN Y COMUNICACIÓN	
Art. 129.-	De la Misión. - Investigar, asesorar, dirigir, planificar, administrar bienes y servicios, y ejecutar planes, programas y proyectos con el fin de proveer nuevas tecnologías de información y comunicaciones (TICs) que permitan optimizar la gestión institucional, atención al cliente y toma de decisiones, con calidad, productividad y mejoramiento continuo; garantizando la disponibilidad, integridad y confiabilidad de la información, software, hardware, datos y comunicaciones institucionales; estableciendo normas y políticas internas para su gestión.
Art. 130.-	De las atribuciones y responsabilidades. - a) Asesorar y gestionar la implementación de nuevas tecnologías de información y comunicación para innovar procesos institucionales; c) Desarrollar, implementar y administrar el sistema integrado de información universitaria; d. Capacitar en el uso de Sistemas de Información, proporcionando el apoyo técnico necesario a los operadores y usuarios; coordinando la provisión de recursos y servicios informáticos y de comunicaciones para el procesamiento de datos e información institucional; f) Planificar, organizar y coordinar la administración y actividades relacionadas a la Dirección TIC's con la implementación de mejores prácticas tecnológicas, permitiendo mejorar el rendimiento, valor y control sobre las inversiones en tecnología de la información; m) Elaborar, proponer, ejecutar y evaluar los planes, programas y proyectos de su gestión, así como efectuar las reformas que permitan retroalimentar y mejorar los procesos críticos;

ESTATUTO DE LA UNIVERSIDAD TÉCNICA DE AMBATO	
	q) Ejercer las atribuciones y responsabilidades que determine la Constitución, Leyes de la República, la Contraloría General del Estado, la Universidad Técnica de Ambato;
Art. 131.-	<p>De la Estructura Interna. - La Dirección de Tecnologías de la Información y Comunicación, estará constituida por las siguientes unidades:</p> <ul style="list-style-type: none"> • Gestión de Recursos de Producción, Redes y Mantenimiento • Gestión de Desarrollo • Gestión de Seguridades • Gestión de Base de Datos

Fuente: (H. Consejo Universitario, 2013)

REGLAMENTO ORGÁNICO DE GESTIÓN ORGANIZACIONAL POR PROCESOS DE LA UNIVERSIDAD TÉCNICA DE AMBATO

Tabla 7. Reglamento Orgánico de Gestión Organizacional por Procesos de la Universidad Técnica de Ambato

REGLAMENTO ORGÁNICO DE GESTIÓN ORGANIZACIONAL POR PROCESOS DE LA UTA	
Artículo	Texto
Art. 8.-	<p>Estructura Orgánica. - La Universidad Técnica de Ambato, tendrá la siguiente estructura organizacional sustentada en su misión y objetivos institucionales.</p> <p>1.4 Procesos Habilitantes de Apoyo 1.4.1 Vicerrectorado Administrativo 1.4.5 Dirección de Tecnología de Información y Comunicación</p> <p>1) Gestión de Recursos de Producción, Redes y Mantenimiento 2) Gestión de Desarrollo 3) Gestión de Seguridades 4) Gestión de Base de Datos</p>

Fuente: (H. Consejo Universitario, 2012)

REGLAMENTO DE LA DIRECCIÓN DE TECNOLOGÍA DE INFORMACIÓN Y COMUNICACIÓN -DITIC- DE LA UNIVERSIDAD TÉCNICA DE AMBATO

Tabla 8. Reglamento de la Dirección de Tecnología de Información y Comunicación -DITIC- de la Universidad Técnica de Ambato

REGLAMENTO DE LA DITIC-UTA	
Artículo	Texto
Art. 1.-	<p>Finalidad. - La Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato con sus siglas DITIC es la encargada de gestionar las Tecnologías de la Información y Comunicaciones de la Institución, mediante su desarrollo, administración y mantenimiento; a fin de contribuir al direccionamiento estratégico, al desarrollo de las capacidades educativas, la seguridad integral de los espacios académicos y el apoyo al desarrollo Institucional.</p>

REGLAMENTO DE LA DITIC-UTA	
Art. 2.-	Ámbito de Trabajo. - La Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato es la dependencia responsable de administrar los recursos de Tecnologías de Información y Comunicación y sistemas de la Institución y de las entidades con las que ésta comparte recursos informáticos, además establecer normas, estándares, planes y metodologías para la gestión de la tecnología de la información, comunicaciones e infraestructura de redes, sistemas, equipos, bases de datos, comunicaciones informáticas y de telefonía IP.
Art. 3.-	Visión. - La visión de la Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato es constituirse en una Dirección líder en el diseño, desarrollo y optimización de tecnologías de la información y comunicación, con calidad y con espíritu de servicio, consolidada como modelo a seguir para otras Direcciones y Unidades de la Universidad Técnica de Ambato.
Art. 4.-	Misión. - La misión de la Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato es investigar, asesorar, dirigir, planificar administrar bienes y servicios, y ejecutar planes, programas y proyectos con el fin de proveer nuevas tecnologías de información y comunicaciones (TICs) que permitan optimizar la gestión institucional, atención al cliente y toma de decisiones, con calidad, productividad y mejoramiento continuo; garantizando la disponibilidad, integridad y confiabilidad de la información, software, hardware, datos y comunicaciones institucionales; estableciendo normas y políticas internas para su gestión.
Art. 7.-	Objetivo General. - Esta Unidad tendrá como objetivo general el coordinar y asegurar el cumplimiento de las actividades de la gestión de las Tecnologías de Información y Comunicación, que permitan el alineamiento con los objetivos estratégicos de la Universidad Técnica de Ambato y sus funciones operativas.
Art. 8.-	Objetivo Específicos. - Los objetivos específicos de la Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato son: a) Estandarizar las normas de usabilidad, accesibilidad y buenas prácticas de las Tecnologías de Información y Comunicación en la Universidad Técnica de Ambato en función a políticas y prácticas de nivel mundial; b) Garantizar la disponibilidad y correcto funcionamiento de los servicios de Tecnologías de Información en todas las unidades organizativas de la comunidad universitaria;
Art. 11.-	De la Gestión de Desarrollo. - La gestión de desarrollo estará a cargo de un Especialista de Tecnología de Información y Comunicación el mismo que será responsable de proponer proyectos para implementar o mejorar los sistemas de información de la Institución, y coordinar las actividades relacionadas a: a. Analizar, desarrollar, implementar y administrar sistemas de información según las necesidades de la Universidad; b. Participar en el diseño de bases de datos y gestión web de la Universidad; c. Desarrollar, implementar y administrar el sistema integrado de información universitaria; d. Coordinar la implementación de sistemas informáticos en las unidades académicas y administrativas de la Institución; l. Documentar técnicamente el software desarrollado;

REGLAMENTO DE LA DITIC-UTA	
	<p>m. Generar manuales técnicos y de usuario de todos los sistemas desarrollados;</p> <p>n. Elaborar políticas respecto al desarrollo de software institucional, considerando normativas y estándares internacionales, reglamentos y políticas institucionales;</p> <p>o. Llevar la documentación de versiones del software desarrollado;</p>
Art. 14.-	De los Procesos de Gestión de Desarrollo. - Los procesos de Gestión de Desarrollo estarán enmarcados siguiendo técnicas y modelos actuales de ingeniería y arquitectura del software, los mismos que cubran actividades de análisis, diseño, desarrollo, implementación, pruebas y mantenimiento del software.

Fuente: (H. Consejo Universitario, 2013)

REGLAMENTO DEL SISTEMA INTEGRADO DE INFORMACIÓN

Tabla 9. Reglamento del Sistema Integrado de Información SI-UTA

REGLAMENTO DEL SISTEMA INTEGRADO DE INFORMACIÓN SI-UTA	
Artículo	Texto
Art. 7.-	Del módulo. - Cada módulo que forme parte del Sistema Integrado de Información de la Universidad Técnica de Ambato (SI-UTA), deberá contar con su respectivo instructivo y manual de usuario.
Art. 13.-	De la administración de los módulos. - Las unidades académicas o administrativas encargadas de la administración de los módulos, solicitará a la Dirección de Tecnología de Información y Comunicación -DITIC-, actualización de los mismos en caso de requerirse.

Fuente: (H. Consejo Universitario, 2018)

2.4 Categorías Fundamentales

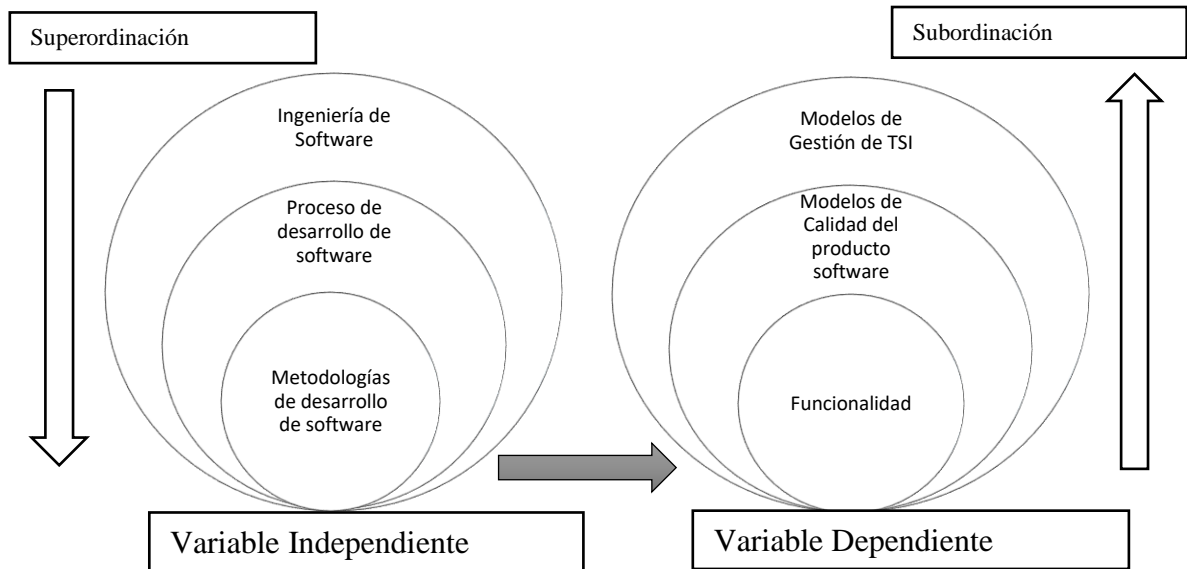


Figura 1. Inclusiones Conceptuales

Elaborado por: Investigador

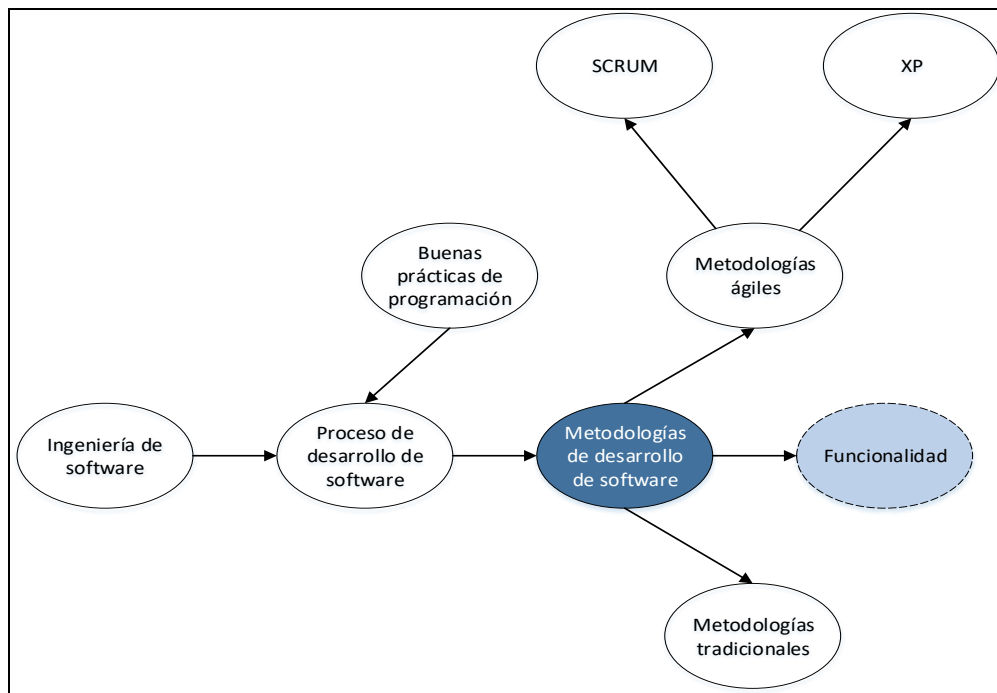


Figura 2. Constelación de Ideas de la Variable Independiente

Elaborado por: Investigador

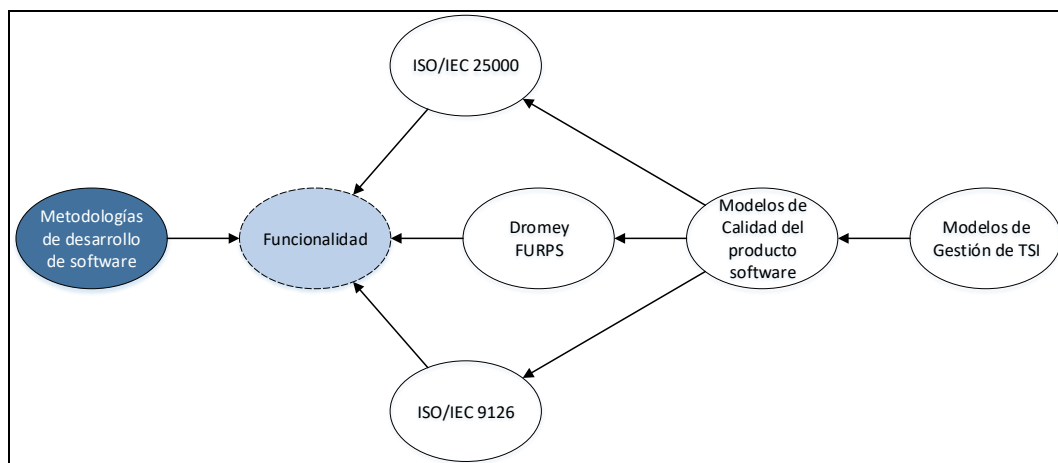


Figura 3. Constelación de Ideas de la Variable Dependiente

Elaborado por: Investigador

2.4.1 Categorías de la Variable Independiente

2.4.1.1 Ingeniería de Software

El término de ingeniería de software (IS) se originó en una conferencia en la OTAN en Garmisch, Alemania, 7-11 de octubre de 1968 (Oshana & Kraeling, 2013).

Jordi Cabot Sagrera define a la IS como la ciencia que estudia cual es la mejor manera de producir software de calidad, no solo que funcione, sino que sea eficiente y libre de errores. Al ser una ingeniería, las técnicas que forman parte de la IS deben estar bien fundamentadas de forma teórica o empírica (Cabot Sagrera, 2013).

La IS no es solo la programación de software (PS). La PS involucra a programadores desarrollando aplicaciones simples en un tiempo de vida corto y con poco mantenimiento. La IS involucra equipos de programadores con diferentes roles construyendo sistemas complejos en ciclos de vida indefinidos (Oshana & Kraeling, 2013).

2.4.1.2 Proceso de Desarrollo de Software

Para lograr que el proceso de desarrollo de software (PDS) sea exitoso, este debe estar soportado por metodologías, herramientas y tecnologías (Pressman, 2002).

Pressman señala que es importante seguir una serie de etapas definidas al momento de construir un sistema (Pressman, 2002). Estas etapas se dividen como se muestra en la Tabla 10:

Tabla 10. Etapas del proceso de desarrollo de software

Etapas	Descripción
Análisis o especificación de requisitos.	Es la etapa que define los requisitos, entregados por el cliente, que deberá tener el software a desarrollarse.
Diseño del software	El diseño define cómo se va a desarrollar el software toma en cuenta las especificaciones y las tecnologías a utilizarse, como lenguajes de programación, plataforma, base de datos, etc.
Programación o codificación	Se codifica el diseño definido de la etapa anterior.
Pruebas	Es la verificación del buen funcionamiento del software desarrollado.
Mantenimiento	Fase donde se realizan cambios al software resultantes de errores, nuevos requerimientos, actualizaciones, etc

Elaborado por: Investigador

Robert Oshana y Mark Kraeling señalan que el PDS incluye el desarrollo y mantenimiento del sistema, siendo el costo del mantenimiento mayor al costo de todo el desarrollo. Un pobre diseño implica un factor crítico en la implementación de un sistema (Oshana & Kraeling, 2013).

2.4.1.3 Buenas prácticas de programación

Los autores José Luis Solleiro y Rosario Castañón hacen referencia al concepto de buenas prácticas como las actividades diarias que se realizan en empresas y universidad con el objetivo de perfeccionar y reforzar sus actividades (Solleiro & Castañón, 2008).

2.4.1.3.1 Estándares de documentación

Los estándares son un conjunto de reglas que están normalizadas para garantizar el cumplimiento de los requisitos de un producto. Algunas ventajas de los estándares de documentación son: tener un código más sencillo, mejor adaptación al usuario, mejora del mantenimiento (Cardador Cabello, 2014).

Según Antonio Luis Cardador Cabello, en la actualidad se puede generar automáticamente documentación técnica que tendrá anotaciones como: (Cardador Cabello, 2014)

- Nombre de la clase y variables que utiliza.
- Constructor de la clase.
- Métodos que componen la clase.
- Parámetros de entrada y de salida, así como variables que componen al método.
- Relación entre un método y otro.

2.4.1.3.2 Documentación de código

La documentación de código es considerada como buena práctica en el desarrollo de software. Su objetivo es poder brindar una guía de lo desarrollado para futuras modificaciones y hacerlo más eficiente (Cardador Cabello, 2014).

2.4.1.3.3 Tabulación

Con la tabulación se logra que el código escrito sea más legible, resultando ser más fácil de ver y entender (Casado Iglesias, 2014).

2.4.1.3.4 Control de versiones

En el ciclo de desarrollo de software es normal que existan modificaciones antes de entregar a los usuarios para su verificación; por ello es muy importante llevar un control de versiones (Cardador Cabello, 2014).

2.4.1.4 Metodologías de desarrollo de software

Las metodologías de desarrollo de software (MDS) son el conjunto de pasos y procedimientos a seguir para el desarrollo de sistemas (Pressman, 2002).

Avison y Fitzgerald señalan a la MDS como una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayuden a los desarrolladores a implementar sistemas de información. Esta metodología está formada por fases que

guiarán en la elección de técnicas apropiadas para planificar, gestionar y evaluar el proyecto (Córdoba, 2014).

Las MDS definen en qué orden se va a seguir cada una de las etapas definidas en el PDS. Las MDS nos indican cuales son las actividades por aplicarse en cada fase del ciclo de vida y la forma en cómo se organizan y estructuran estas fases para lograr el desarrollo requerido (Cabot Sagrera, 2013).

Daynel Díaz Polo en su trabajo de investigación señala que una MDS debería definir con precisión los artefactos, roles y actividades, junto con buenas prácticas y técnicas, guías de adaptación al proyecto y guías para uso de herramientas de apoyo (Díaz Polo, 2011).

2.4.1.5 Metodologías tradicionales

2.4.1.5.1 Modelo en Cascada o Secuencial

El modelo secuencial o en cascada (*waterfall model* en inglés) propone que cada fase debe completarse en su totalidad antes de iniciar la siguiente. Para corregir un error se tenía que volver a la fase anterior, para después, seguir con el proceso secuencial (Cabot Sagrera, 2013). La Figura 4 muestra la interacción de cada etapa del modelo en cascada.

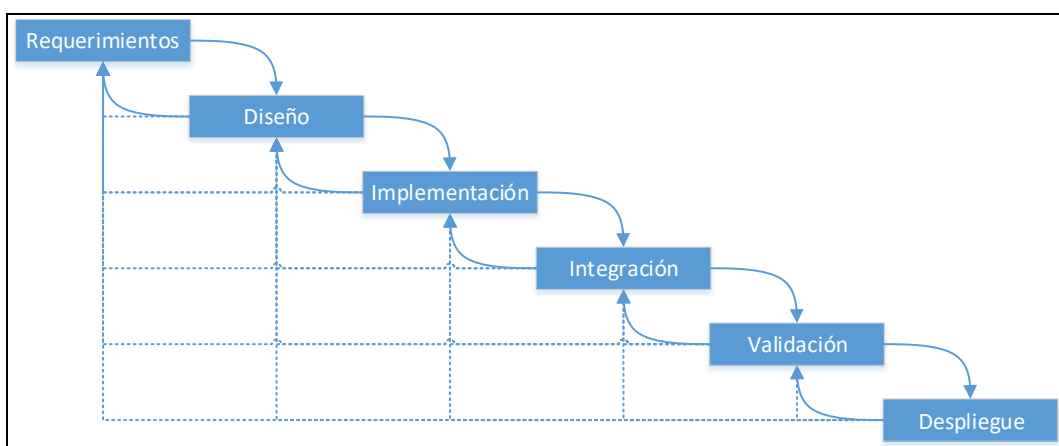


Figura 4. Modelo de desarrollo de software en Cascada o Secuencial

Fuente: (Douglass, 2013)

Este modelo es muy difícil aplicarlo a problemas reales, debido a que el cliente no siempre entrega todas las especificaciones al principio del proyecto. Los resultados solo se los puede observar al final del proyecto, por lo que no permite ningún tipo de retroalimentación; y si no se satisface las expectativas del cliente, se corre el riesgo de volver a empezar todo el proceso (Cabot Sagrera, 2013).

2.4.1.5.2 Modelo Iterativo e Incremental

El modelo iterativo divide al proyecto en tareas, y cada tarea debe alcanzar determinadas metas (Córdoba, 2014).

El modelo incremental reduce el tiempo de desarrollo. Divide a todos los requisitos en incrementos independientemente funcionales, los mismos que son analizados antes de comenzar el desarrollo

En estos modelos el software se desarrolla de forma incremental (Figura 5) y dan la oportunidad al cliente de valorar cada parte del software que se produce en cada incremento (Cabot Sagrera, 2013).

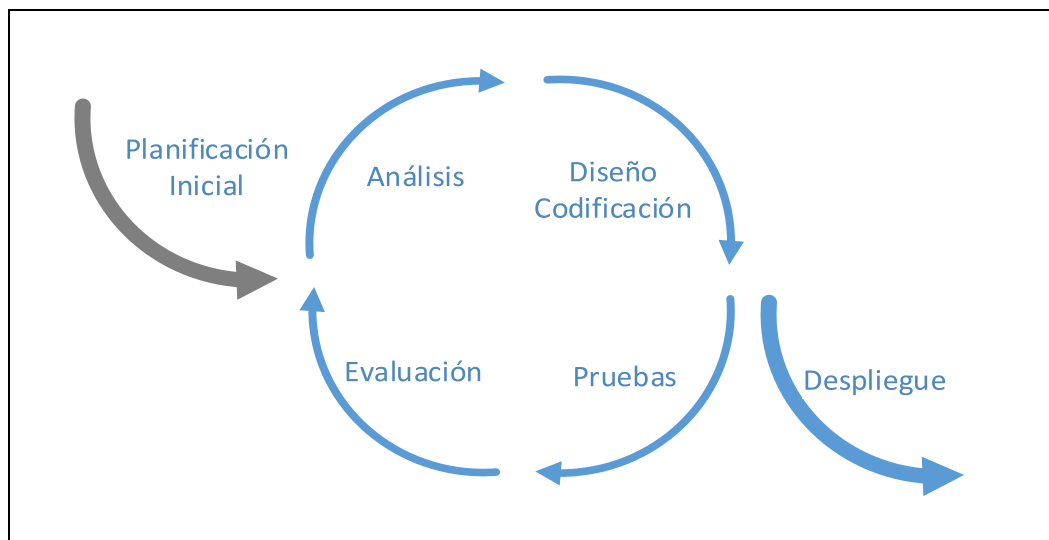


Figura 5. Modelo de desarrollo de software Incremental.

Fuente: (Cabot Sagrera, 2013)

2.4.1.5.3 Modelo en Espiral

El modelo en espiral es una combinación del modelo secuencial y el iterativo, pero incluye el análisis de alternativas y la identificación y reducción del riesgo (Córdoba, 2014).

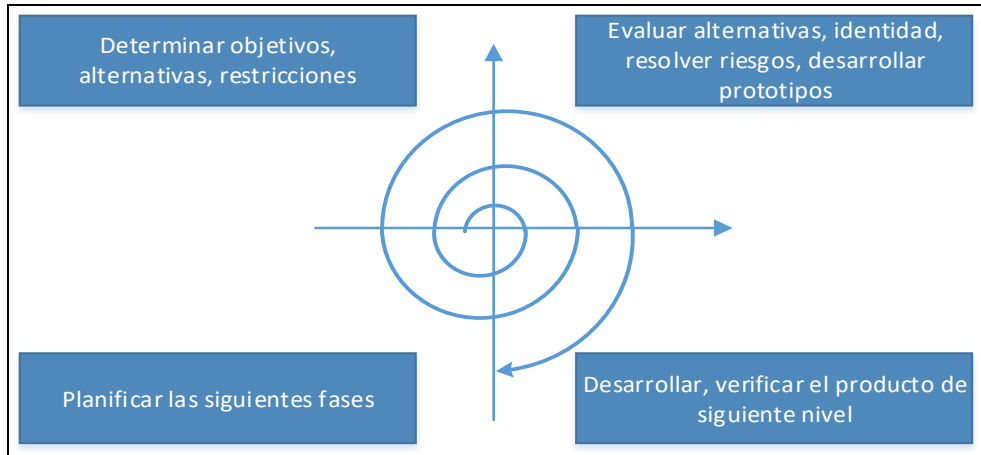


Figura 6. Modelo de desarrollo de software en Espiral

Fuente: (Douglass, 2013)

2.4.1.6 Metodologías Ágiles

Las metodologías de desarrollo ágil (MDA) se basan en un desarrollo iterativo e incremental, donde los requerimientos y entregables van evolucionando con la colaboración de grupos organizados (Córdoba, 2014).

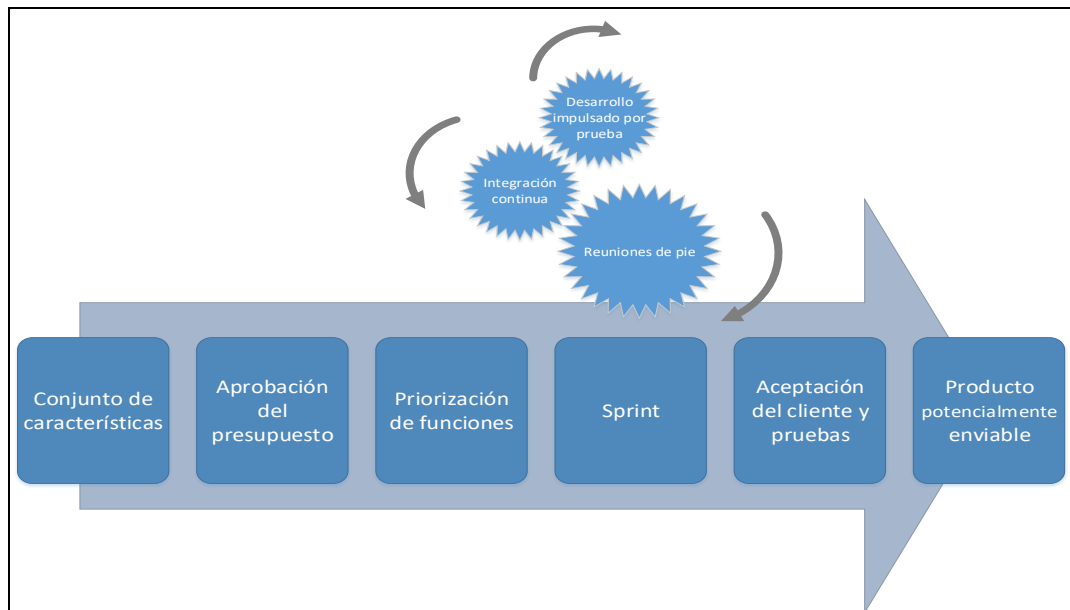


Figura 7. Modelo de desarrollo de software Ágil

Fuente: (Douglass, 2013)

El Manifiesto Ágil (*Agile Manifesto* en inglés) es un conjunto de principios y valores que tienen que cumplir todos los métodos “ágiles” y que surgieron a raíz de una reunión que tuvieron los principales representantes de cada método a principios del siglo XXI (Beck, et al., 2001).

(Beck, et al., 2001) **Manifiesto para el desarrollo ágil de software:**

“Estamos poniendo al descubierto mejores maneras de desarrollar software haciéndolo y ayudando a otros a hacerlo.

Mediante este trabajo hemos acabado valorando:

- **Individuos e interacciones por encima de procesos y herramientas.**
- **Software que funciona por encima de documentación exhaustiva.**
- **Colaboración con el cliente por encima de negociación de contratos.**
- **Respuesta al cambio por encima de ceñirse a una planificación.**

Es decir, aunque los elementos de la derecha tienen valor, nosotros valoramos más los de la izquierda.”

La Tabla 11 describe la interpretación de cada elemento del Manifiesto para el desarrollo ágil de software: (Opelt, Gloger, Pfarl, & Mittermayr, 2013)

Tabla 11. Manifiesto para el desarrollo ágil de software

Elemento	Interpretación
Individuos e interacciones por encima de procesos y herramientas.	Los procesos de desarrollo ágil suponen que para entregar un producto es esencial que todos los miembros del equipo e interesados hablen entre ellos e intercambien ideas constantemente. A pesar de que los equipos con procesos definidos y buenas herramientas de desarrollo son productivos; estos no son en más importantes que los individuos y sus interacciones.
Software que funciona por encima de documentación exhaustiva	La documentación sólo es útil cuando permite a una persona entender un tema de forma más rápida y eficiente. La documentación también es útil cuando permite retomar un desarrollo.
Colaboración con el cliente por encima de negociación de contratos	Los mejores contratos no siempre llevan al éxito de un proyecto. Los proyectos de desarrollo de software son exitosos sólo cuando desarrolladores y clientes trabajan estrechamente juntos.
Respuesta al cambio por encima de ceñirse a una planificación	En proyectos de desarrollo de software con metodologías ágiles, se planea con más frecuencia que con métodos tradicionales. Para que los planes se desarrollen con la mayor eficacia y rapidez posible, es necesario coordinar el proceso comunicativo entre las distintas partes.

Elaborado por: Investigador

Según Opelt & et. al. concluyen que se han desarrollado varios procedimientos y herramientas de planificación, instrucciones concretas sobre cómo organizar la planificación de un sprint. Los participantes son conscientes de que cada una de estas actividades de planificación se repetirán iterativamente y que el plan debe ajustarse continuamente. El equipo de desarrollo planea cada día colaborar para lograr el objetivo de sprint. Durante el sprint, el equipo de desarrollo y el propietario del producto discuten cómo se ejecutará el próximo sprint, por lo tanto, el proceso de planificación es completamente transparente (Opelt, Gloger, Pfarl, & Mittermayr, 2013).

El Manifiesto se basa en 12 principios para el desarrollo de software ágil: (Beck, et al., 2001)“

1. *Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.*
2. *Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.*
3. *Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.*
4. *Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.*
5. *Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.*
6. *El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.*
7. *El software funcionando es la medida principal de progreso.*
8. *Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.*
9. *La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.*
10. *La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.*
11. *Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.*
12. *A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.”*

Existen varios procesos que aplican estos principios, Bruce Douglass señala algunos de ellos: Extreme Programming (XP), Lean Software Development, Crystal Clear, y Scrum (Douglass, 2013).

Las metodologías ágiles ofrecen ventajas significativas sobre los enfoques tradicionales del desarrollo de software, en particular: mejoran la calidad, disminuye tiempos de entrega y, mejoran la satisfacción de los clientes. Estas ventajas se alcanzan con un cambio en el orden en que se realizan las tareas, así como la eliminación de tareas que no agregan valor a los objetivos del desarrollo, y en su lugar ponen mediciones y buenas prácticas de planificación y gestión de proyectos (Douglass, 2013).

2.4.1.6.1 Scrum

Ken Schwaber y Jeff Sutherland definen a Scrum como un marco de trabajo para el desarrollo y mantenimiento de productos, en el cual las personas pueden empezar problemas complejos, y a la vez, entregar productos con un máximo valor de manera productiva y creativa (Schwaber & Sutherland, 2017).

Scrum es ahora un estándar en el desarrollo ágil (Opelt, Gloger, Pfarl, & Mittermayr, 2013). En los últimos años ha evolucionado de ser una metodología de gestión de proyectos a un nuevo sentido de cómo gestionar equipos de trabajo, departamentos, unidades organizativas y empresas. Inicialmente las empresas utilizan Scrum como un método en la gestión de proyectos, sin embargo, otras evolucionan y lo usan para el ciclo de vida de la organización (Opelt, Gloger, Pfarl, & Mittermayr, 2013).

Scrum está diseñado para dar a los equipos libertad al grupo para que pueda existir un ambiente de trabajo agradable y productivo. Según la interpretación de Andreas Opelt et. al. “*Scrum permite a cada miembro del equipo recuperar las habilidades y la competencia que necesitan para asumir la responsabilidad*” (Opelt, Gloger, Pfarl, & Mittermayr, 2013).

En Scrum existen algunas reglas y principios que debe cumplirse estrictamente, que están en concordancia con el manifiesto ágil. Este enfoque se basa en una interacción de reglas, disciplina, responsabilidad, ayudarse unos a otros y no brillar personalmente (Opelt, Gloger, Pfarl, & Mittermayr, 2013).

Teoría de Scrum

Scrum se basa en el empirismo, el cual asegura que el conocimiento depende de la experiencia y de la toma de decisiones. Para optimizar la predictibilidad y el control del riesgo, el enfoque de Scrum es iterativo e incremental. Para la implementación del control de procesos empírico, Scrum es soportado por tres pilares: transparencia, inspección y adaptación (Schwaber & Sutherland, 2017).

Los Valores de Scrum

Scrum propone valores que hacen énfasis en una comunicación eficientes, estos valores se detallan en la Tabla 12: (Monte Galiano, 2016)

Tabla 12. Valores de Scrum

Valores	Descripción
Compromiso	Es necesario un alto grado de compromiso en el grupo.
Enfoque	Permite dividir el problema en partes más pequeñas para concentrarse en la resolución de un único problema.
Organización abierta	Trabajo colaborativo, se ofrece y se pide ayuda, se aprende del resto del equipo.
Respeto	Se logra con el compromiso y el trabajo en equipo.
Coraje	Es la fuerza de voluntad para afrontar los retos de proyectos complejos e inciertos.

Fuente: (Monte Galiano, 2016)

Roles de Scrum

El equipo Scrum está conformado por los roles que se detallan en la Tabla 13: (Monte Galiano, 2016)

Tabla 13. Roles de Scrum

Roles	Descripción
Dueño de producto	<ul style="list-style-type: none"> Única persona. Representate, propietario, o interesado en el desarrollo del producto.

Roles	Descripción
	<ul style="list-style-type: none"> • Conoce a profundidad el negocio. • Conoce los requisitos y funcionalidades del producto. • Máximo responsable que constantemente define los requisitos. • Representa al grupo y se comunica con cada uno de ellos • Dar prioridades a cada etapa.
Equipo de desarrollo	<ul style="list-style-type: none"> • Debe estar compuesto de 3 a 9 personas. • Auto-organizados.
Scrum Master	<ul style="list-style-type: none"> • Es el facilitador o guía. • Se encuentra constantemente organizando al equipo. • Actúa como intermediario. • Se asegura de que todo funcione correctamente.

Fuente: (Monte Galiano, 2016)

Eventos del Sprint

Los eventos del Sprint se detallan en la Tabla 14:

Tabla 14. Eventos del Sprint

Eventos	Descripción
Planificación del Spring	Reunión de 8 horas durante el cual el Equipo Scrum planifica los requisitos o Backlog a desarrollar para la entrega del Sprint y las acciones para garantizar su cumplimiento.
Scrum Diario	Reunión diaria de 15 minutos para los miembros del equipo de desarrollo para inspeccionar el trabajo realizado desde la última reunión y sincronizar el trabajo para las próximas 24 horas.
Revisión del Sprint	Reunión de 4 horas que tiene lugar al final del Sprint. Participa el Equipo Scrum y otras partes interesadas para inspeccionar el incremento y recoger la opinión de los usuarios que ayudan a definir o actualizar los backlogs (feedback).
Retrospectiva del Sprint	Una reunión de 3 horas por el equipo Scrum para descubrir las lecciones que se pueden incorporar en Sprint futuras. Mejora continua.
Sprint	Duración recomendada 4 semanas. Menos afecta el incremento, más pone en riesgo la complejidad de la entrega.

Elaborado por: Investigador

Artefactos

Son elementos o herramientas que permiten realizar la gestión en un proyecto de desarrollo ágil, estos artefactos se detalla en la Tabla 15:

Tabla 15. Artefactos de Scrum

Artefactos	Descripción
Lista de Producto (Product Backlog)	Historias de usuario. Donde se van a definir o que es lo que se debe realizar. La funcionalidad que va a tener la aplicación.
Lista de Pendientes del Sprint (Sprint Backlog)	Conjunto de elementos de la Lista de Producto más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint.
Incremento	Versión entregable. Debe cumplir el concepto de completo.

Elaborado por: Investigador

2.4.1.6.2 XP

eXtreme Programming (XP) es una metodología ágil de desarrollo de software de calidad, de forma eficiente y efectiva (Cortizo Pérez, Expósito Gil, & Ruiz Leyva, 2012). Se encuentra fundamentada en una serie de buenas prácticas y valores, que tiene como objetivo el aumentar la productividad en el proceso de desarrollo de software.

Los valores y principios de XP se detallan en la Tabla 16:

Tabla 16. Valores y principios de XP

Valores y principios	Descripción
Valores	<ul style="list-style-type: none"> • Comunicación • Simplicidad • Retroalimentación • Coraje • Respeto
Principios	<ul style="list-style-type: none"> • Planificación • Versiones pequeñas • Sistema metafórico • Diseños simples • Pruebas continuas • Refactoring • Programación en parejas • Propiedad colectiva del código • Integración continua • 40 horas por semana • El cliente en su sitio • Estándares de codificación

Elaborado por: Investigador

2.4.2 Categorías de la Variable Dependiente

2.4.2.1 Modelos de Gestión de Tecnologías y Sistemas de Información (TSI)

La evolución de los sectores de tecnología de información y comunicación (TIC) ha llevado a un reenfoque del rol que tienen las normas, pasando de la normalización a un enfoque global de sistemas de gestión (Fernández & Piattini, 2012).

Los modelos basados en experiencias exitosas de ingeniería de software, sirven de guía para la mejora continua en las empresas (Álvarez Rodríguez, et al., 2008). En los últimos años se han publicado diversos estándares con las buenas prácticas que se deben seguir para mejorar la calidad en los procesos, proyectos, productos y servicios que componen a los sistemas de información (Pino, Piattini, & Fernández, 2014).

En la actualidad existen varios marcos para el gobierno y gestión de las tecnologías y sistemas de información (TSI), los más conocidos son: COSO, COBIT, CMMi, UNE-ISO/IEC 27001:2007, ISO/IEC 15504:2004, ISO/IEC 15408:2009, ITIL, PMBOK, etc. (Fernández & Piattini, 2012). La Figura 8 muestra las normas propuestas por Asociación Española de Normalización y Certificación (AENOR).

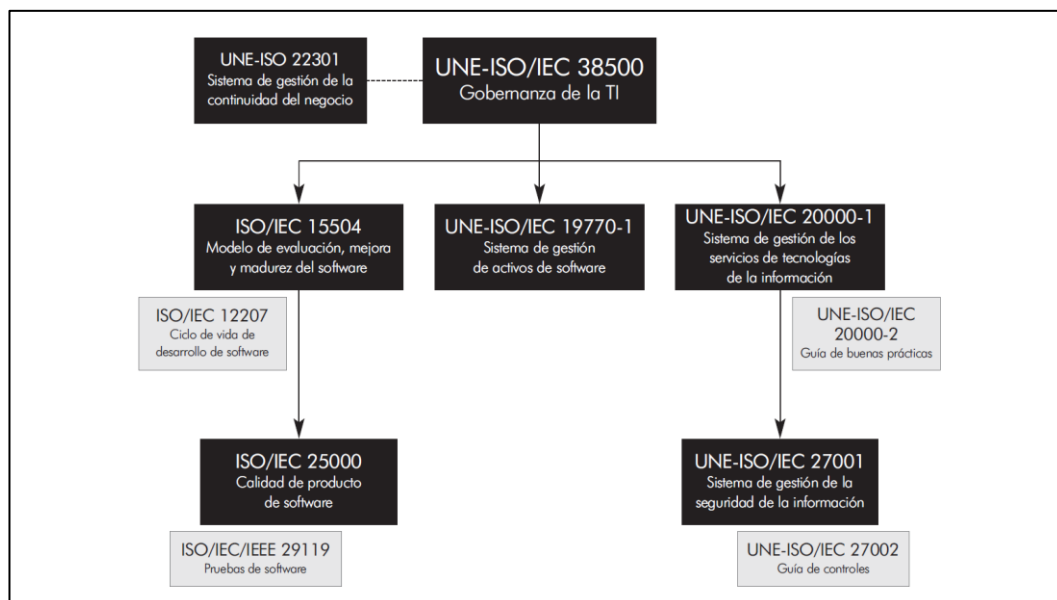


Figura 8. Modelo de gobierno y gestión de las TIC propuesto por AENOR

Fuente: (Pino, Piattini, & Fernández, 2014)

Los beneficios que se reciben con la adaptación de un modelo de mejora de procesos de software son: incrementar la calidad del software, disminuir costos y riesgos, optimizar los recursos, la productividad, tiempos de desarrollo, competitividad y el crecimiento sostenible en la industria de software (Álvarez Rodríguez, et al., 2008; Trujillo Casañola, 2015).

2.4.2.2 Modelos de Calidad del producto software

La calidad del software se basa en evidencias directas sobre el producto y no solo en el proceso para su construcción, por esta razón, las empresas se preocupan en garantizar la calidad del producto software y no solo la calidad del proceso para su desarrollo (Fernandez & Piattini, 2013).

Entre los principales beneficios de evaluar la calidad del producto software en organismos dedicados a desarrollar software están: (Fernandez & Piattini, 2013)

- Controlar la calidad del producto software.
- Mejorar las características del producto software.
- Asegurar a los clientes/usuarios un nivel de calidad.

Hasta la actualidad ha proliferado el desarrollado de diferentes modelos, normas y estándares para evaluar la calidad del producto software (Estayno, Dapozo, Cuenca, & Greiner, 2009), su evolución ha adquirido gran importancia debido a este factor y a la externalización (Fernández & Piattini, 2012). La Tabla 17 muestra los principales modelos relacionados con la evaluación del producto software:

Tabla 17. Modelos relacionados con la evaluación del producto software

Modelos	Descripción
ISO/IEC 9126 Software engineering – Product quality	Serie de normas sobre la calidad del producto software.
UNE-ISO/IEC 14598 Tecnología de la información. Evaluación del producto software	Serie de normas para la evaluación del software.
ISO/IEC 25000 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE)	Normas relacionadas con los modelos, mediciones y evaluaciones de la calidad, que supone la evolución de las normas ISO/IEC 9126 e ISO/IEC 14598

Fuente: (Fernández & Piattini, 2012)

La familia de normas ISO/IEC 25000 conocida como SQuaRE (*Software product Quality Requirements and Evaluation*), que es la evolución de las normas ISO/IEC 9126 e ISO/IEC 14598, tiene el objetivo de crear un marco de trabajo para evaluar la calidad del producto software (Fernández & Piattini, 2012).

La división de la norma ISO/IEC 25000 para definir su terminología es la ISO/IEC 25010, en donde se determinan las características y subcaracterísticas de calidad para evaluar un producto software, las mismas que se detallan en la Tabla 18:

Tabla 18. Características y subcaracterísticas de calidad según ISO/IEC 25010

CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DE CALIDAD SEGÚN ISO/IEC 25010	
Características	Subcaracterísticas
Adecuación Funcional	<ul style="list-style-type: none"> • Completitud • Exactitud • Pertinencia
Eficiencia de Desempeño	<ul style="list-style-type: none"> • Comportamiento temporal • Utilización de Recursos • Capacidad
Compatibilidad	<ul style="list-style-type: none"> • Coexistencia • Interoperabilidad
Usabilidad	<ul style="list-style-type: none"> • Inteligibilidad • Aprendizaje • Operabilidad • Protección a errores de usuario • Estética • Accesibilidad
Fiabilidad	<ul style="list-style-type: none"> • Madurez • Disponibilidad • Tolerancia a fallos • Capacidad de recuperación
Seguridad	<ul style="list-style-type: none"> • Confidencialidad • Integridad • No repudio • Autenticidad • Responsabilidad
Mantenibilidad	<ul style="list-style-type: none"> • Modularidad • Reusabilidad • Analizabilidad • Capacidad de ser modificado • Capacidad de ser probado
Portabilidad	<ul style="list-style-type: none"> • Adaptabilidad • Facilidad de instalación • Capacidad de ser reemplazado

Fuente: (ISO/IEC, 2011)

2.4.2.3 Funcionalidad

La funcionalidad es un requisito específico necesario desde el punto de vista del usuario, mismo que debe ser entregado por un sistema a través de sus componentes (Soffer & Dori, 2013). La propiedad funcional se especifica en función de las entradas, salidas o el comportamiento funcional de un sistema (Maalej & Kumar Thurimella, 2013).

La división de la norma ISO/IEC 25000 que determina su terminología es la norma ISO/IEC 25010; en esta se establece que la funcionalidad, como característica de calidad, es la capacidad que tiene el producto software de cumplir con los requisitos del usuario (Delgado & Fernández, 2018).

La Tabla 19 detalla los modelos de calidad del producto software que se encuentran en relación con los atributos de funcionalidad o adecuación funcional como característica de calidad; esta se encuentra en consideración del análisis comparativo de modelos y estándares para evaluar la calidad del producto software según (Moreno, Bolaños, & Navia, 2010), (Musa & Ikhateeb, 2013) y (González, André, & Hernández, 2015).

Tabla 19. Modelos y estándares de calidad del producto software

MODELOS Y ESTÁNDARES DE CALIDAD DEL PRODUCTO SOFTWARE	
Modelo	Características y Subcaracterísticas
McCall	<p>Desarrollado originalmente por James McCall para la Fuerza Aérea de los Estados Unidos en el año de 1977 (Estévez & Esteban, 2016).</p> <p>La principal contribución del modelo de calidad de McCall es su marco de trabajo, mismo que sirve para medir la calidad del software (Datta, 2007). El modelo de calidad de McCall divide los atributos de calidad del software en tres dimensiones o perspectivas:</p> <ul style="list-style-type: none">• La revisión del producto. Que es la capacidad de cambio.• La transición del producto. Es la capacidad de adaptación.• Las operaciones del producto. Es la característica básica que va a tener un producto. <p>Cada perspectiva a su vez está organizada en factores y criterios de calidad, las perspectivas y factor asociadas a funcionalidad según (Gómez, Andrade, & Macías, 2010) son: (Constanzo, 2014)</p>

MODELOS Y ESTÁNDARES DE CALIDAD DEL PRODUCTO SOFTWARE	
Modelo	Características y Subcaracterísticas
	<p>Operación del producto. - Son las características de operación y abarcan factores como:</p> <ul style="list-style-type: none"> • Corrección: El grado en que una funcionalidad coincide con su especificación. • Integridad: La protección de programa del acceso no autorizado. <p>Transición del Producto. - Es la adaptabilidad al nuevo ambiente, el cual tiene como factores de calidad:</p> <ul style="list-style-type: none"> • Interoperabilidad: El esfuerzo requerido para acoplar el sistema a otro sistema.
Dromey	<p>Propuesta por el Profesor Dr. Robert Geoff Dromey en 1995 (Dromey, 1995). Es un marco de referencia para modelos de calidad y se basa en la influencia de las propiedades de un producto software sobre los atributos de calidad (UTIC, 2013).</p> <p>El atributo de funcionalidad se encuentra dentro de exactitud; esta es una de las cuatro categorías o factores de calidad que proponen el modelo: exactitud, internas, contextuales y descriptivas (Musa & Ikhateeb, 2013).</p> <p>Funcionalidad. - La funcionalidad es un atributo del software de alto nivel de calidad, mismo que debe ser completo, compatible y no superpuesto; además la funcionalidad incide sobre las propiedades de calidad de exactitud como: Computable, Completo, Asignado, Preciso, Inicializado, Progresivo, Variante, Consistente (Dromey, 1995).</p>
FURPS	<p>Modelo presentado por Robert Grady (Vern & Kumar, 2013) donde clasifica a los factores de calidad en cinco categorías cuyas siglas en inglés toman el nombre del modelo: Funcionalidad (Functionality), Usabilidad (Usability), Confiabilidad (Reliability), Desempeño (Performance) y Soportabilidad (Supportability) (Moreno, Bolaños, & Navia, 2010).</p> <p>Funcionalidad. - Los criterios asociados al factor de funcionalidad son:</p> <ul style="list-style-type: none"> • Características y capacidades del programa • Generalidad de las funciones • Seguridad del Sistema
ISO/IEC 9126	<p>ISO/IEC 9126 es una serie de normas para la evaluación de la calidad del producto en ingeniería de software. El modelo de calidad es definido en características y sub características que debe poseer un software con calidad (Granados La Paz, 2014). La característica y sub características son asociadas a funcionalidad son: (ISO/IEC, 2001)</p> <p>Funcionalidad. - Capacidad del producto software para satisfacer las necesidades implícitas y explícitas del producto software esperado. Se divide en las siguientes sub características: (ISO/IEC, 2001)</p> <ul style="list-style-type: none"> • Idoneidad. Capacidad del producto software de cumplir tareas específicas.

MODELOS Y ESTÁNDARES DE CALIDAD DEL PRODUCTO SOFTWARE	
Modelo	Características y Subcaracterísticas
	<ul style="list-style-type: none"> • Eficacia. Capacidad del producto software de proporcionar correctamente los resultados de la manera prevista. • Seguridad. Capacidad del producto software de proteger la información y evitar acceso no autorizado. • Interoperabilidad. Capacidad del producto software de interactuar con otros sistemas especificados. • Cumplimiento funcional. Capacidad del producto software para adherirse a normas o legislación relacionadas con la funcionalidad.
ISO/IEC 25000	<p>Conocida como SQuaRE (Software product Quality Requirements and Evaluation), la familia de normas ISO/IEC 25000 se encuentra relacionada a los modelos, mediciones y evaluaciones de calidad del producto software; esta supone la evolución de las normas ISO/IEC 9126 e ISO/IEC 14598 (Fernández Sánchez & Piattini Velthuis, 2012).</p> <p>El modelo de calidad de esta norma se encuentra determinado por características a ser consideradas al momento de evaluar un producto software. La característica y sub características en relación con la funcionalidad son: (ISO/IEC, 2011)</p> <p>Adecuación funcional. - Capacidad del producto software para proporcionar funciones que satisfacen las necesidades, declaradas e implícitas, cuando el producto se usa en un entorno específico. Se divide en las siguientes sub características: (ISO/IEC, 2011)</p> <ul style="list-style-type: none"> • Completitud funcional. Grado en el que el conjunto de funcionalidades cubre todas las tareas y objetivos especificados por el usuario. • Exactitud funcional. Capacidad del producto software para proveer resultados correctos con el nivel de precisión necesario. • Pertinencia funcional. Capacidad del producto software para facilitar la realización de las tareas y los objetivos especificados por el usuario.

Elaborado por: Investigador

2.5 Hipótesis

Estandarizando el proceso de desarrollo de software con la aplicación de buenas prácticas de programación y Scrum como marco de trabajo ágil se incrementará el grado de funcionalidad del producto software desarrollado por la Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato.

2.6 Señalamiento de Variables

Variable Independiente: Metodologías de desarrollo de software.

Variable Dependiente: Funcionalidad.

CAPITULO III

3 METODOLOGÍA

3.1 Enfoque

El presente trabajo de investigación tiene un enfoque cuantitativo; se va a medir el grado de funcionalidad resultante del producto software desarrollado correspondiente a la variable dependiente en relación directa a la metodología de desarrollo aplicada, que correspondiente a la variable independiente.

3.2 Modalidad básica de la investigación

Investigación Bibliográfica

La investigación se sustenta en información de libros, revistas y artículos científicos, tesis de grado y posgrado, leyes, estatutos, normativas, documentación técnica, estándares, marcos de trabajo y mejores prácticas para la elaboración del marco teórico.

Investigación de Campo

La investigación busca obtener información respecto al grado de funcionalidad de un producto software en proyectos de desarrollo y con el personal asignado.

3.3 Nivel o tipo de investigación

Investigación Experimental

La investigación se desarrolla de forma experimental aplicando distintos marcos de trabajo y procesos de desarrollo en grupos similares, con el objetivo de determinar el grado de funcionalidad del software desarrollado.

Investigación Descriptiva

La investigación es descriptiva porque se realiza un análisis para llegar a determinar la incidencia que tienen las metodologías de desarrollo en el grado de funcionalidad del producto software.

Explicativa

La investigación es explicativa porque se sustenta la importancia que la elección de una metodología o marco de trabajo en el incremento del grado de funcionalidad del software.

Investigación Correlacional

La investigación es correlacional porque busca medir el grado de relación entre la metodología o marco de trabajo aplicado y el grado de funcionalidad del producto software resultante del proceso de desarrollo.

3.4 Población y Muestra

El presente proyecto de investigación trabaja con la muestra total de los sistemas desarrollados que se encuentran en producción en la DITIC-UTA, la Tabla 20 indica el número de productos software evaluados clasificados por ámbitos.

Tabla 20. Muestra de estudio

Ámbito	Número	Porcentaje
Academia	14	45%
Gestión	12	39%
Vinculación	4	13%
Investigación	1	3%
Total	31	100%

Elaborado por: Investigador

La investigación se apoya en el análisis de información de proyectos hasta el primer semestre del 2018, con el propósito de identificar el grado de funcionalidad de los sistemas desarrollados.

3.5 Operacionalización de Variables

3.5.1 Variable Independiente

Tabla 21. Variable Independiente: Metodologías de desarrollo de software

Conceptualización o descripción	Dimensiones	Indicadores	Técnicas e instrumentos
Las metodologías de desarrollo de software son el conjunto de pasos y procedimientos a seguir para el desarrollo de sistemas.	<ul style="list-style-type: none">• Metodologías tradicionales.<ul style="list-style-type: none">○ Cascada○ Espiral• Metodologías ágiles.<ul style="list-style-type: none">○ Scrum○ XP	<ul style="list-style-type: none">• Tipo de metodología de desarrollo de software.	<ul style="list-style-type: none">• Observación estructurada.• Formulario de evaluación y medición.
	<ul style="list-style-type: none">• Buenas prácticas de programación	<ul style="list-style-type: none">• Código fuente documentado• Sistema de versionamiento• Documentación técnica	

Elaborado por: Investigador

3.5.2 Variable Dependiente

Tabla 22. Variable Dependiente: Funcionalidad.

Conceptualización o descripción	Dimensiones	Indicadores	Ítems básicos	Técnicas e instrumentos
La funcionalidad está orientada a satisfacer las necesidades implícitas y explícitas del producto software esperado.	Funcionalidad	Complejidad	Aseguramiento del cumplimiento de las necesidades del cliente	<ul style="list-style-type: none">• Encuesta• Cuestionario• Observación estructurada.• Formulario de evaluación y medición.
		Exactitud	Reducción de errores en el sistema en producción	
		Pertinencia	Mejorar la confianza y satisfacción del cliente	

Elaborado por: Investigador

3.6 Recolección de Información

La técnica de investigación para la recolección de información será la observación estructurada, que es una técnica utilizada en estudios o investigaciones partiendo de categorías o formulaciones específicas (Gallardo & Moreno, 1999); por lo que es necesario utilizar como instrumento un formulario de evaluación y medición, cuyo propósito es realizar una investigación enfocada al estudio de métricas (Marcos, 2018) de funcionalidad como característica de calidad del producto software.

El tipo de preguntas que contiene la evaluación están enfocadas al método de desarrollo utilizado y a la medición cuantitativa del grado de funcionalidad como característica de calidad del producto software (Chavarriaga & Arboleda, 2004).

Tabla 23. Recolección de la información

Preguntas básicas	Explicación
¿Para qué?	Para alcanzar los objetivos de la investigación
¿De qué personas u objetos?	31 productos software desarrollados
¿Sobre qué aspectos?	Desarrollo de software Funcionalidad del producto software
¿Quién, Quiénes?	Investigador: Ing. Jorge Vladimir Chávez Andrade
¿Cuándo?	Segundo semestre de 2018
¿Dónde?	Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato
¿Cuántas veces?	Una
¿Qué técnicas de recolección?	Observación estructurada
¿Con qué?	Formulario de evaluación y medición Software estadístico IBM SPSS
¿En qué situación?	Dentro del horario de trabajo con profesionalismo investigativo y absoluta confidencialidad y reserva.

Elaborado por: Investigador

3.7 Procesamiento y Análisis

- Revisión crítica de la información recogida; es decir depuración de información defectuosa, contradictoria, incompleta, no pertinente y otras fallas.
- Repetición de la recolección, en ciertos casos individuales para corregir errores de digitación.
- Manejo de información (reajuste de cuadros con casillas vacías o con datos tan reducidos cuantitativamente que no influyen significativamente en los análisis).
- Estudio estadístico de datos para presentación de resultados.

Análisis de Resultados

- Análisis de los resultados estadísticos, destacando tendencias o relaciones fundamentales de acuerdo con los objetivos e hipótesis.
- Interpretación de los resultados con apoyo del marco teórico en el aspecto pertinente.
- Establecimiento de conclusiones y recomendaciones.

CAPITULO IV

4 ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1 Análisis de resultados

Por medio de la recopilación de información a través de técnicas de observación estructura utilizando un formulario de evaluación y medición de la funcionalidad del producto software, se procedió a realizar el análisis e interpretación de resultados.

El formulario de evaluación y medición se aplica de manera general sobre cada software desarrollado, con el objetivo de obtener información relacionada a la aplicación de metodologías de desarrollo, al uso de buenas prácticas de programación y a su incidencia en la funcionalidad como métrica de calidad del producto software.

4.1.1 Formulario de evaluación y medición

El formulario de evaluación y medición se detalla en el Anexo 2, fue aplicado de manera general sobre los sistemas desarrollados en la DITIC-UTA. El detalle de cada evaluación se muestra en los anexos: Anexo 3 y Anexo 4, en consideración de los siguientes puntos:

Punto 1. Metodología de desarrollo.

Pregunta 1. ¿Qué tipo metodología de desarrollo se empleó?

Tabla 24. Metodología de desarrollo

Metodología (Tipo)	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
SCRUM (Ágil)	0	0	0	0
Empírica (Tradicional)	23	74,2	74,2	74,2
Cascada (Tradicional)	8	25,8	25,8	100
Total	31	100	100	

Elaborado por: Investigador

El 74,2% ha sido desarrollado de forma empírica sin ninguna metodología y el 25,8% restante utiliza una metodología tradicional (metodología en cascada). Ningún software ha sido desarrollado utilizando una metodología o marco de trabajo ágil.

Punto 2. Documentación de código.

Pregunta 2. ¿Se documentó el código fuente?

Tabla 25. Documentación de código

Respuesta	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
SI	0	0	0	0
NO	31	100	100	100
Total	31	100	100	

Elaborado por: Investigador

El 100% de los sistemas evaluados no cuenta con un código fuente documentado.

Punto 3. Control de cambios sobre el software.

Pregunta 3. ¿Se utilizó un sistema de versionamiento?

Tabla 26. Control de cambios sobre el software

Respuesta	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
SI	0	0	0	0
NO	31	100	100	100
Total	31	100	100	

Elaborado por: Investigador

El 100% de los sistemas evaluados no cuenta con un sistema de versionamiento.

Punto 4. Fuente de medición:

Pregunta 4. ¿Existe documento de especificación de requerimientos y diccionario de datos?

Tabla 27. Documento de especificación de requerimientos

Respuesta	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
SI	8	25,8	25,8	25,8
NO	23	74,2	74,2	100
Total	31	100	100	

Elaborado por: Investigador

Solo el 25,8% de los sistemas evaluados cuenta con un documento de especificación de requerimientos y diccionario de datos, el restante 74,2% no cuenta con la documentación técnica solicitada.

Punto 5. Completitud funcional.

Pregunta 5. ¿Qué tan completa es la implementación de acuerdo a las especificaciones de los requerimientos?

En consideración con la norma internacional ISO/IEC 25023 se utilizó la siguiente métrica de calidad para calcular la sub característica de completitud funcional dentro de la característica de calidad de adecuación funcional:

$$x_1 = 1 - ff/tf$$

Donde:

x_1 = Completitud funcional

ff = Número de funciones que faltan o están incorrectamente implementadas

tf = Número de funciones establecidas en la especificación de los requisitos

Valor deseado = El más cercano a 1 es mejor.

En la Tabla 28 se visualiza el grado de completitud funcional de cada producto software como resultado del proceso de evaluación y medición. El detalle del resultado de la evaluación y medición se encuentra en el Anexo 3 y Anexo 4.

Punto 6. Exactitud funcional.

Pregunta 6. ¿Cuánto del estándar requerido de exactitud se cumple?

En consideración con la norma internacional ISO/IEC 25023 se utilizó la siguiente métrica de calidad para calcular la sub característica de exactitud funcional dentro de la característica de calidad de adecuación funcional:

$$x_2 = e/te$$

Donde:

x_2 = Exactitud funcional

e = Número de elementos de datos implementados con el estándar específico de exactitud

te = Número total de elementos de datos implementados

Valor deseado = El más cercano a 1 es mejor.

En la Tabla 28 se visualiza el grado de exactitud funcional de cada producto software como resultado del proceso de evaluación y medición. El detalle del resultado de la evaluación y medición se encuentra en el Anexo 3 y Anexo 4.

Punto 7. Pertinencia funcional.

Pregunta 7. ¿Qué parte de las funciones implementadas son percibidas como idóneas?

En consideración con la norma internacional ISO/IEC 25023 se utilizó la siguiente métrica de calidad para calcular la sub característica de pertinencia funcional dentro de la característica de calidad de adecuación funcional:

$$x_3 = fu/tfu$$

Donde:

x_3 = Pertinencia funcional

fu = Número de funciones realmente útiles para realizar tareas específicas

tfu = Número de funciones implementadas para la consecución de tareas específicas

Valor deseado = El más cercano a 1 es mejor.

En la Tabla 28 se visualiza el grado de pertinencia funcional de cada producto software como resultado del proceso de evaluación y medición. El detalle del resultado de la evaluación y medición se encuentra en el Anexo 3 y Anexo 4.

Tabla 28. Tabulación de métricas de calidad para adecuación funcional

Ámbito	Software evaluado	Compleitud			Exactitud			Pertinencia			Adecuación Funcional	
		ff	tf	x1	e	te	x2	fu	tfu	x3	%	x
ACADEMIA	Estructura Académica Universitaria	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Distributivos	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	UTAm@tico Docentes	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Estudiantes Matriculas	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Estudiantes Calificaciones	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Evaluación Integral del Desempeño Docente	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Tutorías docentes	6	14	0,57	12	26	0,46	18	21	0,86	63%	Baja
ACADEMIA	Seguimiento al sílabo	2	26	0,92	20	36	0,56	69	84	0,82	77%	Media
ACADEMIA	Titulados	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Seguimiento a estudios de posgrado	3	7	0,57	0	16	0,00	9	9	1,00	52%	Baja
ACADEMIA	Posgrados	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Becas	7	10	0,30	6	10	0,60	10	14	0,71	54%	Baja
ACADEMIA	Matriculas Estudiantes Cultura Física e Idiomas	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Información Académica	1	14	0,93	14	21	0,67	11	24	0,46	68%	Baja
GESTIÓN	Hoja de Vida de Funcionarios	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Plan Estratégico de desarrollo institucional	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Infraestructura	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Vacaciones	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Convenios	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Reportes Información estratégica	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Business Intelligence	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Control Docente	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Evalúate	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Roles de pago	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Facturación electrónica	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Simulador de acreditación	79	79	0,00	0	0	0,00	0	0	0,00	0%	Baja
VINCULACIÓN	Seguimiento a graduados	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
VINCULACIÓN	Bolsa de empleo	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
VINCULACIÓN	Practicas preprofesionales	5	19	0,74	22	35	0,63	55	96	0,57	65%	Baja
VINCULACIÓN	Proyectos de vinculación	1	9	0,89	11	53	0,21	8	10	0,80	63%	Baja
VINCULACIÓN	Seguimiento de proyectos de Investigación	2	35	0,94	24	62	0,39	71	301	0,24	52%	Baja

Elaborado por: Investigador

4.2 Validación de las respuestas obtenidas

El formulario de evaluación y medición de resultados es aplicado de manera general a los sistemas desarrollados por la DITIC-UTA. Se utilizan métricas de calidad de la norma internacional ISO/IEC 25023, con el objetivo de evaluar el grado de funcionalidad del producto software.

4.2.1 Validación del formulario de evaluación y medición

En consideración con la norma internacional ISO/IEC 25023 se utilizó la siguiente métrica de calidad para calcular la característica de calidad de adecuación funcional:

$$\tilde{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Donde:

\tilde{x} = Adecuación funcional

$\sum_{i=1}^n x_i$ = Sumatoria de las sub características de adecuación funcional (completitud, exactitud y pertinencia).

n = Número de sub características de adecuación funcional evaluadas (3).

Valor deseado = El más cercano a 1 es mejor.

Ponderación de calidad: 1 equivale a 100%

Rango de calidad = Mayor a 85% Alto, mayor a 75% Medio, menor a 75% bajo.

4.2.2 Interpretación del resultado del formulario de evaluación y medición

Tabla 29. Resultado del formulario de evaluación y medición

Rango de calidad	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Alta	0	0	0	0
Media	1	3,2	3,2	3,2
Baja	30	96,8	96,8	100
Total	31	100	100	

Elaborado por: Investigador

La métrica de calidad obtenida que determina el grado de funcionalidad del producto software se encuentra en un 96,8% en el rango de calidad baja, apenas el restante 3,2% de los sistemas evaluados se encuentra en el rango de calidad media. No se encuentran sistemas desarrollados que cumplan un alto grado de funcionalidad.

CAPITULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

La información encontrada con la tabulación del formulario de evaluación y medición aplicado de manera general a los sistemas desarrollados por la DITIC-UTA, mostraron un bajo nivel de funcionalidad como característica de calidad de los productos software desarrollados con una metodología tradicional.

De acuerdo al levantamiento de información de cada software evaluado, un alto porcentaje de sistemas carece de documentación técnica como: especificación de requerimientos o diccionario de datos.

Ningún sistema desarrollado utiliza sistemas de control de versiones y no cuentan con un código fuente documentado.

Se debería mejorar los índices de funcionalidad como característica de calidad del producto software, estandarizando el proceso de desarrollo y cambiando la metodología de trabajo tradicional a una metodología ágil enfocada a la satisfacción del usuario final y el uso de buenas prácticas de programación.

No se encuentra estandarizado el proceso de desarrollo de software, debido a que no existe un proceso formal que contenga políticas, procedimientos o métricas de calidad que ayuden a definir el grado de funcionalidad del producto software desarrollado.

5.2 Recomendaciones

Con el propósito de lograr la satisfacción del usuario y los interesados, se debe incrementar la funcionalidad del producto software utilizando un marco de trabajo que permite interactuar con el usuario en todo momento, que durante el desarrollo permita realizar cambios y que los entregables contengan todas las funcionalidades necesarias que sean útiles al usuario final.

Se aplique metodologías o marcos de trabajo ágiles en departamentos de TI similares al de la DITIC-UTA; las metodologías ágiles priorizan la satisfacción del usuario, el producto software resultante puede ser validado mediante varios criterios o métricas de calidad.

Garantizar que el proceso de desarrollo describa políticas, técnicas, procedimientos y estándares basados en buenas prácticas de programación.

Para realizar el proceso de evaluación del producto software, se puede utilizar las características y sub características de calidad de la norma internacional ISO/IEC 25000, aplicando sus métricas de calidad de acuerdo a las necesidades institucionales.

CAPITULO VI

6 PROPUESTA

6.1 Datos Informativos

6.1.1 Título

Estandarización de los procesos de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil en departamentos de TI.

6.1.2 Institución ejecutara

Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato.

6.1.3 Beneficiarios

- Directores departamentales (Stakeholders)
- Especialistas/Coordinadores áreas TI
- Usuarios
- Analistas/Desarrolladores
- Asistentes/Tester

6.1.4 Ubicación

Ambato – Tungurahua

6.1.5 Responsable

Ing. Jorge Vladimir Chávez Andrade

6.1.6 Director

Ing. Darío Javier Robayo Jácome, MSc.

6.2 Antecedentes de la propuesta

La DITIC representa un eje transversal en la cadena de valor de la universidad, sin embargo, el proceso de desarrollo de software carece de estándares, buenas prácticas de programación, guías o procedimientos.

La presente propuesta pretende estandarizar el proceso de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil.

6.3 Justificación

Como resultado del análisis realizado sobre el proceso actual de desarrollo de software en la DITIC-UTA, se puede identificar la necesidad de implementar políticas, normas o procedimientos que permitan estandarizar el proceso de desarrollo de software.

Además, es necesario detallar estándares basados en buenas prácticas de programación, tales como: documentar el código fuente, utilizar entornos de desarrollo integrado actuales que tabulen el código fuente automáticamente y aplicar un sistema de control de versiones.

6.4 Objetivos

6.4.1 Objetivo general

Definir un marco de trabajo ágil, utilizando buenas prácticas de programación, que mejor se adapten a las necesidades del área de desarrollo de software de los departamentos de TI como la DITIC-UTA.

6.4.2 Objetivos específicos

- Especificar el proceso, políticas, técnicas y procedimientos para el desarrollo de software.
- Detallar estándares basados en buenas prácticas de programación.
- Aplicar el marco de trabajo propuesto en un proyecto de desarrollo actual en la Dirección de Tecnología de Información de la Universidad Técnica de Ambato.

6.5 Análisis de factibilidad

6.5.1 Factibilidad operativa

Se cuenta con la aceptación del Director de la DITIC-UTA, con lo cual se tiene las facilidades requeridas para obtener toda la información que sea necesaria, así como la asignación y apoyo del recurso humano que interviene en el desarrollo.

6.5.2 Factibilidad económica

La propuesta no representa un costo adicional para la universidad, debido a que se replantean las mismas actividades asignadas a los desarrolladores de un modelo tradicional a un modelo ágil.

6.5.3 Factibilidad técnica

Para la aplicación del modelo propuesto en un proyecto de desarrollo de software en la DITIC-UTA, se cuenta con equipos tecnológicos con las siguientes características técnicas:

Tabla 30. Especificaciones técnicas Equipos de desarrollo

Equipos de desarrollo	
Marca	Dell Inc.
Modelo	OptiPlex 7050
Procesador	Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz, 3600 Mhz, 4 Core(s)
Memoria RAM	8,00 GB
Disco Duro	931,51 GB
Sistema Operativo	Microsoft Windows 10 Pro

Elaborado por: Investigador

Tabla 31. Especificaciones técnicas Servidor de pruebas y versionamiento

Servidor de pruebas y versionamiento	
Marca	INTEL
Modelo	Intel Corp. TCIBX10H
Procesador	Intel(R) Core(TM) i7 CPU 860 @ 2.80GHz, 2793 Mhz, 1 Core(s)
Memoria RAM	8,00 GB
Disco Duro	309,51 GB

Servidor de pruebas y versionamiento	
Sistema Operativo	Microsoft Windows Server 2016 Datacenter
Gestor de Base de Datos	Microsoft SQL Server 2017 Developer
Gestor de Cambios	Microsoft Visual Studio Team Foundation Server 2018 Update 2

Elaborado por: Investigador

Tabla 32. Especificaciones técnicas Servidor de base de datos de pruebas

Servidor de base de datos de pruebas	
Tipo	Virtualizado
Host	Oracle VM VirtualBox 5.2.4 r119785
Procesador	2 Core(s)
Memoria RAM	4,00 GB
Disco Duro	100 GB
Sistema Operativo	CentOS 7
Gestor de Base de Datos	Microsoft SQL Server 2017 for Linux

Elaborado por: Investigador

Tabla 33. Especificaciones técnicas Servidor de base de datos

Servidor de base de datos	
Marca	LENOVO
Modelo	System x3650 M5: -[5462AC1]-
Procesador	Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz, 2401 Mhz, 6 Core(s)
Memoria RAM	63,4 GB
Disco Duro	2,18 TB
Sistema Operativo	Microsoft Windows Server 2016 Datacenter
Gestor de Base de Datos	Microsoft SQL Server 2016 Enterprise

Elaborado por: Investigador

Tabla 34. Especificaciones técnicas Servidor espejo

Servidor espejo	
Marca	LENOVO
Modelo	System x3650 M5: -[5462AC1]-
Procesador	Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz, 2601 Mhz, 8 Core(s)
Memoria RAM	63,6 GB
Disco Duro	3,63 TB
Sistema Operativo	Microsoft Windows Server 2016 Datacenter
Gestor de Base de Datos	Microsoft SQL Server 2016 Enterprise

Elaborado por: Investigador

6.5.4 Factibilidad legal

La propuesta está en concordancia a las diferentes leyes, estatutos y reglamentos que rigen a instituciones públicas y universidades.

6.6 Fundamentación

6.6.1 Entornos de desarrollo integrado IDE's

La Tabla 35 muestra las características principales de los entornos de desarrollo actuales que serán considerados para ser utilizados para la codificación:

Tabla 35. Características de los Entornos de desarrollo integrado (IDE's)

Entornos de desarrollo integrado (IDE's)	Características
Visual Studio Code	Poderoso editor de código fuente que puede ser instado en sistemas operativos Windows, macOS y Linux. Al ser muy liviana es una buena alternativa a Visual Studio. Soporta varios lenguajes de programación como C++, C#, Java, Python, PHP, etc. Posee integración con sistemas de versionamiento como GIT y TFS. (Microsoft, 2018)
Microsoft Visual Studio	Robusto entorno de desarrollo empresarial integrado con grandes características para desarrollo, depuración, pruebas y colaboración. Herramienta licenciada en sus versiones Professional y Enterprise, además posee una versión libre para estudiantes con licencia individual llamada Community. (Microsoft, 2018)
Sublime Text 3	Liviano editor de texto para código fuente muy utilizado con software libre. Al igual que VS Code puede ser utilizado en sistemas operativos Windows, macOS y Linux. Es altamente configurable, posee una gran variedad de plugins para interpretar varios lenguajes de programación, además soporta integración con GIT y TFS. (Skinner, 2018)
Notepad ++	Es un editor de código que puede ser descargado de forma gratuita pero solo soportado por sistemas operativos Windows. Muy liviano, reduce el consumo de procesador, y junto a la característica de autocompletado logra que la programación sea fluida. (Don Ho, 2018)
Atom	Editor de código fuente que puede ser instalado en varios sistemas operativos. Es distribuido bajo licencia de código abierto y soporta una gran cantidad de lenguajes como: HTML, CSS, C/C++, C#, Java, Javascript, Python, PHP, Ruby, etc. (GitHub Inc, 2018)
NetBeans	Es un entorno de desarrollo integrado especialmente desarrollado para Java; sin embargo, se pueden construir aplicaciones de escritorio, móviles o web con Java, JavaScript, HTML5, PHP, C/C++ y más. Es gratuito y de código abierto y se está integrando a Apache Software Foundation. (Apache Software Foundation, 2018)
Eclipse	Desarrollado originalmente por IBM, Eclipse es una plataforma de herramientas de programación multipropósito. Cuenta con varios proyectos orientados a desarrollo, modelamiento, testing, etc. (Eclipse Foundation, 2018)

Elaborado por: Investigador

6.6.2 Métricas de calidad del producto software para Funcionalidad

Las métricas de software comprenden varias actividades las cuales incluyen: cierto grado de medición, recopilación de datos, modelos, medidas de calidad, entre otros (Fenton & Bieman, 2015). Las métricas de calidad del producto software permiten definir el grado en que las características, atributos o factores de calidad se encuentra en el producto software.

Medir las características de calidad es la clave para obtener un producto software de mejor calidad (Genero, Piattini, & Calero, 2005). Los modelos y estándares de calidad ofrecen la posibilidad de convertir la calidad en algo tangible, que se pueda definir, medir y planificar (Lopez, Sánchez, & Montejano, 2016).

La división de la norma ISO/IEC 25000 que define las métricas para la medición de la calidad del producto software es la norma ISO/IEC 25023 (ISO/IEC, 2016). Las métricas para Adecuación Funcional consideran las subcaracterísticas de calidad: completitud, exactitud y pertinencia, estas se detallan en la Tabla 36:

Tabla 36. Métricas de calidad para adecuación funcional

Nombre	Descripción	Fórmula
Completitud	¿Qué tan completa es la implementación de acuerdo a las especificaciones de los requerimientos?	$X = 1 - A/B$ A = Número de funciones que faltan o están incorrectamente implementadas, que han sido detectadas en la evaluación. B = Número de funciones establecidas en la especificación de los requerimientos.
Exactitud	¿Cuánto del estándar requerido de exactitud se cumple?	$X = A/B$ A = Número de elementos de datos implementados con el estándar específico de exactitud B = Número total de elementos de datos implementados
Pertinencia	¿Qué parte de las funciones implementadas son percibidas como idóneas?	$X = A / B$ A = Número de funciones realmente útiles para realizar tareas específicas. B = Número de funciones implementadas para la consecución de tareas específicas

Fuente: (ISO/IEC, 2016)

6.7 Elaboración de la propuesta

Estandarización de los procesos de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil para incrementar el grado de funcionalidad del producto software

6.7.1 Ficha del proceso

Se definirá los lineamientos necesarios que debe tener la caracterización del proceso:

Tabla 37. Ficha del Proceso de Gestión de Desarrollo de Software utilizando Scrum

Ficha del Proceso de Gestión de Desarrollo de Software utilizando Scrum	
MacroProceso:	Gestión de Tecnología de Información y Comunicación
Proceso:	Gestión de Desarrollo de Software
Código del Proceso:	UTA-DITIC-DS
Descripción:	Determinar los mecanismos que faciliten la formulación, ejecución y administración de proyectos de desarrollo de software
Entradas:	Solicitud de Requerimientos de nuevo desarrollo.
Productos/Servicios del Proceso:	<ul style="list-style-type: none">• Software operativo• Documento de especificaciones del sistema• Historias de usuario• Tareas de ingeniería
Tipo de Proceso:	Adjetivo (Soporte)
Responsable del Proceso:	Coordinado del Área de Gestión de Desarrollo
Tipo de cliente:	Interno
Marco Legal:	Reglamento Orgánico de Gestión Organizacional por Procesos de la Universidad Técnica de Ambato.

Elaborado por: Investigador

6.7.2 Alcance del proceso

El proceso de Gestión de Desarrollo inicia con la solicitud de requerimientos de un nuevo desarrollo de software por parte de las unidades académicas o administrativas de la Universidad Técnica de Ambato y finaliza con el software operativo y entrega de documentación técnica.

6.7.3 Políticas del proceso

El proceso de Gestión de Desarrollo requiere considerar las siguientes políticas en consideración de leyes y estatutos que rigen a instituciones públicas y universidades, así como: la Constitución de la República del Ecuador, Ley Orgánica de Educación Superior, Normas de Control Interno de la Contraloría General del Estado, Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, Orgánico Funcional de la Universidad Técnica de Ambato, Estructura Interna de la Dirección de Tecnología de Información y Comunicación:

- El desarrollo se hará de acuerdo con el marco de trabajo y estándares de la DITIC, mismos que se encuentran establecidos en el documento “Estándares de Desarrollo de Software” del numeral 6.8 de la presente propuesta.
- La solicitud de desarrollo deberá ser formalmente presentada por los directores departamentales de las unidades administrativas o académicas que lo requieran.
- Las solicitudes de desarrollo deberán ser evaluadas y aprobadas por el Director de la DITIC, de acuerdo a las prioridades de los desarrollos actuales.
- Se deberá elaborar un acta que contenga las especificaciones del sistema, la misma que estará suscrita por los directores departamentales.
- Se brindarán capacitaciones permanentes al personal técnico y de soporte con el objetivo de mantenerlo comprometido con sus actividades.
- Previo al desarrollo, se deberá firmar un acta de compromiso para afianzar su responsabilidad con todo el grupo de trabajo.
- Las etapas del desarrollo estarán definidas en el documento de “Estándares de Desarrollo de Software” del numeral 6.8 y en el “Diagrama de flujo del proceso” del numeral 6.7.4 de la presente propuesta.
- El proceso de pruebas estará a cargo del Área de Calidad de la DITIC
- Se deberá documentar y mantener actualizados el manual técnico y de usuario.
- Al finalizar las etapas del desarrollo y concluir satisfactoriamente el proceso de pruebas, se debe parametrizar los perfiles, roles y usuarios del sistema.
- Se procederá a capacitar a los usuarios finales y entregar el manual de usuario elaborado.

- Todo el proceso de desarrollo estará en un ambiente de pruebas/desarrollo, pasando a un ambiente de producción, en la fase de despliegue o implementación, posterior a haber superado el proceso de pruebas y la entrega formal al usuario.
- La propiedad intelectual del software desarrollado está en concordancia con lo dispuesto en los Artículo 131 y Artículo 133 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación.

6.7.4 Diagrama de flujo del proceso

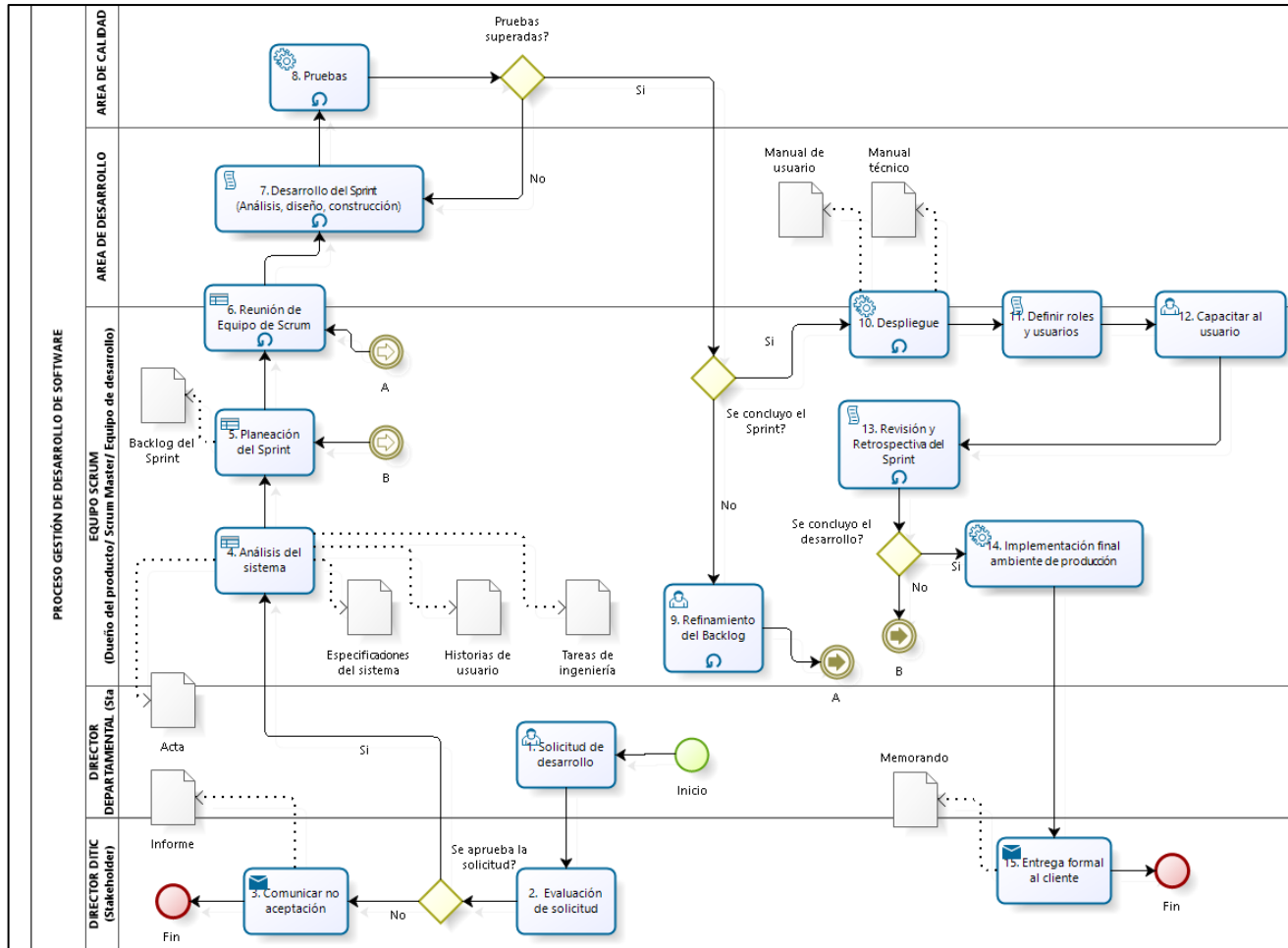


Figura 9. Proceso de Gestión de Desarrollo de Software utilizando Scrum

Elaborado por: Investigador

6.7.5 Descripción de actividades del proceso

Tabla 38. Descripción de actividades del proceso

#	Responsable	Actividad	Descripción	Documento
1	DIRECTOR DEPARTAMENTAL (Stakeholder)	1. Solicitud de desarrollo	Los directores departamentales presentan formalmente la solicitud de desarrollo de software	Solicitud
2	DIRECTOR DITIC (Stakeholder)	2. Evaluación de solicitud	Se evalúa la solicitud de acuerdo a prioridades de desarrollos actuales	
3	DIRECTOR DITIC (Stakeholder)	3. Comunicar no aceptación	De no ser factible el desarrollo se envía comunica un informe con la no aceptación	Informe
4	EQUIPO SCRUM (Dueño del producto/ Scrum Master/ Miembros del equipo de desarrollo)	4. Análisis del sistema	De ser favorable el desarrollo, todo el equipo Scrum mediante un trabajo conjunto se reúnen para recopilar información y generar la documentación necesaria.	Especificaciones del sistema Historias de usuario Tareas de ingeniería
5	EQUIPO SCRUM	5. Planeación del Sprint	Se especifica las tareas de ingeniería de las historias de usuario que contemplará el Sprint	Backlog del Sprint
6	EQUIPO SCRUM AREA DE DESARROLLO	6. Reunión de Equipo de Scrum	Reunión diaria para sincronizar actividades	
7	AREA DE DESARROLLO	7. Desarrollo del Sprint (Análisis, diseño, construcción)	Se desarrolla cada elemento de trabajo pendiente	
8	AREA DE CALIDAD	8. Pruebas	El área de calidad es la encargada de ejecutar todo el proceso de pruebas	
9	EQUIPO SCRUM USUARIOS (Product Owner)	9. Refinamiento del Backlog	Se revisa cada elemento de trabajo pendiente y se explórese cualquier duda.	

#	Responsable	Actividad	Descripción	Documento
10	EQUIPO SCRUM AREA DE DESARROLLO	10. Despliegue	De haber concluido el Sprint satisfactoriamente, se realiza el despliegue del entregable correspondiente al Sprint terminado.	Manual de usuario Manual técnico
11	EQUIPO SCRUM AREA DE DESARROLLO	11. Definir roles y usuarios	Se definen los roles y usuarios que van a utilizar el sistema	
12	EQUIPO SCRUM AREA DE DESARROLLO	12. Capacitar al usuario	Se brinda una capacitación al usuario final	
13	EQUIPO SCRUM (Dueño del producto/ Scrum Master/ Miembros del equipo de desarrollo)	13. Revisión y Retrospectiva del Sprint	Al finalizar el Sprint se realiza las reuniones de retroalimentación del Sprint	
14	EQUIPO SCRUM (Dueño del producto/ Scrum Master/ Miembros del equipo de desarrollo)	14. Implementación final ambiente de producción	Al haber terminado satisfactoriamente las fases del desarrollo, el software completo pasa del ambiente de desarrollo al de producción	
15	DIRECTOR DITIC (Stakeholder)	15. Entrega formal al cliente	Se realiza la entrega formal del sistema a la unidad solicitante	Memorando

Elaborado por: Investigador

6.7.6 Indicadores del proceso

Con el objetivo de incrementar la funcionalidad del producto software resultante al proceso de desarrollo, se considerará las métricas de calidad de adecuación funcional según la norma internacional ISO/IEC 25023.

Tabla 39. Indicadores del proceso

Nombre	Descripción	Objetivo	Fórmula
Compleitud funcional	Grado en el que el conjunto de funcionalidades cubre todas las tareas y objetivos especificados por el usuario. - ¿Qué tan completa es la implementación de acuerdo a las especificaciones de los requerimientos?	Asegurar el cumplimiento de las necesidades del cliente.	$x_1 = 1 - ff/tf$ <p><i>ff</i> = Número de funciones que faltan o están incorrectamente implementadas, que han sido detectadas en la evaluación. <i>tf</i> = Número de funciones establecidas en la especificación de los requerimientos.</p>
Exactitud funcional	Capacidad del producto software para proveer resultados correctos con el nivel de precisión necesario. - ¿Cuánto del estándar requerido de exactitud se cumple?	Lograr la reducción de errores en el sistema en producción.	$x_2 = e/te$ <p><i>e</i> = Número de elementos de datos implementados con el estándar específico de exactitud <i>te</i> = Número total de elementos de datos implementados</p>
Pertinencia funcional	Capacidad del producto software para facilitar la realización de las tareas y los objetivos especificados por el usuario. - ¿Qué parte de las funciones implementadas son percibidas como idóneas?	Mejorar la confianza y satisfacción del cliente.	$x_3 = fu/tfu$ <p><i>fu</i> = = Número de funciones realmente útiles para realizar tareas específicas. <i>tfu</i> = Número de funciones implementadas para la consecución de tareas específicas</p>

Nombre	Descripción	Objetivo	Fórmula
Adecuación Funcional	Capacidad del producto software para proporcionar funciones que satisfacen las necesidades, declaradas e implícitas, cuando el producto se usa en un entorno específico	Satisfacer las necesidades implícitas y explícitas del producto software esperado.	$\tilde{x} = \frac{\sum_{i=1}^n x_i}{n}$ <p>$\sum_{i=1}^n x_i$ = Sumatoria de las sub características de adecuación funcional (completitud, exactitud y pertinencia). n = Número de sub características de adecuación funcional evaluadas.</p> <p>Valor deseado: El más cercano a 1 es mejor.</p> <p>Ponderación de calidad: 1 equivale a 100%</p> <p>Rango de calidad: Mayor a 85% Alto Mayor a 75% Medio Menor a 75% Bajo.</p>

Fuente: Métricas de calidad de software ISO/IEC 25023 (ISO/IEC, 2016)

Elaborado por: Investigador

6.8 Estándares de Desarrollo de Software

El presente manual de estándares de desarrollo de software considera a Scrum como marco de trabajo, donde se definen roles, actividades y herramientas que permiten el trabajo colaborativo entre todos los involucrados en el proceso de desarrollo.

6.8.1 Roles dentro del Equipo Scrum (Scrum Team)

El equipo Scrum está conformado por los siguientes roles:

6.8.1.1 Interesados (Stakeholders)

Los directores departamentales asumen el rol de “interesados”, ellos velarán porque el proyecto de desarrollo se ejecute con normalidad.

6.8.1.2 Dueño del producto (Product Owner)

Un representante de los funcionarios de la unidad departamental solicitante asume el rol de “dueño del producto” quien intervendrá constantemente con la retroalimentación durante el desarrollo del software.

6.8.1.3 Scrum Master

El rol de “Scrum master” lo desempeña el Coordinador del Área de Desarrollo, Jefe de Proyectos o un Especialista de la DITIC, quien deberá estar capacitado en el uso del marco de trabajo Scrum.

6.8.1.4 Miembros del equipo de desarrollo (Development Team Members)

El equipo de desarrollo está conformado por Analistas y/o Asistentes de Tecnología, quienes son los encargados de escribir y probar el código.

6.8.2 Etapas del desarrollo (Sprint)

El Sprint es un ciclo repetitivo de duración constante durante el cual se desarrolla un incremento que sea entregable. Se definen las siguientes etapas dentro del Sprint:

6.8.2.1 Planeación del Sprint (Sprint Planning)

En las actividades de análisis de sistemas intervendrán todo el equipo Scrum, que mediante un trabajo conjunto deberá recopilar información técnica para su análisis y posterior elaboren la siguiente documentación:

- Documento de especificaciones del sistema
- Historias de usuario
- Tareas de ingeniería
- Bitácoras de reuniones

La elaboración del documento de especificaciones debe contener un alto nivel de detalle técnico considerando:

- Incremento, entregables o funcionalidad.
- Módulos o funcionalidades de forma jerárquica.
- Campos o atributos de entidades, incluyendo tipo de dato, tamaño, origen, validación, etc.

6.8.2.2 Reunión de Equipo de Scrum (Scrum team meeting)

Es una reunión diaria de trabajo de 15 minutos, con el objetivo de dar a conocer las actividades realizadas el día anterior, sincronizar las actividades a ser elaboradas el presente día e indicar si existe algún obstáculo o percance que deba ser solucionado.

6.8.2.3 Refinamiento del Backlog (Backlog Refinement)

El dueño del producto revisa cada uno de los elementos de trabajo pendiente, donde debe aclarar cualquier duda que tenga el equipo de desarrollo. También se puede volver a estimar tiempos a cada requerimiento.

6.8.2.4 Revisión del Sprint (Sprint Review)

Esta reunión se lleva a cabo al finalizar el Sprint con todo el equipo Scrum, su objetivo es inspeccionar el incremento que contiene los elementos de trabajo realizados.

6.8.2.5 Retrospectiva del Sprint (Retrospective)

Es una reunión de retroalimentación con el objetivo de identificar: qué se hizo mal, qué se hizo bien, y cuáles fueron los inconvenientes encontrados durante el Sprint: con ello crear un plan de mejoras que se ejecutará en el siguiente Sprint.

Se deberá llevar una bitácora de reuniones que debe ser llenada inmediatamente culminada la reunión; indicando fecha, hora, objetivo, actividades, resultado y participantes, las mismas que serán ingresadas en el formato del Anexo B.

6.8.3 Herramientas de Scrum

6.8.3.1 Backlog de Producto (Product Backlog)

Son todos los elementos de trabajo pendientes que incluyen a tareas de ingeniería como a errores encontrados.

6.8.3.2 Historias de Usuario (User Stories)

Las historias de usuario deben indicar a que Sprint y funcionalidad pertenecen, las mismas que serán ingresadas en el formato del Anexo C.

6.8.3.3 Backlog del Sprint (Sprint Backlog)

Son los elementos de trabajo pendientes que fueron priorizados y estimados, e identifican las tareas de ingeniería definidas por cada historia de usuario. Las estimaciones de las tareas de ingeniería se las realizarán en horas, asignando un responsable e indicando a que historia de usuario pertenecen, las mismas que serán ingresadas en el formato del Anexo D.

6.8.3.4 El panel de Tareas (The Taskboard)

La planificación y seguimiento de los elementos de trabajo pendientes se los gestionará mediante un tablero dividido en columnas que indica el estado de los elementos de trabajo, estos estados son: pendiente, en curso, terminado. Cuando las tareas están en estado terminado, estas se encuentran lista para que el personal de calidad realice las pruebas respectivas.

6.8.4 Buenas prácticas de programación

6.8.4.1 Estándares de documentación

Se definen reglas normalizadas para garantizar la mantenibilidad del software, con ello obtener un código más sencillo de entender para su futura actualización.

6.8.4.1.1 Documentación técnica

Se establece el formato a ser utilizado en toda documentación técnica generada durante el proyecto de desarrollo, así como su ubicación y accesibilidad.

6.8.4.1.1.1 Formato

Toda documentación técnica presenta el siguiente formato:

TITULO 1: CALIBRI 12ptos NEGRITA MAYUSCULAS CENTRADO

Titulo 2: Calibri 11ptos Negrita No centrado

Texto: Calibri 11ptos Normal Justificado Interlineado sencillo

6.8.4.1.1.2 Ubicación

Toda información se almacena en un directorio compartido en la nube de Microsoft (OneDrive para la empresa), suscripción que mantiene la institución.

6.8.4.1.2 Estructura directorios

El proyecto de desarrollo debe ser organizado por tipo de archivo o semejanza, agrupando en directorios específicos para hojas de estilos, código fuente, imágenes, documentación, archivos de configuración, etc.

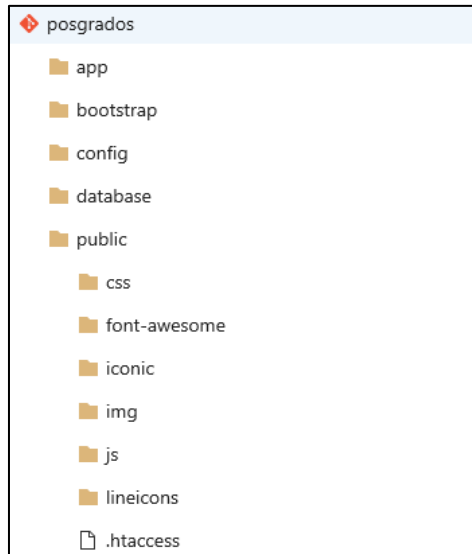


Figura 10. Estructura de directorios desarrollo interno

Elaborado por: Investigador

6.8.4.1.3 Nombres de objetos

El nombre de las variables, procedimientos, funciones, clases o cualquier objeto debe ser completo e indicar por sí mismo que representa, almacena, función que realiza u objetivo programático.

6.8.4.1.4 Archivos

El nombre de cada archivo dentro del proyecto software debe representar claramente su funcionalidad, objetivo o nombre de tabla con la que interactúa mayormente o representa.

6.8.4.2 Documentación de Código

Se tiene que documentar claramente el código fuente y todo objeto generado en la base de datos, documentación mínima necesaria que permita comprender su estructura física y lógica, así como su contenido.

Se debe utilizar entornos de desarrollo integrados (IDE's) que permitan agilizar el proceso de codificación, así como tabular el código fuente para su mejor entendimiento. Los IDE's a ser utilizados en consideración de la Tabla 40 para la programación del código fuente son:

Tabla 40. Entornos de desarrollo integrado (IDE's) considerados para el desarrollo

Entornos de desarrollo integrado (IDE's)		Lenguaje de programación
Principal	Visual Studio Code	Varios lenguajes (PHP, .NET, JAVA, etc.)
Secundario	Microsoft Visual Studio	Tecnología Microsoft (.Net, ASP, etc.)

Elaborado por: Investigador

6.8.4.3 Tabulación

Se debe utilizar la funcionalidad que brindan los IDE's actuales para organizar y tabular jerárquicamente el código de forma automática, con el objetivo de visualizar la programación realizada de forma clara y fácil de entender.

6.8.4.4 Control de versiones

Es muy importante contar con un control de versiones de código que permita un desarrollo colaborativo, además del versionamiento del código fuente para poder acceder a un punto específico de la programación cuando sea necesario.

El modelo de trabajo para llevar el sistema de control de versiones es el siguiente:

- Asignar un usuario del sistema de versionamiento por cada Analista de TI que ocupa el rol de “desarrollador/programador” dentro del equipo de desarrollo.
- Obtener una copia del repositorio de versionamiento.
- Actualizar la versión local con los cambios realizados por otros desarrolladores.
- Guardar los cambios en el repositorio agregando una descripción del trabajo realizado.

El seguimiento de proyectos de desarrollo se lo gestiona a través de la herramienta Microsoft Team Foundation Server. Al momento de generar un nuevo proyecto, se especificará el gestor de versionamiento de acuerdo al siguiente detalle:

Tabla 41. Gestores de versionamiento

Gestores de versionamiento		Lenguaje de programación
Principal	GIT	Varios lenguajes (PHP, .Net, JAVA, etc.)
Secundario	Team Foundation Version Control	Tecnología Microsoft (.Net, ASP, etc.)

Elaborado por: Investigador

6.8.5 Anexos propuesta

6.8.5.1 Anexo A

Modelo de Carta de Compromiso de Integrar el Equipo de Scrum

(Ciudad, fecha)

Yo (nombre del integrante del equipo) con el cargo (Cargo que desempeña en la universidad), declaro que formaré parte integral del Equipo Scrum en calidad de (Rol: Dueño del producto/ Scrum Master/ Miembro del equipo de desarrollo) del proyecto de desarrollo (Nombre del proyecto) aportando de manera positiva y profesional al desarrollo del mismo, aplicando los valores de Scrum para garantizar su correcta ejecución y efectiva culminación.

Firma del integrante del equipo Scrum

6.8.5.2 Anexo B

Tabla 42. Formato Retrospectiva del Sprint

RETROSPECTIVA DEL SPRINT	
Fecha: Fecha de la reunión	Hora: Hora de la reunión
Objetivo: Objetivo de la reunión de Retrospectiva del Sprint	
Actividades: Se deben especificar las actividades a desarrollarse toda vez que haya finalizado el Sprint	
Resultado: Se expone los resultados obtenidos, puntos a mejorar o conclusiones resultantes de las actividades desarrolladas en la reunión de retrospectiva de Sprint	
Participantes: Todo el equipo Scrum	

Elaborado por: Investigador

6.8.5.3 Anexo C

Tabla 43. Formato Historias de usuario

Historias de usuario	
Número: Permite identificar a una historia de usuario	Usuario: Persona que utilizará la funcionalidad del sistema descrita en la historia de usuario.
Nombre Historia: Describe de manera general a una historia de usuario.	
Prioridad en Negocio: Grado de importancia que el cliente asigna a una historia de usuario.	Riesgo en Desarrollo: Valor de complejidad que una historia de usuario representa al equipo de desarrollo.
Puntos Estimados: Número de semanas que se necesitará para el desarrollo de una historia de usuario	Iteración Asignada: Número de iteración, en que el cliente desea que se implemente una historia de usuario
Programador Responsable: Persona encargada de programar cada historia de usuario.	
Descripción: Información detallada de una historia de usuario.	
Observaciones: Campo opcional utilizado para aclarar, si es necesario, el requerimiento descrito de una historia de usuario	

Fuente: (Ibarra, 2018)

6.8.5.4 Anexo D

Tabla 44. Formato Tarea de ingeniería

Tarea de ingeniería	
Número de tarea: Servirá para identificar a una tarea de ingeniería.	Número de historia: Número o código asignado a la historia correspondiente.
Nombre de Tarea: Se describe de manera general a una tarea de ingeniería.	
Tipo de tarea: Tipo al que corresponde la tarea de ingeniería	Puntos estimados: Cantidad o número de horas que se necesitará para el desarrollo de la tarea de ingeniería.
Fecha inicio: Fecha inicial de la creación de la tarea.	Fecha fin: Fecha de finalización de la tarea.
Programador responsable: Desarrollador encargado de programar la tarea.	
Descripción: Información detallada de la tarea que se realiza.	

Fuente: (Ibarra, 2018)

6.9 Modelo Operativo

Aplicación de la presente propuesta al desarrollo del Sistema de Acciones de Personal

El sistema de acciones de personal está desarrollado con la finalidad de sistematizar y agilizar el proceso de generación de los diferentes tipos de acciones de personal como: nombramientos, renunciaciones, vacaciones, maternidad, enfermedad, lactancia, etc.

6.9.1 Capacitación

Previo al inicio del proyecto que aplicó el marco de trabajo propuesto y con la finalidad de contar con un equipo de desarrollo motivado, involucrado y preparado profesionalmente; se brindó una capacitación sobre Scrum a través de la Escuela Superior de Redes RED CEDIA con una duración de 40 horas, evento realizado en el mes de mayo de 2018.

6.9.2 Roles dentro del Equipo Scrum

El equipo Scrum está conformado por los roles que se muestran en la Tabla 45:

Tabla 45. Roles dentro del Equipo Scrum

PERSONA	CARGO	ROL
Investigador	Especialista de TI	Scrum Master
Desarrollador 1	Analista de TI	Miembro del equipo de desarrollo Scrum
Desarrollador 2	Analista de TI	Miembro del equipo de desarrollo Scrum
Desarrollador 3	Asistente de TI	Miembro del equipo de desarrollo Scrum
Usuario	Especialista de TTHH	Dueño del producto

Elaborado por: Investigador

Además, se considera a los directores departamentales tanto de la Dirección de Tecnología de Información y Comunicación como de la Dirección de Talento Humano

como los interesados del proyecto (Stakeholders), quienes velan para que el proyecto se ejecute con normalidad.

6.9.3 Análisis del sistema

En conjunto, todos los integrantes del equipo Scrum realizan el análisis del sistema, definiendo el documento de especificaciones que contiene las funcionalidades, historias de usuario y tareas de ingeniería.

6.9.3.1 Funcionalidades

Del documento de especificaciones resultante de la reunión de todos los involucrados en el proyecto, se puede identificar que, el desarrollo del sistema de acciones de personal consta de las siguientes funcionalidades:

- Seguridades y Parametrización
- Gestión de las Acciones de Personal
- Reportería e Integración con el SI-UTA

6.9.4 Backlog del Producto

El Backlog del Producto considera a todos los elementos de trabajo pendientes que son las historias de usuario y tareas de ingeniería.

6.9.4.1 Historias de Usuario

Las historias de usuario creadas de acuerdo al Anexo C, identifican las estimaciones a considerarse en la Planeación del Sprint; estas dan lugar a la generación de las tareas de ingeniería.

Tabla 46. Historia de usuario: Acceso al Sistema de Acciones de Personal

HISTORIA DE USUARIO	
Número: 1	Usuario: Usuarios Talento Humano
Nombre Historia: Acceso al Sistema de Acciones de Personal	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 0.25	Iteración Asignada: 1

HISTORIA DE USUARIO	
Programador Responsable: Equipo de Desarrollo	
Descripción: El acceso al sistema requiere controlar: Ingresos exitosos y fallidos, bloqueo, control de URL autorizado y sesión expirada.	

Elaborado por: Investigador

Tabla 47. Historia de usuario: Plantilla

HISTORIA DE USUARIO	
Número: 2	Usuario: Usuarios Talento Humano
Nombre Historia: Plantilla	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 0,6	Iteración Asignada: 1
Programador Responsable: Equipo de Desarrollo	
Descripción: Diseñar y crear la plantilla del sistema considerando la generación de hojas de estilo, imágenes personalizadas, plantilla HTML, e integración del diseño al aplicativo.	

Elaborado por: Investigador

Tabla 48. Historia de usuario: Asignación a los perfiles de usuarios

HISTORIA DE USUARIO	
Número: 3	Usuario: usuario DTH
Nombre Historia: Asignación a los perfiles de usuarios	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 0,8	Iteración Asignada: 1
Programador Responsable: Equipo de Desarrollo	
Descripción: Esta historia de usuario, permite el desarrollo de las interfaces para usuarios, perfiles y asignaciones de los perfiles de usuarios. Permite asignar también acceso al sistema (privilegios)	

Elaborado por: Investigador

Tabla 49. Historia de usuario: Parametrización de datos iniciales para Acciones de personal

HISTORIA DE USUARIO	
Número: 4	Usuario: Usuarios Talento Humano
Nombre Historia: Parametrización de datos iniciales para acciones de personal	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 1,5	Iteración Asignada: 1
Programador Responsable: Equipo de Desarrollo	
Descripción: Se necesita crear interfaces para la parametrización de datos iniciales para la gestión de acciones de personal.	

Elaborado por: Investigador

Tabla 50. Historia de usuario: Gestionar las Acciones de Personal

HISTORIA DE USUARIO	
Número: 5	Usuario: Usuarios Talento Humano
Nombre Historia: Gestionar las Acciones de Personal	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 2,1	Iteración Asignada: 1
Programador Responsable: Equipo de Desarrollo	
Descripción: Se requiere que el sistema permita registrar la información solicitada y se puede visualizar en un formato preestablecido la acción de personal generada.	

Elaborado por: Investigador

Tabla 51. Historia de usuario: Revisión y Aprobación

HISTORIA DE USUARIO	
Número: 6	Usuario: Usuarios Talento Humano
Nombre Historia: Revisión y Aprobación	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 1,1	Iteración Asignada: 1
Programador Responsable: Equipo de Desarrollo	
Descripción: Se necesita que el sistema gestione la revisión y aprobación de acciones de personal en consideración de los permisos de los perfiles de usuarios registrados en el sistema	

Elaborado por: Investigador

Tabla 52. Historia de usuario: Reportes

HISTORIA DE USUARIO	
Número: 7	Usuario: Usuarios Talento Humano
Nombre Historia: Reportes	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 1,8	Iteración Asignada: 2
Programador Responsable: Equipo de Desarrollo	
Descripción: Los reportes generados por el sistema consideran los formatos y campos de la dirección de talento humano	

Elaborado por: Investigador

Tabla 53. Historia de usuario: Integración del Sistema de acción de personal con el SIUTA

HISTORIA DE USUARIO	
Número: 8	Usuario: Usuarios Talento Humano
Nombre Historia: Integración del Sistema de acción de personal con el SIUTA	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 0,25	Iteración Asignada: 2

Programador Responsable: Equipo de Desarrollo
Descripción: Se realizará la integración del Sistema de acción de personal con el SIUTA

Elaborado por: Investigador

6.9.4.2 Tareas de Ingeniería

Las tareas de ingeniería forman parte de las historias de usuarios y dan lugar a los elementos de trabajo pendiente, las mismas que se encuentran elaboradas de acuerdo al Anexo D de la propuesta.

Tabla 54. Tarea de ingeniería. Ingreso Exitoso

TAREA DE INGENIERIA	
Número de tarea: 1	Número de historia: 1
Nombre de Tarea: Ingreso Exitoso	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Programador responsable: Equipo de Desarrollo	
Descripción: Se requiere verificar el acceso solo a usuarios autorizados	

Elaborado por: Investigador

Tabla 55. Tarea de ingeniería. Ingreso fallido

TAREA DE INGENIERIA	
Número de tarea: 2	Número de historia: 1
Nombre de Tarea: Ingreso fallido	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Programador responsable: Equipo de Desarrollo	
Descripción: Se niega el acceso a usuarios no autorizados	

Elaborado por: Investigador

Tabla 56. Tarea de ingeniería. Bloqueo

TAREA DE INGENIERIA	
Número de tarea: 3	Número de historia: 1
Nombre de Tarea: Bloqueo	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Programador responsable: Equipo de Desarrollo	
Descripción: Se realiza el bloqueo a usuarios no autorizados	

Elaborado por: Investigador

Tabla 57. Tarea de ingeniería. Control URL Autorizado

TAREA DE INGENIERIA	
Número de tarea: 4	Número de historia: 1

Nombre de Tarea: Control URL Autorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita controlar el acceso autorizado mediante URL	

Elaborado por: Investigador

Tabla 58. Tarea de ingeniería. Sesión Expirada

TAREA DE INGENIERIA	
Número de tarea: 5	Número de historia: 1
Nombre de Tarea: Sesión Expirada	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Programador responsable: Equipo de Desarrollo	
Descripción: El sistema cierra la sesión después de un intervalo de tiempo de inactividad	

Elaborado por: Investigador

Tabla 59. Tarea de ingeniería. Creación de hojas de estilo

TAREA DE INGENIERIA	
Número de tarea: 6	Número de historia: 2
Nombre de Tarea: Creación de hojas de estilo	
Tipo de tarea: Desarrollo	Puntos estimados: 7
Programador responsable: Equipo de Desarrollo	
Descripción: Se crean hojas de estilo que se aplican en general a la plantilla del sistema	

Elaborado por: Investigador

Tabla 60. Tarea de ingeniería. Creación de Imágenes

TAREA DE INGENIERIA	
Número de tarea: 7	Número de historia: 2
Nombre de Tarea: Creación de Imágenes	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Programador responsable: Equipo de Desarrollo	
Descripción: Se diseñan las imágenes necesarias para el sistema	

Elaborado por: Investigador

Tabla 61. Tarea de ingeniería. Creación Plantilla HTML

TAREA DE INGENIERIA	
Número de tarea: 8	Número de historia: 2
Nombre de Tarea: Creación Plantilla HTML	
Tipo de tarea: Desarrollo	Puntos estimados: 7
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita crear una plantilla principal HTML que toma como base cada formulario	

Tabla 62. Tarea de ingeniería. Integrar plantilla al aplicativo y verificación del diseño

TAREA DE INGENIERIA	
Número de tarea: 9	Número de historia: 2
Nombre de Tarea: Integrar plantilla al aplicativo y verificación del diseño	
Tipo de tarea: Desarrollo	Puntos estimados: 4
Programador responsable: Equipo de Desarrollo	
Descripción: Se integra la plantilla al sistema y se verifica su diseño	

Elaborado por: Investigador

Tabla 63. Tarea de ingeniería. Creación de usuarios

TAREA DE INGENIERIA	
Número de tarea: 10	Número de historia: 3
Nombre de Tarea: Creación de usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 4
Programador responsable: Equipo de Desarrollo	
Descripción: Se requiere que el sistema facilite la creación de usuarios verificados de la nómina de la universidad	

Elaborado por: Investigador

Tabla 64. Tarea de ingeniería. Eliminación de usuarios

TAREA DE INGENIERIA	
Número de tarea: 11	Número de historia: 3
Nombre de Tarea: Eliminación de usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 5
Programador responsable: Equipo de Desarrollo	
Descripción: Los usuarios pueden ser eliminados previo la verificación de que un usuario no se encuentre relacionado con información del sistema	

Elaborado por: Investigador

Tabla 65. Tarea de ingeniería. Asignación de Perfiles

TAREA DE INGENIERIA	
Número de tarea: 12	Número de historia: 3
Nombre de Tarea: Asignación de Perfiles	
Tipo de tarea: Desarrollo	Puntos estimados: 6
Programador responsable: Equipo de Desarrollo	
Descripción: Debe permitir la asignación de uno o varios perfiles de usuario de acuerdo a las necesidades del cliente	

Elaborado por: Investigador

Tabla 66. Tarea de ingeniería. Acceso a gestionar diferentes tipos de acción de personal

TAREA DE INGENIERIA	
Número de tarea: 13	Número de historia: 3

Nombre de Tarea: Acceso a gestionar diferentes tipos de acción de personal	
Tipo de tarea: Desarrollo	Puntos estimados: 5
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita poder conceder diferentes accesos para gestionar los diferentes tipos de acción de personal ingresados.	

Elaborado por: Investigador

Tabla 67. Tarea de ingeniería. Asignación de carreras y departamentos

TAREA DE INGENIERIA	
Número de tarea: 14	Número de historia: 3
Nombre de Tarea: Asignación de especialidades y departamentos	
Tipo de tarea: Desarrollo	Puntos estimados: 6
Programador responsable: Equipo de Desarrollo	
Descripción: Es necesario poder asignar a un usuario el acceso a generar acciones de personal de una o varias especialidades o departamentos.	

Elaborado por: Investigador

Tabla 68. Tarea de ingeniería. Gestión del estado de la acción de personal

TAREA DE INGENIERIA	
Número de tarea: 15	Número de historia: 3
Nombre de Tarea: Gestión del estado de la acción de personal	
Tipo de tarea: Desarrollo	Puntos estimados: 6
Programador responsable: Equipo de Desarrollo	
Descripción: Se requiere controlar el acceso para actualizar los diferentes estados que puede tener una acción de personal.	

Elaborado por: Investigador

Tabla 69. Tarea de ingeniería. Denegar accesos a usuarios - roles

TAREA DE INGENIERIA	
Número de tarea: 16	Número de historia: 3
Nombre de Tarea: Denegar accesos a usuarios - roles	
Tipo de tarea: Desarrollo	Puntos estimados: 6
Programador responsable: Equipo de Desarrollo	
Descripción: Cada módulo o formulario debe controlar el acceso al mismo de acuerdo al rol asignado al usuario.	

Elaborado por: Investigador

Tabla 70. Tarea de ingeniería. Verificación de perfiles de usuario

TAREA DE INGENIERIA	
Número de tarea: 17	Número de historia: 3
Nombre de Tarea: Verificación de perfiles de usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Programador responsable: Equipo de Desarrollo	
Descripción: Se debe verificar y mostrar los permisos correspondientes al perfil asignado al usuario	

Tabla 71. Tarea de ingeniería. Tipo de acciones de personal

TAREA DE INGENIERIA	
Número de tarea: 18	Número de historia: 4
Nombre de Tarea: Tipo de acciones de personal	
Tipo de tarea: Desarrollo	Puntos estimados: 8
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita que el sistema permita la gestión de los tipos de acción de personal	

Elaborado por: Investigador

Tabla 72. Tarea de ingeniería. Autorizado por

TAREA DE INGENIERIA	
Número de tarea: 19	Número de historia: 4
Nombre de Tarea: Autorizado por	
Tipo de tarea: Desarrollo	Puntos estimados: 8
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita que el sistema permita la gestión del este campo de selección al momento de generar una acción de personal	

Elaborado por: Investigador

Tabla 73. Tarea de ingeniería. Solicitado por

TAREA DE INGENIERIA	
Número de tarea: 20	Número de historia: 4
Nombre de Tarea: Solicitado por	
Tipo de tarea: Desarrollo	Puntos estimados: 8
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita que el sistema permita la gestión del este campo de selección al momento de generar una acción de personal	

Elaborado por: Investigador

Tabla 74. Tarea de ingeniería. Grado Académico

TAREA DE INGENIERIA	
Número de tarea: 21	Número de historia: 4
Nombre de Tarea: Grado Académico	
Tipo de tarea: Desarrollo	Puntos estimados: 8
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita que el sistema permita la gestión del este campo de selección al momento de generar una acción de personal	

Elaborado por: Investigador

Tabla 75. Tarea de ingeniería. Firmas

TAREA DE INGENIERIA

Número de tarea: 22	Número de historia: 4
Nombre de Tarea: Firmas	
Tipo de tarea: Desarrollo	Puntos estimados: 8
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita que el sistema permita la gestión del este campo de selección al momento de generar una acción de personal	

Elaborado por: Investigador

Tabla 76. Tarea de ingeniería. Gestión automática de campos

TAREA DE INGENIERIA	
Número de tarea: 23	Número de historia: 4
Nombre de Tarea: Gestión automática de campos	
Tipo de tarea: Desarrollo	Puntos estimados: 50
Programador responsable: Equipo de Desarrollo	
Descripción: Se requiere que el sistema permita la gestión automática de los campos: Régimen Laboral, Nivel Ocupacional, Escala Ocupacional, Unidad Orgánica, Modalidad Laboral, Denominación Puesto al momento de subir el distributivo	

Elaborado por: Investigador

Tabla 77. Tarea de ingeniería. Creación de acciones personal

TAREA DE INGENIERIA	
Número de tarea: 24	Número de historia: 5
Nombre de Tarea: Creación de acciones de personal	
Tipo de tarea: Desarrollo	Puntos estimados: 52
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita contar con un formulario que permita la creación de acciones de personal	

Elaborado por: Investigador

Tabla 78. Tarea de ingeniería. Actualización y Validación de la Acción Personal

TAREA DE INGENIERIA	
Número de tarea: 25	Número de historia: 5
Nombre de Tarea: Actualización y validación de la Acción Personal	
Tipo de tarea: Desarrollo	Puntos estimados: 28
Programador responsable: Equipo de Desarrollo	
Descripción: Debe permitir una actualización y verificación de una acción de personal	

Elaborado por: Investigador

Tabla 79. Tarea de ingeniería. Carga masiva del distributivo de trabajo

TAREA DE INGENIERIA	
Número de tarea: 26	Número de historia: 5
Nombre de Tarea: Carga masiva del distributivo de trabajo	
Tipo de tarea: Desarrollo	Puntos estimados: 10
Programador responsable: Equipo de Desarrollo	

Descripción: Permite registrar el distributivo de trabajo en la base de datos

Elaborado por: Investigador

Tabla 80. Tarea de ingeniería. Revisión y aprobación

TAREA DE INGENIERIA	
Número de tarea: 27	Número de historia: 6
Nombre de Tarea: Revisión y aprobación	
Tipo de tarea: Desarrollo	Puntos estimados: 16
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita un formulario para actualizar el estado de la acción de personal	

Elaborado por: Investigador

Tabla 81. Tarea de ingeniería. Firma y Digitalización

TAREA DE INGENIERIA	
Número de tarea: 28	Número de historia: 6
Nombre de Tarea: Firma y Digitalización	
Tipo de tarea: Desarrollo	Puntos estimados: 16
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita un formulario para actualizar el estado de la acción de personal y que permita subir el documento digitalizado	

Elaborado por: Investigador

Tabla 82. Tarea de ingeniería. Anulación

TAREA DE INGENIERIA	
Número de tarea: 29	Número de historia: 6
Nombre de Tarea: Anulación	
Tipo de tarea: Desarrollo	Puntos estimados: 12
Programador responsable: Equipo de Desarrollo	
Descripción: Permitir realizar la anulación de una acción de personal solo por el usuario autorizado	

Elaborado por: Investigador

Tabla 83. Tarea de ingeniería. Reporte de distributivo

TAREA DE INGENIERIA	
Número de tarea: 30	Número de historia: 7
Nombre de Tarea: Reporte de distributivo	
Tipo de tarea: Desarrollo	Puntos estimados: 12
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita un reporte con la información de distributivos subida al sistema	

Elaborado por: Investigador

Tabla 84. Tarea de ingeniería. Reporte de Contratos - Horas

TAREA DE INGENIERIA	
Número de tarea: 31	Número de historia: 7
Nombre de Tarea: Reporte de Contratos – Horas	
Tipo de tarea: Desarrollo	Puntos estimados: 12
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita un reporte acorde a los formatos suministrados por el usuario	

Elaborado por: Investigador

Tabla 85. Tarea de ingeniería. Reporte de Contratación

TAREA DE INGENIERIA	
Número de tarea: 32	Número de historia: 7
Nombre de Tarea: Reporte de Contratación	
Tipo de tarea: Desarrollo	Puntos estimados: 12
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita un reporte acorde a los formatos suministrados por el usuario	

Elaborado por: Investigador

Tabla 86. Tarea de ingeniería. Reporte Formación Docente Terminado

TAREA DE INGENIERIA	
Número de tarea: 33	Número de historia: 7
Nombre de Tarea: Reporte Formación Docente Terminado	
Tipo de tarea: Desarrollo	Puntos estimados: 12
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita un reporte acorde a los formatos suministrados por el usuario	

Elaborado por: Investigador

Tabla 87. Tarea de ingeniería. Reporte Formación Docente en Curso

TAREA DE INGENIERIA	
Número de tarea: 34	Número de historia: 7
Nombre de Tarea: Reporte Formación Docente en Curso	
Tipo de tarea: Desarrollo	Puntos estimados: 12
Programador responsable: Equipo de Desarrollo	
Descripción: Se necesita un reporte acorde a los formatos suministrados por el usuario	

Elaborado por: Investigador

Tabla 88. Tarea de ingeniería. Reporte de Funcionarios

TAREA DE INGENIERIA	
Número de tarea: 35	Número de historia: 7
Nombre de Tarea: Reporte de Funcionarios	
Tipo de tarea: Desarrollo	Puntos estimados: 12
Programador responsable: Equipo de Desarrollo	

Descripción: Se necesita un reporte acorde a los formatos suministrados por el usuario

Elaborado por: Investigador

Tabla 89. Tarea de ingeniería. Creación de roles y asignación de Active Directory

TAREA DE INGENIERIA	
Número de tarea: 36	Número de historia: 8
Nombre de Tarea: Creación de roles y asignación de permisos Active Directory	
Tipo de tarea: Desarrollo	Puntos estimados: 20
Programador responsable: Equipo de Desarrollo	
Descripción: Integración del sistema al SIUTA por medio del servicio de Active Directory	

Elaborado por: Investigador

6.9.5 Planeación del Sprint

En la planificación se especifica las historias de usuario que forma parte de cada Sprint, planificación resultante de las estimaciones de historias de usuario y tareas de ingeniería.

Sprint 1

El primer Sprint abarca las funcionalidades de “Seguridades y Parametrización” y, “Gestión de las Acciones de Personal”, siendo estos los entregables de este Sprint. Las tablas Tabla 90 y Tabla 91 muestran las historias de usuario que contienen estas funcionalidades.

Tabla 90. Funcionalidad: Seguridades y Parametrización

Historia de usuario	Fecha inicio	Fecha fin
Acceso al Sistema de Acciones de Personal	04/06/2018	05/06/2018
Plantilla	05/06/2018	07/06/2018
Asignación a los perfiles de usuarios	07/06/2018	12/06/2018
Parametrización de datos iniciales para Acciones de personal	12/06/2018	19/06/2018

Elaborado por: Investigador

Tabla 91. Funcionalidad: Gestión de las Acciones de Personal

Historia de usuario	Fecha inicio	Fecha fin
Gestionar las Acciones de Personal	19/06/2018	02/07/2018
Revisión y Aprobación	02/07/2018	06/07/2018

Elaborado por: Investigador

Objetivo del Sprint 1: Brindar al usuario la utilidad de poder realizar una parametrización inicial del sistema, así como lograr sistematizar la gestión del proceso de generación de los diferentes tipos de acciones de personal.

Sprint 2

El segundo Sprint abarca a la funcionalidad de “Reportería e Integración con el SI-UTA”, misma que es el entregable de este Sprint. La Tabla 92 muestra las historias de usuario que contiene esta funcionalidad.

Tabla 92. Funcionalidad: Reportería e Integración con el SI-UTA

Historia de usuario	Fecha inicio	Fecha fin
Reportes	09/07/2018	20/07/2018
Integración del Sistema de acción de personal con el SI-UTA	20/07/2018	24/07/2018

Elaborado por: Investigador

Objetivo del Sprint 2: Satisfacer la necesidad de contar con los reportes necesarios con la información ingresada, además de integrar el sistema de acciones de personal al sistema integrado de la universidad SI-UTA.

6.9.6 Reunión de Equipo de Scrum

Las reuniones diarias de trabajo tuvieron una duración de aproximadamente 15 minutos, donde se analizaba el trabajo realizado el día anterior y los elementos de trabajo pendientes para el presente día. En estas reuniones de trabajo se aclararon y solventaron cualquier duda o inquietud y los Sprint se desarrollaron con normalidad.

6.9.7 Refinamiento del Backlog

El dueño del producto como usuario final del sistema de acciones de personal constantemente revisaba cada uno de los elementos de trabajo pendientes aclarando cualquier inquietud que el equipo de desarrollo tenía.

6.9.8 Revisión del Sprint

Al finalizar cada Sprint se mantuvo una reunión para inspeccionar el incremento con las funcionalidades a ser entregadas en donde intervino todo el equipo Scrum.

6.9.9 Retrospectiva del Sprint

Al finalizar cada Sprint se realizó la reunión de retrospectiva con el fin de obtener una retroalimentación con los puntos positivos y negativos encontrados durante el desarrollo de los mismos. Las tablas: Tabla 93 y Tabla 94 muestran los datos de las reuniones de retrospectivas de cada Sprint realizadas al finalizar los mismos.

Sprint 1

Tabla 93. Retrospectiva del Sprint 1

RETROSPECTIVA DEL SPRINT 1	
Fecha: 09/07/2018	Hora: 08:30
Objetivo: Realizar una retroalimentación para mejorar el comportamiento del Sprint 1	
Actividades: Identificar los inconvenientes surgidos durante el desarrollo del primer Sprint	
Resultado: Se concluyó que el Sprint se ejecutó con normalidad al no tener retrasos en el desarrollo de los elementos de trabajo pendientes. El siguiente se ejecutará con la planeación original del segundo Sprint.	
Participantes: Todo el equipo Scrum	

Elaborado por: Investigador

La gestión de los elementos de trabajo pendiente se la realizó con ayuda del software Team Foundation Server. La Figura 11 muestra el panel de tareas con la vista “Trabajo pendiente” del Sprint 1.



Figura 11. TFS. Panel de tareas: Trabajo pendiente del Sprint 1

Elaborado por: Investigador

Sprint 2

Tabla 94. Retrospectiva del Sprint 2

RETROSPECTIVA DEL SPRINT 2	
Fecha: 25/07/2018	Hora: 08:30
Objetivo: Realizar una retroalimentación final de los Sprint ejecutados	
Actividades: Identificar los inconvenientes, puntos positivos y negativos del desarrollo de los Sprint	
Resultado: De igual forma que el Sprint anterior, este se ejecutó con normalidad. Se concluye que es primordial el comprometimiento de todos los involucrados en el desarrollo del producto: Scrum Master, Equipo Scrum, Dueño del producto; y también de los interesados del mismo: directores departamentales (Stakeholders)	
Participantes: Todo el equipo Scrum	

Elaborado por: Investigador

La Figura 12 muestra el panel de tareas en detalle para el Sprint 2:

Orden	Tipo de elemento de trabajo	Título	Estado de tarea
1	Elemento de trabajo pendiente del producto	Integración del Sistema de acción de personal con el SIUTA	● Acabado
2	Elemento de trabajo pendiente del producto	Reportes	● Acabado
	Tarea	Reporte de distributivo	● Acabado
	Tarea	Reporte de Contratos - Horas	● Acabado
	Tarea	Reporte de Contratación	● Acabado
	Tarea	Reporte Formación Docente Terminado	● Acabado
	Tarea	Reporte Formación Docente en Curso	● Acabado
	Tarea	Reporte de Funcionarios	● Acabado

Figura 12. TFS. Panel de tareas: Detalle del Sprint 2

Elaborado por: Investigador

6.9.10 Buenas prácticas de programación aplicadas al proyecto

6.9.10.1 Documentación técnica

La documentación técnica resultante del proceso de desarrollo de software del sistema de acciones de personal está enmarcada en el formato establecido en la presente propuesta.

6.9.10.2 Documentación de código

El código fuente del sistema de acciones de personal se codificó en consideración de las buenas prácticas de programación propuestas en el presente trabajo de investigación, con el fin de permitir su entendimiento tanto de su estructura como de su contenido. Las figuras: Figura 13, Figura 14 y Figura 15 muestran ejemplos de la documentación aplicada al código fuente y a la estructura de la base de datos.

```
/// <summary>
/// Variable temporal para la asignacion de las iniciales de los nombres del
/// creador(Role-Usuario) de la accion de personal.
/// </summary>
public string inicial;

/// <summary>
/// Variable temporal para la asignacion del path y extension de archivo o
/// documento PDF a subir de la accion de personal firmada y sellada
/// </summary>
public string pdf_pth;
```

Figura 13. Documentación en la definición de variables

Elaborado por: Investigador

```
/// <summary>
/// Funcion para poblar el ddlTipoAccion(DropDownList) los tipo de accion
/// de personal almacenados en la base de datos.
/// </summary>
void LlenarDropDownListTipoAccionPersonal()
{
    sqlcmd = new SqlCommand("proc_PersonnelActionsTypeLoadAll", sqlcon);
    sqlcmd.CommandType = CommandType.StoredProcedure;
    try
    {
```

Figura 14. Documentación en una función

Elaborado por: Investigador

no	Name	Data Type	Not Null	Identity	Default	Description
<input checked="" type="checkbox"/>	serial_doc	int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		PRIMARY KEY
<input type="checkbox"/>	name_doc	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>		Indicates the name of the document
<input type="checkbox"/>	date_doc	datetime	<input type="checkbox"/>	<input type="checkbox"/>		Indicates the date of the document
<input type="checkbox"/>	serial_acc	varchar(30)	<input type="checkbox"/>	<input type="checkbox"/>		Foreign key of the accionpersonal
<input type="checkbox"/>	serial_sta	varchar(1)	<input type="checkbox"/>	<input type="checkbox"/>		Foreign key of the states
<input type="checkbox"/>	pdfname_doc	varchar(80)	<input type="checkbox"/>	<input type="checkbox"/>		Indicates the name of the pdf
<input type="checkbox"/>	pdfurl_doc	varchar(100)	<input type="checkbox"/>	<input type="checkbox"/>		Indicates the url of the pdf


```

1 CREATE TABLE dbAccionesPersonal.dbo.Documents (
2     serial_doc INT IDENTITY
3     ,name_doc VARCHAR(50) NULL
4     ,date_doc DATETIME NULL
5     ,serial_acc VARCHAR(30) NULL
6     ,serial_sta VARCHAR(1) NULL
7     ,pdfname_doc VARCHAR(80) NULL
8     ,pdfurl_doc VARCHAR(100) NULL
9     ,CONSTRAINT PK_Documents_serial_doc PRIMARY KEY CLUSTERED (serial_doc)
10    ,CONSTRAINT KEY_Documents_UNIQUE (serial_acc, serial_sta)
11    ) ON [PRIMARY]
12 GO

```

Figura 15. Documentación en la base de datos

Elaborado por: Investigador

6.9.10.3 Tabulación

La tabulación del código fuente la realizó de forma automática el IDE de desarrollo utilizado. La Figura 16 muestra la tabulación resultante de la codificación con el IDE de desarrollo Microsoft Visual Studio.

```

protected void txtFechaRige_Validation(object sender, ValidationEventArgs e)
{
    DateTime dateStart = txtFechaRige.Date;
    int resultado = dateStart.CompareTo(txtFechaRigeHasta.Date);

    if (resultado == 1)
    {
        e.IsValid = false;
        e.ErrorMessage = "Rige desde debe ser menor o igual a Rige hasta";
        txtFechaRige.ValidationSettings.ErrorMessagePosition = ErrorMessagePosition.Bottom;
        Session["fd"] = "0";
    }
    else
    {
        Session["fd"] = "1";
        e.IsValid = true;
        e.ErrorMessage = "";

        if (txtFechaRigeHasta.IsValid == false)
        {
            txtFechaRigeHasta.IsValid = true;
        }
    }
}

```

Figura 16. Tabulación de código

Elaborado por: Investigador

6.9.10.4 Control de versiones

El control de versiones se encuentra gestionado por la herramienta TFVC de Microsoft. En la Figura 17 se puede observar las diferencias de código entre dos versiones, característica muy útil al comparar cambios o programación realizada.

```
Se muestran 1 cambio en archivos: 1 ediciones | Diferencias en paralelo | ↺ ↻ ↗

C# AdminUsuarios.aspx.cs +15 -5
$/Acciones/AccionesPersonal/AccionesPersonal/AccionesPersonal/AccionesAdmin/AdminUsuarios.aspx.cs

...
560     ///respectivos permisos en la base de datos dbAccionesPersonal y
561     /// </summary>
562     /// <param name="Cedula"></param>
563 - void spEliminarAsignados(string Cedula)
564     {
565     ///////////////////////////////////////////////////////////////////
566         sqlcmd = new SqlCommand("UsersPermissionsDelete", sqlcon);
567         sqlcmd.CommandType = CommandType.StoredProcedure;
568
569     .....
570         sqlcon.Open();
571         sqlcmd.Parameters.Add("@serial_user", SqlDbType.VarChar)
572
573 - sqlcmd.ExecuteNonQuery();
574
575     }
576     catch (Exception ex)

...
560     ///respectivos permisos en la base de datos dbAccionesPersonal y
561     /// </summary>
562     /// <param name="Cedula"></param>
563 + int spEliminarAsignados(string Cedula)
564     {
565 + int rowaffect=0;
566         sqlcmd = new SqlCommand("UsersPermissionsDelete", sqlcon);
567         sqlcmd.CommandType = CommandType.StoredProcedure;
568
569     .....
571         sqlcon.Open();
572         sqlcmd.Parameters.Add("@serial_user", SqlDbType.VarChar)
573
574 + rowaffect = sqlcmd.ExecuteNonQuery();
575
576     }
577     catch (Exception ex)
```

Figura 17. Control de versiones TFVC

Elaborado por: Investigador

6.9.11 Diseño de la Base de Datos

La Figura 18 muestra el diagrama de base de datos que utiliza el sistema de acciones de personal.

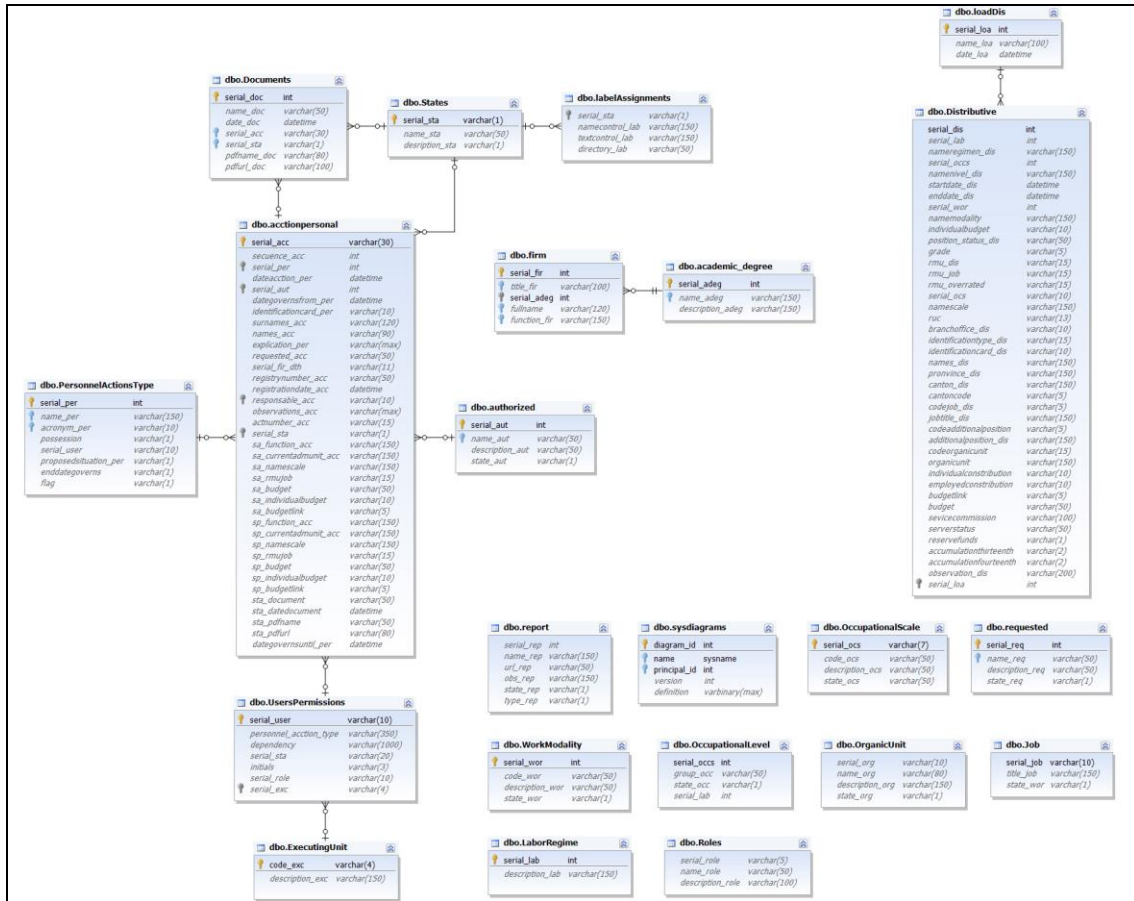


Figura 18. Diagrama de la base de datos

Elaborado por: Investigador

6.9.12 Pruebas

El Área de Gestión de Calidad realizó las respectivas pruebas de funcionalidad al sistema desarrollado, las observaciones encontradas fueron resueltas para su posterior verificación y validación.

6.9.13 Indicadores de calidad

Se utilizaron las métricas de calidad de adecuación funcional definidas en la presente propuesta, métricas que definen el nivel de funcionalidad del producto software resultante del proceso de desarrollo según la norma internacional ISO/IEC 25023.

Tabla 95. Adecuación funcional del sistema de Acciones de Personal

Software evaluado:	Acciones de Personal
Tipo de Metodología:	AGIL
Metodología o Marco de Trabajo utilizado:	SCRUM
Código documentado:	SI
Versionamiento:	SI
Especificación de requerimientos:	SI

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	<i>x₁</i>	<i>e</i>	<i>te</i>	<i>x₂</i>	<i>fu</i>	<i>tfu</i>	<i>x₃</i>	%	\bar{x}
0	28	1,00	46	46	1,00	22	22	1,00	100%	Alta

Elaborado por: Investigador

CAPITULO VII

7 ANÁLISIS DE IMPACTOS

7.1 Introducción

Posterior al desarrollo del Sistema de Acciones de Personal desarrollado por la DITIC-UTA utilizando la presente propuesta, se realiza el análisis de impactos que tiene la utilización de buenas prácticas de programación y Scrum como marco de trabajo ágil en el incremento de la funcionalidad del producto software.

Se utilizan dos instrumentos para la recopilación de información: 1) una encuesta aplicada mediante cuestionario al personal relacionado directa o indirectamente con el software desarrollado que utiliza el marco de trabajo propuesto; 2) el formulario de evaluación y medición aplicado anteriormente a los sistemas ya desarrollados por la DITIC-UTA.

Para el procesamiento estadístico se utiliza el software IBM SPSS Statistics versión 25; las preguntas correspondientes a la encuesta son formuladas mediante la escala de Likert, y para su validación se aplica la prueba Alfa de Cronbach garantizando la fiabilidad de la encuesta.

Se utiliza la prueba no paramétrica de Chi Cuadrado de Pearson para la validación de la hipótesis, así como la aplicación de métricas de calidad según la norma internacional ISO/IEC 25023 para verificar el incremento de funcionalidad.

7.2 Análisis de resultados

7.2.1 Encuesta a usuarios, personal técnico y funcionarios

Encuesta aplicada a usuarios, personal técnico y funcionarios que se encuentran relacionados directa o indirectamente con el software evaluado. La encuesta consta de 8 puntos:

Punto 1. Aseguramiento del cumplimiento de las necesidades del cliente.

Pregunta 1. ¿Se tiene la garantía que el software desarrollado sea completo y funcional?

Tabla 96. Aseguramiento del cumplimiento de las necesidades del cliente

Alternativas	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Totalmente de acuerdo	12	60	60	60
De acuerdo	8	40	40	100
Indeciso				
En desacuerdo				
Totalmente en desacuerdo				
Total	20	100	100	

Elaborado por: Investigador

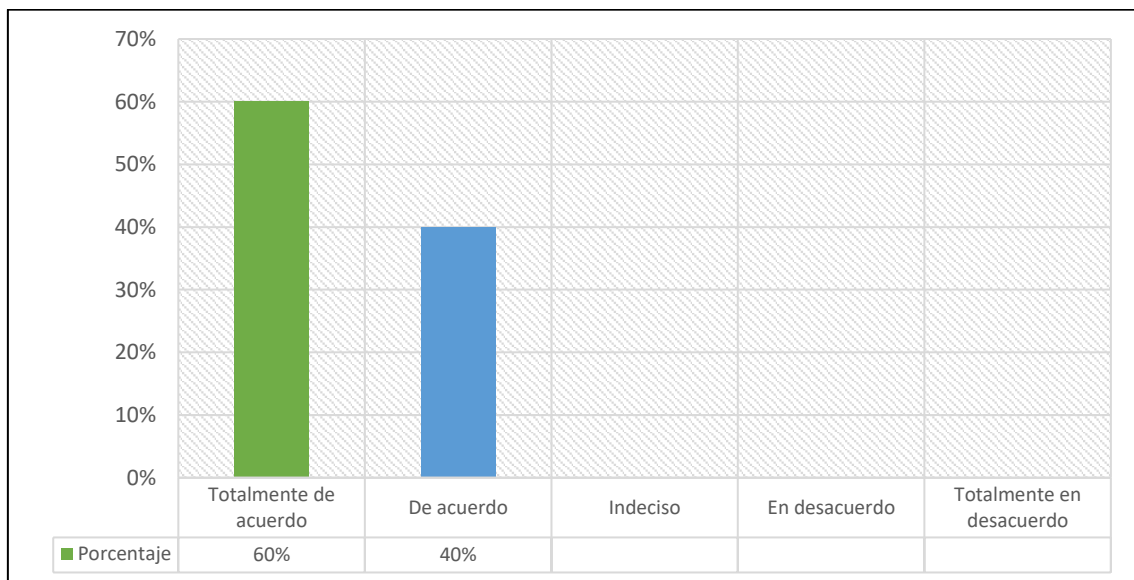


Figura 19. Aseguramiento del cumplimiento de las necesidades del cliente

Elaborado por: Investigador

Como se puede observar, todos los encuestados están de acuerdo (40%) o totalmente de acuerdo (60%) en tener la garantía que el software desarrollado sea completo y funcional, lo que significa que se ha asegurado el cumplimiento de las necesidades del cliente

Punto 2. Reducción de errores en el sistema en producción.

Pregunta 2. ¿El sistema emite resultados correctos con un nivel de precisión necesario?

Tabla 97. Reducción de errores en el sistema en producción

Alternativas	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Totalmente de acuerdo	11	55	55	55
De acuerdo	9	45	45	100
Indeciso				
En desacuerdo				
Totalmente en desacuerdo				
Total	20	100	100	

Elaborado por: Investigador

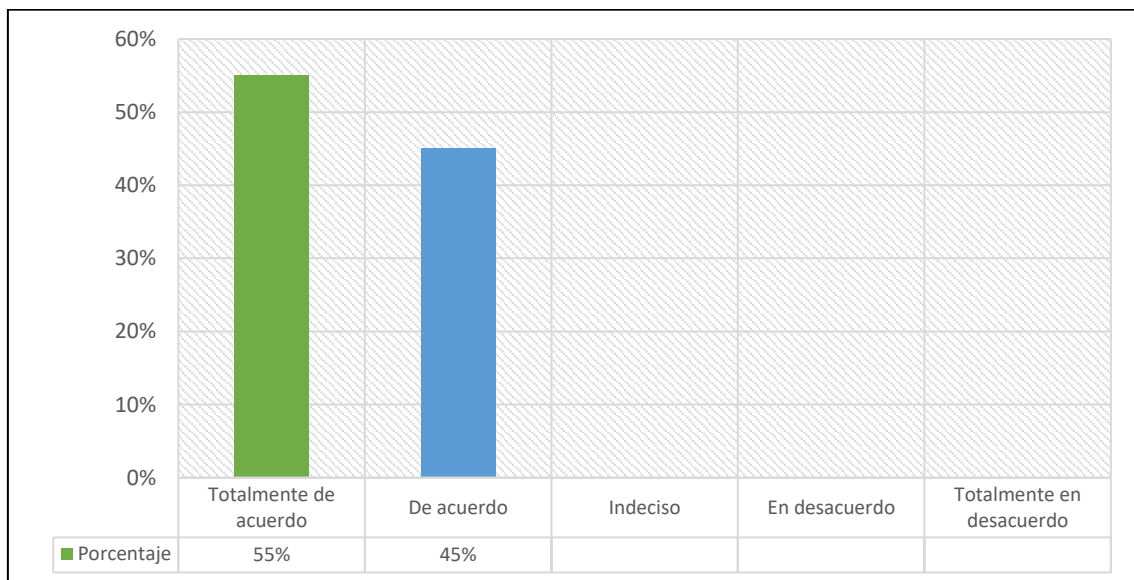


Figura 20. Reducción de errores en el sistema en producción

Elaborado por: Investigador

Aquí, todos los encuestados se encuentran de acuerdo (45%) o totalmente de acuerdo (55%) en que el sistema emite resultados correctos con un nivel de precisión necesario, lo que significa que se ha conseguido la reducción de errores en el sistema en producción.

Punto 3. Mejorar la confianza y satisfacción del cliente.

Pregunta 3. ¿El conjunto de funciones proporcionado por el sistema facilita la realización de las tareas y los objetivos especificados?

Tabla 98. Mejorar la confianza y satisfacción del cliente

Alternativas	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Totalmente de acuerdo	11	55	55	55
De acuerdo	9	45	45	100
Indeciso				
En desacuerdo				
Totalmente en desacuerdo				
Total	20	100	100	

Elaborado por: Investigador

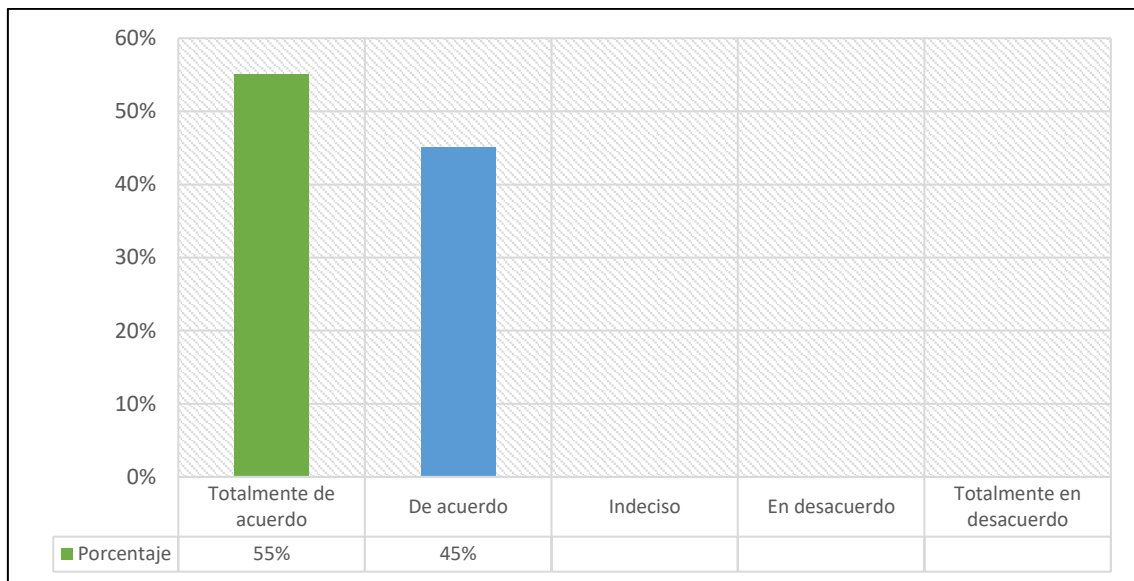


Figura 21. Mejorar la confianza y satisfacción del cliente

Elaborado por: Investigador

En este punto, los encuestados perciben que están de acuerdo (45%) o totalmente de acuerdo (55%) que el conjunto de funciones proporcionado por el sistema facilita la realización de las tareas y los objetivos especificados, logrando mejorar la confianza y satisfacción del cliente.

Punto 4. Aprendizaje y operabilidad.

Pregunta 4. ¿El usuario puede aprender a utilizar el sistema y operarlo con facilidad?

Tabla 99. Aprendizaje y operabilidad

Alternativas	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Totalmente de acuerdo	13	65	65	65
De acuerdo	6	30	30	95
Indeciso	1	5	5	100
En desacuerdo				
Totalmente en desacuerdo				
Total	20	100	100	

Elaborado por: Investigador

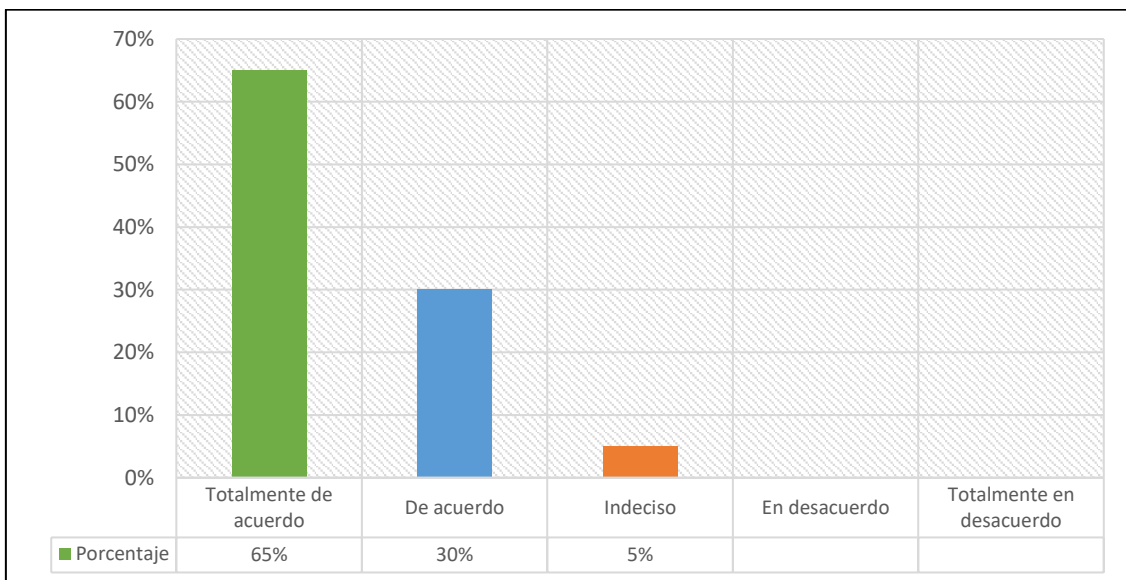


Figura 22. Aprendizaje y operabilidad

Elaborado por: Investigador

Como se puede observar, la mayoría de los encuestados se encuentran de acuerdo (30%) o totalmente de acuerdo (65%) en que el usuario puede aprender a utilizar el sistema y operarlo con facilidad; apenas el 5% se encuentra indeciso, logrando un buen nivel de aprendizaje y operabilidad del sistema.

Punto 5. Protección contra errores del usuario.

Pregunta 5. ¿El sistema tiene la capacidad de evitar que el usuario realice operaciones incorrectas?

Tabla 100. Protección contra errores del usuario

Alternativas	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Totalmente de acuerdo	7	35	35	35
De acuerdo	12	60	60	95
Indeciso	1	5	5	100
En desacuerdo				
Totalmente en desacuerdo				
Total	20	100	100	

Elaborado por: Investigador

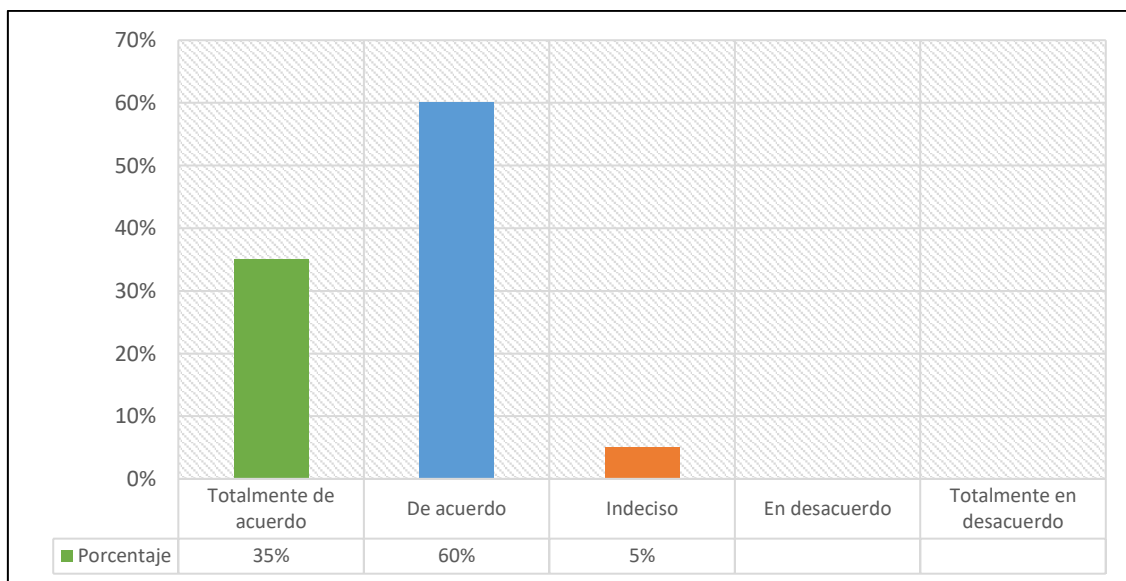


Figura 23. Protección contra errores del usuario

Elaborado por: Investigador

En consideración con las respuestas obtenidas, los encuestados se encuentran de acuerdo (60%) o totalmente de acuerdo (35%) en que el sistema tiene la capacidad de evitar que el usuario realice operaciones incorrectas; apenas el 5% se encuentra indeciso, de esta forma brindar protección contra errores del usuario.

Punto 6. Estética de la interfaz de usuario.

Pregunta 6. ¿La apariencia del sistema agrada y satisface la interacción con la interfaz de usuario?

Tabla 101. Estética de la interfaz de usuario

Alternativas	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Totalmente de acuerdo	11	55	55	55
De acuerdo	8	40	40	95
Indeciso	1	5	5	100
En desacuerdo				
Totalmente en desacuerdo				
Total	20	100	100	

Elaborado por: Investigador

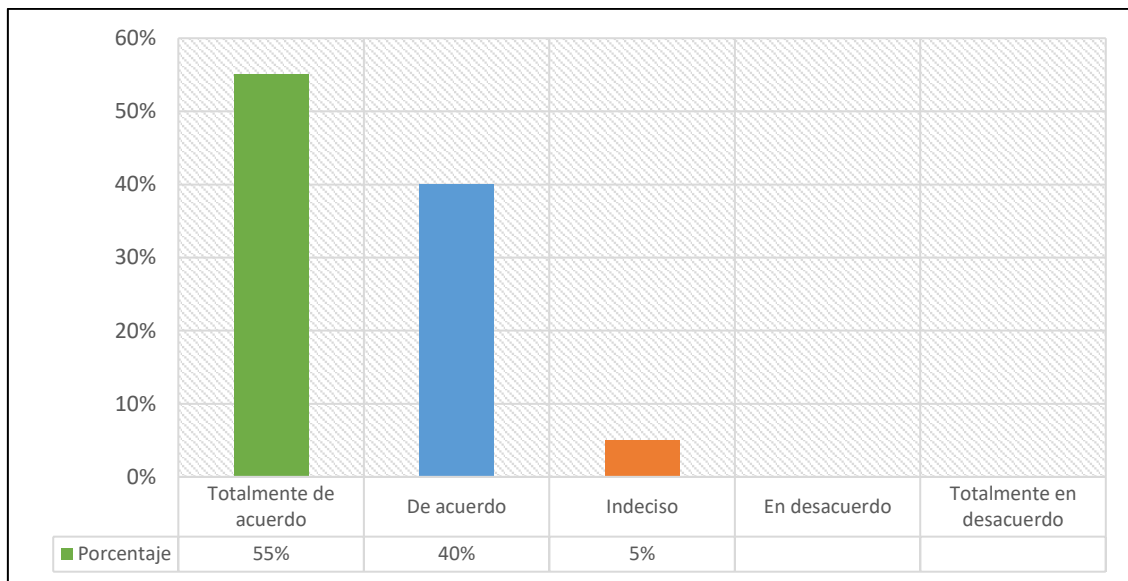


Figura 24. Estética de la interfaz de usuario

Elaborado por: Investigador

La mayoría de los encuestados (95%) se encuentran de acuerdo o totalmente de acuerdo en que la apariencia del sistema agrada y satisface la interacción con la interfaz de usuario; apenas el 5% se encuentra indeciso, lo que significa una buena percepción a la estética de la interfaz de usuario.

Punto 7. Utilidad del sistema.

Pregunta 7. ¿El uso del sistema proporciona un nivel alto de satisfacción de los usuarios?

Tabla 102. Utilidad del sistema

Alternativas	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Totalmente de acuerdo	10	50	50	50
De acuerdo	9	45	45	95
Indeciso	1	5	5	100
En desacuerdo				
Totalmente en desacuerdo				
Total	20	100	100	

Elaborado por: Investigador

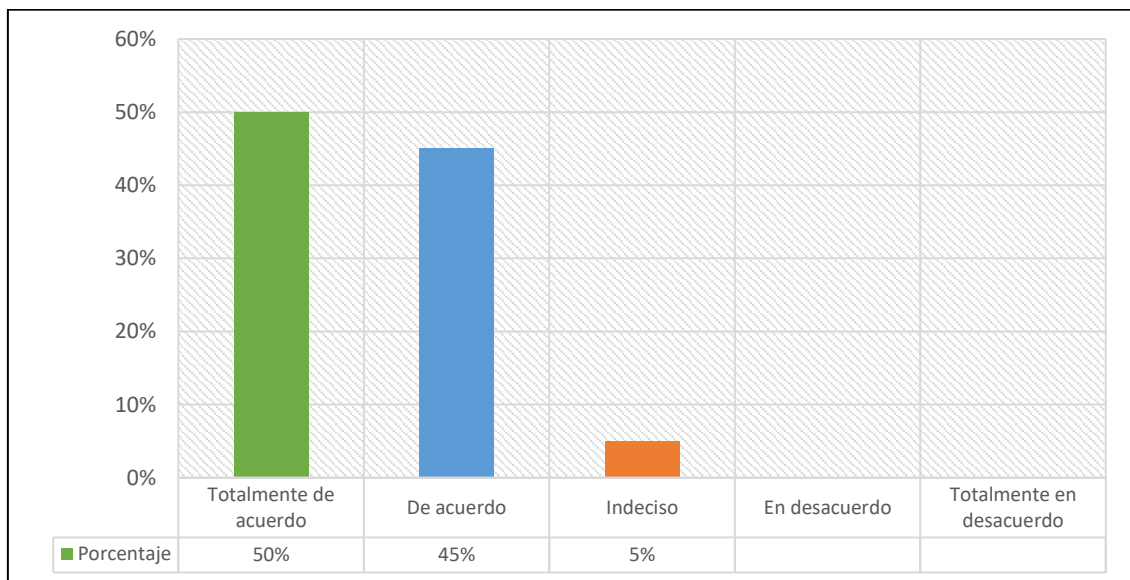


Figura 25. Utilidad del sistema

Elaborado por: Investigador

Como se puede observar, el 95% de los encuestados se encuentra de acuerdo o totalmente de acuerdo que el uso del sistema proporciona un nivel alto de satisfacción de los usuarios; apenas el 5% se encuentra indeciso, logrando que el sistema sea útil para el usuario.

Punto 8. Buenas prácticas de programación y Scrum

Pregunta 8. ¿Considera usted que utilizar buenas prácticas de programación y Scrum como marco de trabajo ágil incrementan el nivel de funcionalidad del producto software?

Tabla 103. Mejores prácticas

Alternativas	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
SI	20	100	100	100
NO				
Total	20	100	100	

Elaborado por: Investigador

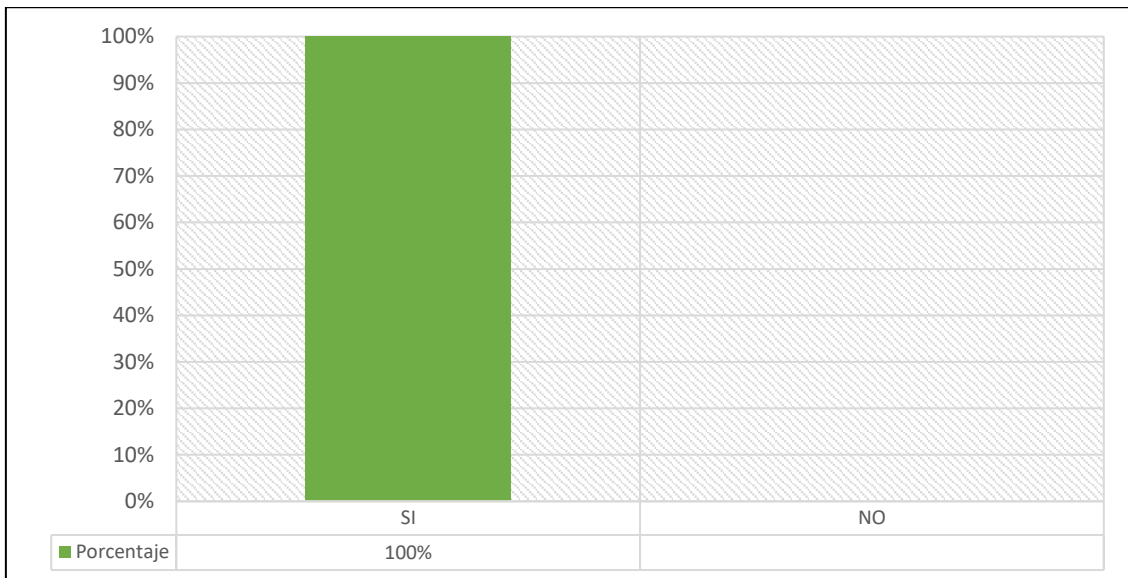


Figura 26. Mejores prácticas

Elaborado por: Investigador

En este punto, todos los encuestados consideran que utilizar buenas prácticas de programación y Scrum como marco de trabajo ágil incrementan el nivel de funcionalidad del producto software.

7.2.2 Formulario de evaluación y medición

Formulario de evaluación y medición al software desarrollado que aplicó la presente propuesta. El formulario consta de los siguientes puntos:

Punto 1. Metodología de desarrollo.

Pregunta 1. ¿Qué tipo metodología de desarrollo se empleó?

El software desarrollado que aplicó la presente propuesta utilizó un tipo de metodología ágil y Scrum como marco de trabajo.

Punto 2. Documentación de código.

Pregunta 2. ¿Se documentó el código fuente?

El software desarrollado que aplicó la presente propuesta si cuenta con un código fuente documentado.

Punto 3. Control de cambios sobre el software.

Pregunta 3. ¿Se utilizó un sistema de versionamiento?

El software desarrollado que aplicó la presente propuesta si utilizó un sistema de versionamiento.

Punto 4. Fuente de medición.

Pregunta 4. ¿Existe documento de especificación de requerimientos y diccionario de datos?

El software desarrollado que aplicó la presente propuesta si cuenta con la documentación técnica solicitada.

Punto 5. Completitud funcional.

Pregunta 5. ¿Qué tan completa es la implementación de acuerdo a las especificaciones de los requerimientos?

En consideración con la norma internacional ISO/IEC 25023 se utilizó la siguiente métrica de calidad para calcular la sub característica de completitud funcional dentro de la característica de calidad de adecuación funcional:

$$x_1 = 1 - ff/tf$$

Donde:

x_1 = Completitud funcional

ff = Número de funciones que faltan o están incorrectamente implementadas

tf = Número de funciones establecidas en la especificación de los requisitos

Valor deseado = El más cercano a 1 es mejor.

La métrica de calidad para el software desarrollado con el tipo de metodología ágil y Scrum marco de trabajo propuesto es de 1, como se visualiza en la Tabla 95.

$$x_1 = 1 - ff/tf = 1 - 0/28 = 1$$

, dicho resultado indica que en consideración a la utilización de buenas prácticas de programación (aplicadas con las preguntas 2 y 3) y Scrum como marco de trabajo ágil (aplicado con la pregunta 1) incrementan la completitud funcional dentro de la característica de calidad de adecuación funcional.

Punto 6. Exactitud funcional.

Pregunta 6. ¿Cuánto del estándar requerido de exactitud se cumple?

En consideración con la norma internacional ISO/IEC 25000 se utilizó la siguiente métrica de calidad para calcular la sub característica de exactitud funcional dentro de la característica de calidad de adecuación funcional:

$$x_2 = e/te$$

Donde:

x_2 = Exactitud funcional

e = Número de elementos de datos implementados con el estándar específico de exactitud

te = Número total de elementos de datos implementados

Valor deseado = El más cercano a 1 es mejor.

La métrica de calidad para el software desarrollado con el tipo de metodología ágil y Scrum marco de trabajo propuesto es de 1, como se visualiza en la Tabla 95

$$x_2 = e/te = 46/46 = 1$$

, dicho resultado indica que en consideración a la utilización de buenas prácticas de programación (aplicadas con las preguntas 2 y 3) y Scrum como marco de trabajo ágil (aplicado con la pregunta 1) incrementan la corrección funcional dentro de la característica de calidad de adecuación funcional.

Punto 7. Pertinencia funcional.

Pregunta 7. ¿Qué parte de las funciones implementadas son percibidas como idóneas?

En consideración con la norma internacional ISO/IEC 25000 se utilizó la siguiente métrica de calidad para calcular la sub característica de pertinencia funcional dentro de la característica de calidad de adecuación funcional:

$$x_3 = fu/TFU$$

Donde:

x_3 = Pertinencia funcional

fu = Número de funciones realmente útiles para realizar tareas específicas

TFU = Número de funciones implementadas para la consecución de tareas específicas

Valor deseado = El más cercano a 1 es mejor.

La métrica de calidad para el software desarrollado con el tipo de metodología ágil y Scrum marco de trabajo propuesto es de 1, como se visualiza en la Tabla 95

$$x_3 = fu/tfu = 22/22 = 1$$

, dicho resultado indica que en consideración a la utilización de buenas prácticas de programación (aplicadas con las preguntas 2 y 3) y Scrum como marco de trabajo ágil (aplicado con la pregunta 1) incrementan la pertinencia funcional dentro de la característica de calidad de adecuación funcional.

7.3 Validación de las respuestas obtenidas

Para validar la fiabilidad de las preguntas y respuestas obtenidas en la encuesta aplicada a usuarios, personal técnico y funcionarios que se encuentran relacionados directa o indirectamente con el software evaluado, se calculó el coeficiente Alfa de Cronbach usando el programa IBM SPSS Statistics.

La encuesta aplicada utiliza la escala de Likert; el grado de fiabilidad al coeficiente Alfa de Cronbach estaría en el siguiente rango: excelente cuando es mayor a 0,9; bueno cuando es mayor a 0,8; aceptable cuando es mayor a 0,7; cuestionable cuando es mayor a 0,6; pobre cuando es mayor a 0,5 e inaceptable cuando es menor a 0,5.

7.3.1 Validación a la encuesta a usuarios, personal técnico y funcionarios

Tabla 104. Resumen de casos de la encuesta a usuarios, personal técnico y funcionarios

		N	%
Casos	Válido	20	100
	Excluido	0	0
	Total	20	100

Elaborado por: Investigador

Tabla 105. Estadística de fiabilidad de la encuesta a usuarios, personal técnico y funcionarios

Alfa de Cronbach	Alfa de Cronbach basada en elementos estandarizados	N de elementos
0,816960537	0,816088386	7

Elaborado por: Investigador

Se observa que el grado de fiabilidad es bueno y existe también consistencia entre los elementos de la encuesta a usuarios, personal técnico y funcionarios que se encuentran relacionados directa o indirectamente con el software evaluado, al obtener un coeficiente Alfa de Cronbach de 0,8169 usando el programa IBM SPSS Statistics.

7.3.2 Interpretación del resultado del formulario de evaluación y medición

La métrica de calidad para el software desarrollado con el tipo de metodología ágil y Scrum marco de trabajo propuesto es de 1, como se visualiza en la Tabla 95.

$$\bar{x} = \frac{\left(1 - \frac{ff}{tf}\right) + \frac{e}{te} + \frac{fu}{tfu}}{n} = \frac{\left(1 - \frac{0}{28}\right) + \frac{46}{46} + \frac{22}{22}}{3} = \frac{1 + 1 + 1}{3} = \frac{3}{3} = 1$$

El resultado indica que en consideración a la utilización de buenas prácticas de programación (aplicadas con las preguntas 2 y 3) y Scrum como marco de trabajo ágil (aplicado con la pregunta 1) incrementan la característica de calidad logrando un alto grado de funcionalidad, equivalente a un 100%, en comparación con la evaluación de todos los sistemas implementados y cuyo mayor valor obtenido fue de 77% correspondiente a una calidad media como se visualiza en la Tabla 28.

7.4 Verificación de la hipótesis

Para la verificación de la hipótesis se utiliza la pregunta 8 de la encuesta aplicada a usuarios, personal técnico y funcionarios que se encuentran relacionados directa o indirectamente con el software desarrollado, debido a que permite evaluar directamente el impacto del tipo de metodología y marco de trabajo propuesto con el uso de buenas prácticas de programación en el incremento del grado de funcionalidad del producto software desarrollados por la DITIC-UTA.

7.5 Verificación estadística de la hipótesis

Para la verificación de la hipótesis se ha utilizado el método Chi Cuadrado de Pearson usando el programa IBM SPSS Statistics.

7.5.1 Planteamiento de la hipótesis

Hipótesis Nula (H0): Aplicando buenas prácticas de programación y Scrum como marco de trabajo ágil NO se incrementará el grado de funcionalidad de un producto software desarrollado por la Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato.

Hipótesis Alternativa (H1): Aplicando buenas prácticas de programación y Scrum como marco de trabajo ágil SI se incrementará el grado de funcionalidad de un producto software desarrollado por la Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato.

Modelo matemático

H0: Observado (O) = Esperado (E)

H1: Observado (O) \neq Esperado (E)

Modelo estadístico

Se utilizó la prueba no paramétrica de Chi Cuadrado de Pearson que nos permite obtener como resultado si existe o no relación entre variables.

$$x^2 = \sum_{i=1}^n \frac{(f_o - f_E)^2}{f_E}$$

Donde:

x^2 = resultado Chi Cuadrado de Pearson

f_o = Frecuencias observadas

f_E = Frecuencias esperadas

Nivel de significancia (α)

Para la prueba se escogió un nivel de significancia de 0,01 que da un nivel de confianza del 99%.

7.5.2 Cálculo de Chi Cuadrado de Pearson

Tomando en cuenta que el 100% de los encuestados contestó Sí a la pregunta 8 de la encuesta aplicada a usuarios, personal técnico y funcionarios que se encuentran

relacionados directa o indirectamente con el software desarrollado, la aprueba del Chi Cuadrado de Pearson en el programa IBM SPSS Statistics no realiza ningún cálculo debido a que la variable es una constante.

Tabla 106. Chi-cuadrado de Pearson para la pregunta 8

	Valor
Chi-cuadrado de Pearson	.a
N de casos válidos	20

a. No se han calculado estadísticos porque Buenas prácticas aplicada al proceso de desarrollo incrementa el nivel de calidad del producto software final es una constante.

Elaborado por: Investigador

De acuerdo al resultado obtenidos, se rechaza la hipótesis nula y se acepta la hipótesis alternativa, por lo tanto, se acepta que aplicando buenas prácticas de programación y Scrum como marco de trabajo ágil SI se incrementará el grado de funcionalidad de un producto software desarrollado por la Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato.

7.6 Datos estadísticos de funcionalidad como indicador de calidad.

Con la tabulación de datos obtenidos por medio del formulario de evaluación y medición de resultados aplicado de manera general a los sistemas desarrollados, se pretende comprobar el incremento del grado de funcionalidad como característica de calidad en el producto software desarrollado que implementa el tipo de metodología y marco de trabajo propuesto.

Se toma como variable dependiente el resultado obtenido aplicando las métricas de calidad de adecuación funcional de cada software evaluado a través de la tabulación de las preguntas 5, 6 y 7, debido a que este valor representa el resultado real de la adecuación funcional como característica de calidad según la norma internacional ISO/IEC 25000.

Como variable independiente se utiliza la pregunta 1, debido a que esta variable indica el tipo de metodología aplicada durante el desarrollo de software.

Se utilizó la prueba no paramétrica de Chi Cuadrado de Pearson que nos permite obtener como resultado si existe o no relación entre variables por medio del programa IBM SPSS Statistics.

$$x^2 = \sum_{i=1}^n \frac{(f_o - f_E)^2}{f_E}$$

Donde:

x^2 = resultado Chi Cuadrado de Pearson

f_o = Frecuencias observadas

f_E = Frecuencias esperadas

Para la prueba se escogió un nivel de significancia de 0,01 que da un nivel de confianza del 99%.

Tabla 107. Resumen de procesamiento de casos

	Casos					
	Válido		Perdidos		Total	
	N	Porcentaje	N	Porcentaje	N	Porcentaje
Tipo de metodología * Funcionalidad	32	100,0%	0	0,0%	32	100,0%

Elaborado por: Investigador

Tabla 108: Tabulación cruzada: Tipo de metodología * Funcionalidad

		Adecuación funcional			Total
		Alta	Baja	Media	
Tipo de metodología	AGIL (Scrum)	1	0	0	1
	TRADICIONAL	0	30	1	31
Total		1	30	1	32

Elaborado por: Investigador

Tabla 109. Prueba del Chi Cuadrado de Pearson para Tipo de metodología * Funcionalidad

	Valor	gl	Significación asintótica (bilateral)
Chi-cuadrado de Pearson	32,000a	2	0,000
Razón de verosimilitud	8,900	2	0,012
N de casos válidos	32		

a. 5 casillas (83,3%) han esperado un recuento menor que 5. El recuento mínimo esperado es ,03.

Elaborado por: Investigador

Se observa que el valor de la significación asintótica bilateral (0,000) es menor al valor del nivel de significancia con el trabajado (0,01), por lo tanto, se rechaza la hipótesis de la independencia de las dos variables, lo que significa que las dos variables son dependientes una de otra o están relacionadas entre sí.

Por lo tanto, se puede concluir que el grado de funcionalidad de un producto software desarrollado por la Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato utilizando el modelo conceptual propuesto, Scrum como marco de trabajo ágil, es alto.

7.6.1 Resultados

Los resultados obtenidos con la aplicación de la propuesta: “Estandarización de los procesos de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil para incrementar el grado de funcionalidad del producto software” son:

- Implementación del modelo de gestión ágil en el Área de Desarrollo de la Dirección de Tecnología de Información y Comunicación de la Universidad Técnica de Ambato logrando incrementar la funcionalidad como característica de calidad en el producto software desarrollado.
- Especificación de las actividades, políticas y procedimientos para el desarrollo de las funciones del personal que intervienen en el Proceso de Gestión de Desarrollo.
- Definición y utilización de buenas prácticas de programación durante la etapa de codificación del producto software.

BIBLIOGRAFÍA

- Álvarez Rodríguez, F. J., Muñoz Arteaga, J., Cardona Salas, J. P., Brizuela Sandoval, M., Quezada Aguilera, F. S., & Ponce Gallegos, J. C. (2008). *Interpretación del Modelo de Madurez de Capacidades (CMM) para pequeñas industrias de software*. México: Universidad Autónoma de Aguascalientes.
- Apache Software Foundation. (2018). *NetBeans*. Obtenido de <https://netbeans.org/>
- Asamblea Nacional. (2008). *Constitución de la República del Ecuador*. Montecristi.
- Asamblea Nacional. (2010). *Ley Orgánica de Educación Superior*. Quito.
- Asamblea Nacional. (2016). *Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación*. Quito.
- Asamblea Nacional. (2018). *Ley Orgánica Reformatoria a la Ley Orgánica de Educación Superior*. Quito.
- Beck, K., Schwaber, K., Grenning, J., Martin, R. C., Beedle, M., Highsmith, J., . . . Marick, B. (Febrero de 2001). *Manifiesto for Agile Software Development*. Obtenido de <http://agilemanifesto.org/>
- Britto Montoya, J. A. (2014). *Adaptación de un proceso de desarrollo de software basado en buenas prácticas*. Manizales: Universidad Autónoma de Manizales.
- Cabezas Gutiérrez, C. F. (2016). *Estudio cualitativo del impacto de aplicación de buenas prácticas para la administración de proyectos de software en 11 organizaciones del sector público*. Quito: Pontificia Universidad Católica del Ecuador.
- Cabot Sagrera, J. (2013). *Ingeniería del software*. Barcelona: Editorial UOC.
- Cadena Loayza, P. N. (2015). *Adaptación de marcos de referencia de calidad a la industria de desarrollo de software en Ecuador*. Quito: Universidad de las Américas.
- Cardador Cabello, A. L. (2014). *Implantación de aplicaciones web en entornos internet, intranet y extranet*. Antequera: IC Editorial.
- Casado Iglesias, C. (2014). *Entornos de desarrollo*. Madrid: RA-MA.
- Chavarriga, J., & Arboleda, H. (2004). Modelo de Investigación en Ingeniería del Software: Una propuesta de investigación tecnológica. *Sociedad de Lógica, Metodología y Filosofía de la Ciencia en España*, 564-567.

- Constanzo, M. (2014). Comparación de Modelos de Calidad, Factores y Métricas en el Ámbito de la Ingeniería de Software. *Universidad Nacional de la Patagonia Austral*.
- Contraloría General del Estado. (2009). *Normas de Control Interno de la Contraloría General del Estado*. Quito.
- Córdoba, J. R. (2014). *Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión*. Málaga: IC Editorial.
- Cortizo Pérez, J. C., Expósito Gil, D., & Ruiz Leyva, M. (2012). *eXtreme Programming*. Buenos Aires.
- Datta, S. (2007). *METRICS-DRIVEN Enterprise Software Development: A Metrics-Driven Approach*. Fort Lauderdale: J. Ross Publishing Inc.
- Delgado, B., & Fernández, M. (2018). Las 9 certificaciones para la transformación digital. (AENOR, Ed.) *Las 9 certificaciones que ayudan a tu empresa en la transformación digital*(340), 13-19.
- Díaz Polo, D. (2011). *Definición de un proceso de desarrollo de software en un entorno universitario*. La Habana: Instituto Superior Politécnico José Antonio Echeverría.
- Don Ho. (2018). *Notepad ++*. Obtenido de <https://notepad-plus-plus.org/>
- Douglass, B. (2013). Agile Development for Embedded Systems. En *Software Engineering for Embedded Systems* (págs. 731-766). Waltham: Elsevier.
- Dromey, R. (1995). A Model for Software Product Quality. *IEEE Transactions on Software Engineering*, 146 - 162.
- Eclipse Foundation. (2018). *Eclipse*. Obtenido de <https://www.eclipse.org/>
- Estayno, M., Dapozo, G., Cuenca, L., & Greiner, C. (2009). Modelos y métricas para evaluar calidad de software. *XI Workshop de Investigadores en Ciencias de la Computación*, 382-388.
- Estévez, Y., & Esteban, L. (2016). Modelo de Calidad para Evaluar el Software Desarrollado en el Centro de Investigación Aplicada y Desarrollo en Tecnologías de Información CIADTI. *Revista TECKNE*, 8-18.
- Fenton, N., & Bieman, J. (2015). *Software Metrics: A Rigorous and Practical Approach*. Boca Raton: CRC Press.

- Fernández, C., & Piattini, M. (2012). *Modelo para el gobierno de las TIC basado en las normas ISO*. Madrid: AENOR.
- Fernandez, C., & Piattini, M. (2013). ISO 25000: Calidad del producto software. (AENOR, Ed.) *Revista de la Normalización y la Certificación*(288), 30-35.
- Gallardo, Y., & Moreno, A. (1999). Recolección de la Información. En ICFES, *Aprender a Investigar*. Santa Fe de Bogotá: ICFES.
- Genero, M., Piattini, M., & Calero, C. (2005). *Metrics for Software Conceptual Models*. London: Imperial College Press.
- GitHub Inc. (2018). *Atom*. Obtenido de <https://atom.io/>
- Gómez, J., Andrade, E., & Macías, J. (2010). Utilizando el Modelo de Calidad de McCall y el Estándar ISO-9126 para la Evaluación de la Calidad de Sistemas de Información por los Usuarios. *AMCIS*.
- González, A., André, M., & Hernández, A. (2015). Análisis comparativo de modelos y estándares para evaluar la calidad del producto de software. *Revista Cubana de Ingeniería*, 43-52.
- Granados La Paz, R. L. (2014). *Despliegue y puesta en funcionamiento de componentes software*. Antequera: IC Editorial.
- Granda Gudiño, P. D. (2016). *Análisis de las metodologías ágiles y su incidencia en la creación del portafolio de servicio para la unidad de extensión universitaria de la Universidad Técnica del Norte de la ciudad de Ibarra*. Ambato: Universidad Técnica de Ambato.
- H. Consejo Universitario. (2012). *Reglamento Orgánico de Gestión Organizacional por Procesos de la Universidad Técnica de Ambato*.
- H. Consejo Universitario. (2013). *Estatuto de la Universidad Técnica de Ambato*.
- H. Consejo Universitario. (2013). *Reglamento de la Dirección de Tecnología de Información y Comunicación -DITIC- de la Universidad Técnica de Ambato*. Ambato.
- H. Consejo Universitario. (2018). *Reglamento del Sistema Integrado de Información SI-UTA*. Ambato.
- Ibarra, F. (2018). *Sistemas de información gerencial y su influencia en la gestión de la inversión*. Ambato: Universidad Técnica de Ambato.

- Institute, T. S. (2010). *CMMI para Desarrollo, Versión 1.3*. Carnegie Mellon University.
- ISO/IEC. (2001). *ISO/IEC 9126-1:2001. Software engineering -- Product quality -- Part 1: Quality model*. Geneve: ISO/IEC.
- ISO/IEC. (2011). *ISO/IEC 25010:2011. Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models*. Geneve: ISO/IEC.
- ISO/IEC. (2016). *ISO/IEC 25023:2016. Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Measurement of system and software product quality*. Geneve: ISO/IEC.
- Lopez, A., Sánchez, A., & Montejano, G. (2016). Definición de Métricas de Calidad para Productos de Software. *XVIII Workshop de Investigadores en Ciencias de la Computación*, 1-6.
- Maalej, W., & Kumar Thurimella, A. (2013). *Managing Requirements Knowledge*. London: Springer.
- Marcos, E. (2018). Investigación en Ingeniería del Software vs. Desarrollo Software.
- Microsoft. (2018). *Microsoft Visual Studio*. Obtenido de <https://visualstudio.microsoft.com/es/>
- Microsoft. (2018). *Visual Studio Code*. Obtenido de <https://code.visualstudio.com/>
- Monte Galiano, J. L. (2016). *Implantar Scrum con éxito*. Barcelona: Editorial UOC.
- Moreno, J., Bolaños, L., & Navia, M. (2010). Exploración de Modelos y Estándares de Calidad Para el Producto Software.
- Musa, K., & Ikhateeb, J. (2013). Quality Model Based on Cots Quality Attributes. *International Journal of Software Engineering & Applications (IJSEA)*, Vol.4, No.1, January 2013, 1-8.
- Opelt, A., Gloger, B., Pfarl, W., & Mittermayr, R. (2013). *Agile Contracts*. Danvers: Wiley.
- Oshana, R., & Kraeling, M. (2013). *Software Engineering for Embedded Systems*. Waltham: Elsevier.
- Palacios, A., & Merchán, V. (2014). *Guía de Fundamentos de Desarrollo de Proyectos de Software y Metodologías Agiles*. Quito: ESPE.

- Pino, F., Piattini, M., & Fernández, C. (2014). *Modelo de madurez de ingeniería del software*. Madrid: AENOR.
- Pressman, R. (2002). *Ingeniería del Software Un Enfoque Práctico*.
- Ramos, C. (2015). Los paradigmas de la investigación científica. *Avances en Psicología*, 9-17.
- Schwaber, K. (2004). *Agile Project Management*. Redmond, Washington: Microsoft Press.
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide*. Obtenido de <https://www.scrum.org/>
- Skinner, J. (2018). *Sublime Text*. Obtenido de <https://www.sublimetext.com/>
- Soffer, A., & Dori, D. (2013). Model-Based Requirements Engineering Framework for Systems Life-Cycle Support. En W. Maalej, A. Kumar Thurimella, & Springer (Ed.), *Managing Requirements Knowledge* (págs. 291-311). London: Springer.
- Solleiro, J. L., & Castañón, R. (2008). *Gestión tecnológica: conceptos y prácticas*. México, D. F.
- The Standish Group. (2015). *Chaos Report*.
- Trujillo Casañola, Y. (2015). *Modelo para valorar las organizaciones desarrolladoras de software al iniciar la mejora de procesos*. La Habana: Editorial Universitaria.
- UTIC. (2013). *Modelo de Calidad de Software para Desarrollo de Sistemas en la DGSC*. Costa Rica: Dirección General de Servicio Civil.
- Vern, R., & Kumar, S. (2013). A Survey on Evaluation of the Quality of Software System by Using Fuzzy Logic Approach. *Global Journal of Computer Science and Technology Software & Data Engineering*.

ANEXOS

Anexo 1



Universidad Técnica de Ambato
Dirección de Tecnología de Información y Comunicación
Estudio de Investigación

Encuesta aplicada a usuarios, personal técnico y funcionarios que se encuentran relacionados directa o indirectamente con el software evaluado.

La presente encuesta forma parte de un estudio de investigación, su objetivo es obtener información relacionada a la calidad del software desarrollado.

Instrucciones: Marque con una X la alternativa que considere apropiada.

Aseguramiento del cumplimiento de las necesidades del cliente.

1. ¿Se tiene la garantía que el software desarrollado sea completo y funcional?

Totalmente de acuerdo:[_] De acuerdo:[_] Indeciso:[_] En desacuerdo:[_] Totalmente en desacuerdo:[_]

Reducción de errores en el sistema en producción.

2. ¿El sistema emite resultados correctos con un nivel de precisión necesario?

Totalmente de acuerdo:[_] De acuerdo:[_] Indeciso:[_] En desacuerdo:[_] Totalmente en desacuerdo:[_]

Mejorar la confianza y satisfacción del cliente.

3. ¿El conjunto de funciones proporcionado por el sistema facilita la realización de las tareas y los objetivos especificados?

Totalmente de acuerdo:[_] De acuerdo:[_] Indeciso:[_] En desacuerdo:[_] Totalmente en desacuerdo:[_]

Aprendizaje y operabilidad.

4. ¿El usuario puede aprender a utilizar el sistema y operarlo con facilidad?

Totalmente de acuerdo:[_] De acuerdo:[_] Indeciso:[_] En desacuerdo:[_] Totalmente en desacuerdo:[_]

Protección contra errores del usuario.

5. ¿El sistema tiene la capacidad de evitar que el usuario realice operaciones incorrectas?

Totalmente de acuerdo:[_] De acuerdo:[_] Indeciso:[_] En desacuerdo:[_] Totalmente en desacuerdo:[_]

Estética de la interfaz de usuario.

6. ¿La apariencia del sistema agrada y satisface la interacción con la interfaz de usuario?

Totalmente de acuerdo:[_] De acuerdo:[_] Indeciso:[_] En desacuerdo:[_] Totalmente en desacuerdo:[_]

Utilidad del sistema.

7. ¿El uso del sistema proporciona un nivel alto de satisfacción de los usuarios?

Totalmente de acuerdo:[_] De acuerdo:[_] Indeciso:[_] En desacuerdo:[_] Totalmente en desacuerdo:[_]

Buenas prácticas de programación y Scrum.

8. ¿Considera usted que utilizar buenas prácticas de programación y Scrum como marco de trabajo ágil incrementan el nivel de funcionalidad del producto software?

Si:[_] No:[_]

Cargo.

Favor indicar el cargo que ocupa:

[_____]

Anexo 2



Universidad Técnica de Ambato
Dirección de Tecnología de Información y Comunicación
Estudio de Investigación

Formulario de evaluación y medición de resultados aplicado de manera general a los sistemas desarrollados.

El presente formulario de evaluación y medición forma parte de un estudio de investigación, su objetivo es obtener información relacionada al uso de buenas prácticas de programación y su incidencia en la calidad del producto software.

Software evaluado: [_____]

Metodología de desarrollo.

1. ¿Qué tipo metodología de desarrollo se empleó? Ágil: Tradicional:
Metodología o marco de trabajo utilizado: [_____]

Documentación de código.

2. ¿Se documentó el código fuente? Si: No:

Control de cambios sobre el software.

3. ¿Se utilizó un sistema de versionamiento? Si: No:

Métricas de calidad para Adecuación Funcional

Fuente de medición:

4. ¿Existe documento de especificación de requerimientos y diccionario de datos? Si: No:

Completitud funcional

5. ¿Qué tan completa es la implementación de acuerdo a las especificaciones de los requerimientos?

Número de funciones que faltan o están incorrectamente implementadas: _____

Número de funciones establecidas en la especificación de los requisitos: _____

Exactitud funcional

6. ¿Cuánto del estándar requerido de exactitud se cumple?

Número de elementos de datos implementados con el estándar específico de exactitud: _____

Número total de elementos de datos implementados: _____

Pertinencia funcional

7. ¿Qué parte de las funciones implementadas son percibidas como idóneas?

Número de funciones realmente útiles para realizar tareas específicas: _____

Número de funciones implementadas para la consecución de tareas específicas: _____

Anexo 3

Métricas de calidad para Adecuación Funcional.

Software S01. Estructura Académica Universitaria.

Tabla 110. Adecuación funcional del sistema de Estructura Académica Universitaria

Software evaluado:	Estructura Académica Universitaria
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S02. Distributivos.

Tabla 111. Adecuación funcional del sistema de Distributivos

Software evaluado:	Distributivos
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S03. UTAm@tico Docentes.

Tabla 112. Adecuación funcional del sistema de UTAm@tico Docentes

Software evaluado:	UTAm@tico Docentes
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S04. Estudiantes Matriculas.

Tabla 113. Adecuación funcional del sistema de Estudiantes Matriculas

Software evaluado:	Estudiantes Matriculas
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S05. Estudiantes Calificaciones.

Tabla 114. Adecuación funcional del sistema de Estudiantes Calificaciones

Software evaluado:		Estudiantes Calificaciones
Tipo de Metodología:		TRADICIONAL
Metodología o Marco de Trabajo utilizado:		EMPÍRICA
Código documentado:		NO
Versionamiento:		NO
Especificación de requerimientos:		NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S06. Evaluación Integral del Desempeño Docente.

Tabla 115. Adecuación funcional del sistema de Adecuación funcional del sistema de

Software evaluado:		Evaluación Integral del Desempeño Docente
Tipo de Metodología:		TRADICIONAL
Metodología o Marco de Trabajo utilizado:		EMPÍRICA
Código documentado:		NO
Versionamiento:		NO
Especificación de requerimientos:		NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S07. Tutorías docentes.

Tabla 116. Adecuación funcional del sistema de Tutorías docentes

Software evaluado:	Tutorías docentes
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	CASCADA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	SI

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
8	14	0,57	12	26	0,46	18	21	0,86	63%	Baja

Elaborado por: Investigador

Software S08. Seguimiento al sílabo.

Tabla 117. Adecuación funcional del sistema de Seguimiento al sílabo

Software evaluado:	Seguimiento al sílabo
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	CASCADA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	SI

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
24	26	0,92	20	36	0,56	69	84	0,82	77%	Media

Elaborado por: Investigador

Software S09. Titulados.

Tabla 118. Adecuación funcional del sistema de Titulados

Software evaluado:	Titulados
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S10. Seguimiento a estudios de posgrado.

Tabla 119. Adecuación funcional del sistema de Seguimiento a estudios de posgrado

Software evaluado:	Seguimiento a estudios de posgrado
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	CASCADA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	SI

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
4	7	0,57	0	16	0,00	9	9	1,00	52%	Baja

Elaborado por: Investigador

Software S11. Posgrados.

Tabla 120. Adecuación funcional del sistema de Posgrados

Software evaluado:	Posgrados
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S12. Becas.

Tabla 121. Adecuación funcional del sistema de Becas

Software evaluado:	Becas
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	CASCADA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	SI

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
3	10	0,30	6	10	0,60	10	14	0,71	54%	Baja

Elaborado por: Investigador

Software S13. Matriculas Estudiantes Cultura Física e Idiomas.

Tabla 122. Adecuación funcional del sistema de Matriculas Estudiantes Cultura Física e Idiomas

Software evaluado:	Matriculas Estudiantes Cultura Física e Idiomas
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S14. Información Académica.

Tabla 123. Adecuación funcional del sistema de Información Académica.

Software evaluado:	Estructura Académica Universitaria
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	CASCADA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	SI

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
13	14	0,93	14	21	0,67	11	24	0,46	68%	Baja

Elaborado por: Investigador

Software S15. Hoja de Vida de Funcionarios.

Tabla 124. Adecuación funcional del sistema de Hoja de Vida de Funcionarios

Software evaluado:		Hoja de Vida de Funcionarios
Tipo de Metodología:		TRADICIONAL
Metodología o Marco de Trabajo utilizado:		EMPÍRICA
Código documentado:		NO
Versionamiento:		NO
Especificación de requerimientos:		NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S16. Plan Estratégico de desarrollo institucional.

Tabla 125. Adecuación funcional del sistema de Plan Estratégico de desarrollo institucional

Software evaluado:		Plan Estratégico de desarrollo institucional
Tipo de Metodología:		TRADICIONAL
Metodología o Marco de Trabajo utilizado:		EMPÍRICA
Código documentado:		NO
Versionamiento:		NO
Especificación de requerimientos:		NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S17. Infraestructura.

Tabla 126. Adecuación funcional del sistema de Infraestructura

Software evaluado:	Infraestructura
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S18. Vacaciones.

Tabla 127. Adecuación funcional del sistema de Vacaciones

Software evaluado:	Vacaciones
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S19. Convenios.

Tabla 128. Adecuación funcional del sistema de Convenios

Software evaluado:	Convenios
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S20. Reportes Información estratégica.

Tabla 129. Adecuación funcional del sistema de Reportes Información estratégica

Software evaluado:	Reportes Información estratégica
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S21. Business Intelligence.

Tabla 130. Adecuación funcional del sistema Business Intelligence

Software evaluado:	Business Intelligence
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S22. Control Docente.

Tabla 131. Adecuación funcional del sistema de Control Docente

Software evaluado:	Control Docente
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S23. Evalúate.

Tabla 132. Adecuación funcional del sistema Evalúate

Software evaluado:	Evalúate
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S24. Roles de pago.

Tabla 133. Adecuación funcional del sistema de Roles de pago

Software evaluado:	Roles de pago
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S25. Facturación electrónica.

Tabla 134. Adecuación funcional del sistema de Facturación electrónica

Software evaluado:	Estructura Académica Universitaria
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S26. Simulador de acreditación.

Tabla 135. Adecuación funcional del sistema de Simulador de acreditación

Software evaluado:	Simulador de acreditación
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S27. Seguimiento a graduados.

Tabla 136. Adecuación funcional del sistema de Seguimiento a graduados

Software evaluado:	Seguimiento a graduados
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S28. Bolsa de empleo.

Tabla 137. Adecuación funcional del sistema de Bolsa de empleo

Software evaluado:	Bolsa de empleo
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	EMPÍRICA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	NO

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Elaborado por: Investigador

Software S29. Practicas preprofesionales.

Tabla 138. Adecuación funcional del sistema de Practicas preprofesionales

Software evaluado:	Practicas preprofesionales
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	CASCADA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	SI

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
14	19	0,74	22	35	0,63	55	96	0,57	65%	Baja

Elaborado por: Investigador

Software S30. Proyectos de vinculación.

Tabla 139. Adecuación funcional del sistema de Proyectos de vinculación

Software evaluado:	Proyectos de vinculación
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	CASCADA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	SI

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
8	9	0,89	11	53	0,21	8	10	0,80	63%	Baja

Elaborado por: Investigador

Software S31. Seguimiento de proyectos de Investigación.

Tabla 140. Adecuación funcional del sistema de Seguimiento de proyectos de Investigación.

Software evaluado:	Seguimiento de proyectos de Investigación
Tipo de Metodología:	TRADICIONAL
Metodología o Marco de Trabajo utilizado:	CASCADA
Código documentado:	NO
Versionamiento:	NO
Especificación de requerimientos:	SI

$x_1 = 1 - ff/tf$ Compleitud funcional			$x_2 = e/te$ Exactitud funcional			$x_3 = fu/tfu$ Pertinencia funcional			$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ Adecuación Funcional	
<i>ff</i>	<i>tf</i>	x_1	<i>e</i>	<i>te</i>	x_2	<i>fu</i>	<i>tfu</i>	x_3	%	\bar{x}
33	35	0,94	24	62	0,39	71	301	0,24	52%	Baja

Elaborado por: Investigador

Anexo 4

Resumen formulario de evaluación y medición.

Tabla 141. Resumen formulario de evaluación y medición

Ámbito	Software evaluado	Tipo de Metodología	Metodología o Marco de Trabajo utilizado	Código documentado	Versionamiento	Especificación de requerimientos	Complejidad			Exactitud			Pertinencia			Adecuación Funcional	
							ff	tf	x1	e	te	x2	fu	tfu	x3	%	x
ACADEMIA	Estructura Académica Universitaria	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Distributivos	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	UTAm@tico Docentes	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Estudiantes Matriculas	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Estudiantes Calificaciones	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Evaluación Integral del Desempeño Docente	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Tutorías docentes	TRADICIONAL	CASCADA	NO	NO	SI	6	14	0,57	12	26	0,46	18	21	0,86	63%	Baja
ACADEMIA	Seguimiento al sílabo	TRADICIONAL	CASCADA	NO	NO	SI	2	26	0,92	20	36	0,56	69	84	0,82	77%	Media
ACADEMIA	Titulados	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Seguimiento a estudios de posgrado	TRADICIONAL	CASCADA	NO	NO	SI	3	7	0,57	0	16	0,00	9	9	1,00	52%	Baja
ACADEMIA	Posgrados	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Becas	TRADICIONAL	CASCADA	NO	NO	SI	7	10	0,30	6	10	0,60	10	14	0,71	54%	Baja
ACADEMIA	Matriculas Estudiantes Cultura Física e Idiomas	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
ACADEMIA	Información Académica	TRADICIONAL	CASCADA	NO	NO	SI	1	14	0,93	14	21	0,67	11	24	0,46	68%	Baja
GESTIÓN	Hoja de Vida de Funcionarios	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Plan Estratégico de desarrollo institucional	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Infraestructura	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Vacaciones	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja

Ámbito	Software evaluado	Tipo de Metodología	Metodología o Marco de Trabajo utilizado	Código documentado	Versionamiento	Especificación de requerimientos	Compleitud			Exactitud			Pertinencia			Adecuación Funcional	
							ff	tf	x1	e	te	x2	fu	tfu	x3	%	x
GESTIÓN	Convenios	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Reportes Información estratégica	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Business Intelligence	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Control Docente	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Evalúate	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Roles de pago	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Facturación electrónica	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
GESTIÓN	Simulador de acreditación	EMPÍRICA	NINGUNA	NO	NO	NO	79	79	0,00	0	0	0,00	0	0	0,00	0%	Baja
VINCULACIÓN	Seguimiento a graduados	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
VINCULACIÓN	Bolsa de empleo	EMPÍRICA	NINGUNA	NO	NO	NO	0	0	0,00	0	0	0,00	0	0	0,00	0%	Baja
VINCULACIÓN	Practicas preprofesionales	TRADICIONAL	CASCADA	NO	NO	SI	5	19	0,74	22	35	0,63	55	96	0,57	65%	Baja
VINCULACIÓN	Proyectos de vinculación	TRADICIONAL	CASCADA	NO	NO	SI	1	9	0,89	11	53	0,21	8	10	0,80	63%	Baja
INVESTIGACIÓN	Seguimiento de proyectos de Investigación	TRADICIONAL	CASCADA	NO	NO	SI	2	35	0,94	24	62	0,39	71	301	0,24	52%	Baja

Elaborado por: Investigador