



**UNIVERSIDAD
NACIONAL
DE LOJA**



***Facultad de la Energía, las Industrias y los Recursos Naturales
No Renovables***

ESCUELA DE INGENIERÍA EN SISTEMAS

**“Propuesta de una Herramienta para la Gestión del
Ciclo de Vida de Aplicaciones (ALM) en los procesos
de la Ingeniería del Software.”**

**Tesis previa a la obtención del
título de ingeniero en sistemas**

Autor:

· Silvana-Yasmín, Reyes-Jaramillo

Director:

· Ing. Alex-Vinicio, Padilla-Encalada, Mg. Sc.

LOJA-ECUADOR

2017

CERTIFICACIÓN DEL DIRECTOR

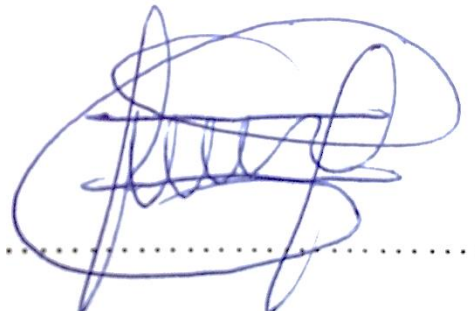
Ing. Alex Vinicio Padilla Encalada, Mg. Sc.

DOCENTE DE LA ESCUELA DE INGENIERÍA EN SISTEMAS

CERTIFICA

Haber dirigido, revisado y corregido en todas sus partes el desarrollo del Trabajo de Titulación de Ingeniería en Sistemas titulado: **“Propuesta de una Herramienta para la Gestión del Ciclo de Vida de Aplicaciones (ALM) en los procesos de la Ingeniería del Software.”**, con autoría de la egresada Silvana Yasmín Reyes Jaramillo. En razón de que la misma reúne a satisfacción los requisitos de fondo y forma, exigidos para la investigación de éste nivel, autorizo su presentación, sustentación y defensa ante tribunal designado para el efecto.

Loja, 19 de mayo de 2017

A handwritten signature in blue ink, consisting of several overlapping loops and lines, positioned above a horizontal dotted line.

Ing. Alex-Vinicio, Padilla-Encalada, Mg. Sc.

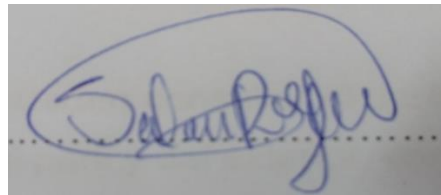
DIRECTOR DEL TRABAJO DE TITULACIÓN.

AUTORÍA

Yo, **SILVANA YASMÍN REYES JARAMILLO**, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional – Biblioteca Virtual.

Firma:



Cédula: 1104679871

Fecha: 23 de junio de 2017

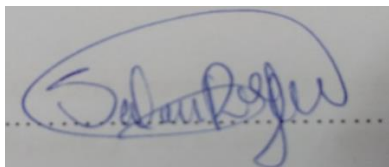
CARTA DE AUTORIZACIÓN DEL TRABAJO DE TITULACIÓN POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo, **SILVANA YASMÍN REYES JARAMILLO**, declaro ser autor de la tesis titulada: **PROPUESTA DE UNA HERRAMIENTA PARA LA GESTIÓN DEL CICLO DE VIDA DE APLICACIONES (ALM) EN LOS PROCESOS DE LA INGENIERÍA DEL SOFTWARE**, como requisito para optar el grado de **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para la constancia de esta autorización, en la ciudad de Loja, a los veinte días del mes de junio del dos mil diecisiete.



Firma:

Autor: Silvana Yasmín Reyes Jaramillo

Cédula: 1104679871

Dirección: Cdla. "El Rosal", Rubén Darío y Rosario Castellanos.

Correo Electrónico: syreyesj@unl.edu.ec

Teléfono: 2710160 **Celular:** 0986681505

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Alex Vinicio Padilla Encalada, Mg. Sc.

Tribunal de Grado: Ing. Mario Andrés Palma Jaramillo Mg. Sc.

Ing. Boris Marcel Díaz Pauta Mg. Sc.

Ing. Marlon Santiago Viñan Ludeña Mg. Sc.

AGRADECIMIENTO

Agradezco a la Virgen del Cisne que me dio fuerza y fe para creer en lo que me parecía imposible terminar. A la Universidad Nacional de Loja y a la Escuela de Ingeniería en Sistemas, que me acogió en su seno científico para dar forma a mi futuro como profesional y a los diferentes docentes que me brindaron sus conocimientos.

A mi tutor Ing. Alex Padilla, por haberme brindado la oportunidad de recurrir a sus conocimientos, así como también por su tiempo, paciencia y apoyo constante en el desarrollo del presente proyecto.

Y finalmente a mis amigos, ya que gracias a su apoyo moral han aportado a mis ganas de seguir en mi carrera profesional.

DEDICATORIA

Mi trabajo de titulación lo dedico con mucho amor a mi madre: Esthelita Jaramillo, quien me dio educación, apoyo, consejos y me formó con buenos sentimientos, hábitos y valores. A Ella por estar conmigo, luchar por mí y corregir mis fallas; sin su motivación, palabras de aliento y amor incondicional esta meta no hubiera podido realizarse. ¡Este y todos mis logros son suyos!

A mi hija Tiffany, por ser mi inspiración y la fuerza para superarme cada día, su inmenso amor me motivó a cumplir con uno más de mis objetivos.

A mi hermano Miguel que me ha brindado su compañía, apoyo y ayuda.

A mi familia en general, por brindarme su amor, ser mi soporte y compartir conmigo los buenos y malos momentos.

a. TÍTULO

Propuesta de una Herramienta para la Gestión del Ciclo de Vida de Aplicaciones (ALM) en los procesos de la Ingeniería del Software.

b. RESUMEN

El presente Trabajo de Titulación se realizó con la finalidad de dar solución al problema de desconocimiento de las herramientas ALM (véase Anexos 2, 3, 4 y 5). En el cual se efectúa un análisis de estas, enfocándose principalmente en aquellas que son libres, donde los datos recopilados provienen de las páginas oficiales de cada una de ellas. Este trabajo está enfocado en realizar un aporte en el área de desarrollo de software, en especial a la gestión del ciclo de vida de las aplicaciones.

La metodología que se utilizó para el análisis, fue el método Descriptivo mismo que permitió evaluar las características de las herramientas open source ALM. Y el método Comparativo el cual ayudó a precisar la mejor herramienta entre las alternativas estudiadas. En cuanto a las técnicas empleadas, se consideró la técnica de Entrevista de la cual se obtuvo la lista de requerimientos de las herramientas; la técnica de Observación para determinar las características de cada herramienta y la técnica Demostrativa para mostrar el correcto uso de las mismas.

Se trabajó con un conjunto de herramientas libres, de las cuales fueron seleccionadas aquellas que cumplieron con la mayoría de requisitos básicos considerados por desarrolladores top de aplicaciones en las empresas de la ciudad de Loja. Para luego ser sometidas a una evaluación ante un escenario de prueba que permitió determinar cuál de ellas es la más apropiada para su utilización.

Adicionalmente se hizo el levantamiento de información relacionada con la metodología de desarrollo más usada en la Escuela de Ingeniería en Sistemas con la finalidad de parametrizarla en la herramienta ALM resultante del análisis, para lo cual se empleó Pentaho que permitió analizar los datos y realizar la predicción de la metodología, cuya deducción fue UWE.

La herramienta resultante de la evaluación fue Hansoft ALM, misma que se aplicó en un caso de estudio en tiempo real, obteniendo como resultado principal organización y apoyo tecnológico en el desarrollo de software.

Summary

The present degree work was done in order to solve the problem of lack of knowledge of the ALM tools (see Annexes 3, 4, 5 and 6). In which an analysis of these is made, focusing mainly on those that are free, where the data collected come from the official pages of each one of them. This work is focused on developing a development in the area of software development, especially; the duration of the software tools.

The methodology that was used for the analysis was the same Descriptive method that allowed to evaluate the characteristics of the open source ALM tools. And the Comparative method which helped to specify the best tool among the alternatives studied. As for the techniques used, we considered the Interview technique from which the list of tool requirements was obtained; The Observation technique to determine the characteristics of each tool and the Demonstrative technique to show the correct use of them.

We worked with a set of free tools, from which we selected those that fulfilled most of the basic requirements for the developers of the applications in the companies of the city of Loja. In order to be submitted to an evaluation before a test scenario that allowed determining which one is the most appropriate for its use.

Additionally, it was found the collection of information related to the development methodology most used in the School of Engineering in Systems with the purpose of parameterizing it in the ALM tool resulting from the analysis, for which Pentaho was used to analyze the data and make the prediction of the methodology, whose deduction was UWE. The tool resulting from the evaluation was Hansoft ALM, which is applied in a real-time case study, obtaining as the main result of the organization and technological support in software development.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN DEL DIRECTOR	II
AUTORÍA	III
CARTA DE AUTORIZACIÓN	IV
AGRADECIMIENTO	V
DEDICATORIA	VI
a. TÍTULO	VII
b. RESUMEN	VIII
Summary	IX
ÍNDICE DE CONTENIDOS	X
ÍNDICE DE FIGURAS	XVII
ÍNDICE DE TABLAS	XXI
c. INTRODUCCIÓN	1
d. REVISIÓN DE LITERATURA	3
1. CAPÍTULO I: La Ingeniería de Software en ALM	3
1.1 Ingeniería de Software	3
1.2 Procesos de Ingeniería de Software	3
1.3 Ingeniería de Software y el Ciclo de Vida del Software	3
2. CAPÍTULO II: Metodologías de Desarrollo	4
2.1 XP (eXtreme Programming)	5
2.1.1 Introducción	5
2.1.2 Características	6
2.1.3 Fases	7

2.1.3.1	<i>Fase I: Exploración</i>	7
2.1.3.2	<i>Fase II: Planificación de la Entrega</i>	7
2.1.3.3	<i>Fase III: Iteraciones</i>	7
2.1.3.4	<i>Fase IV: Producción</i>	7
2.1.3.5	<i>Fase V: Mantenimiento</i>	8
2.1.3.6	<i>Fase VI: Muerte del Proyecto</i>	8
2.1.4	<i>Ventajas</i>	8
2.1.5	<i>Desventajas</i>	8
2.2	<i>Iconix</i>	8
2.2.1	<i>Introducción</i>	8
2.2.2	<i>Características</i>	9
2.2.3	<i>Fases</i>	9
2.2.3.1	<i>Fase I.- Análisis de requisitos</i>	9
2.2.3.2	<i>Fase II.- Análisis y diseño preliminar</i>	9
2.2.3.3	<i>Fase III.- Diseño</i>	9
2.2.3.4	<i>Fase IV.- Implementación</i>	9
2.3	<i>Cristal Methods</i>	10
2.3.1	<i>Introducción</i>	10
2.3.2	<i>Características</i>	11
2.3.3	<i>Valores</i>	11
2.4	<i>FDD (Feature Driven Development – Desarrollo Basado en Características)</i>	11
2.4.1	<i>Introducción</i>	11
2.5	<i>ASD (Adaptative Software Development - Desarrollo Adaptable de Software)</i>	12
2.5.1	<i>Introducción</i>	12

2.5.2	Características	12
2.5.3	Fases.....	12
2.5.3.1	Especulación.....	12
2.5.3.2	Colaboración	13
2.5.3.3	Aprendizaje.....	13
2.6	DSDM (Dynamic Systems Development Method - Método de Desarrollo de Sistema Dinámico)	13
2.6.1	Introducción	13
2.6.2	Características	13
2.6.3	Fases.....	14
2.7	Scrum	14
2.7.1	Introducción	14
2.7.2	Características	15
2.7.3	Fases.....	16
3.	CAPÍTULO III: Gestión del Ciclo de Vida del Software.....	16
3.1	Herramientas ALM Privadas.....	18
3.1.1	Atlassian JIRA.....	18
3.1.2	IBM Rational solution for Collaborative Lifecycle Management	19
3.1.3	IBM Rational Team Concert	20
3.1.4	HP Application Lifecycle Management	20
3.1.5	StarTeam.....	21
3.1.6	SAP Solution Manager.....	22
3.1.7	TraceCloud ALM.....	22
3.1.8	Tomos	23
3.2	Herramientas ALM Libres.....	25

3.2.1	<i>Tuleap</i>	25
3.2.2	<i>Mylyn</i>	27
3.2.3	<i>Endeavour Agile</i>	28
3.2.4	<i>Topcased o Polaris</i>	30
3.2.5	<i>Hansoft</i>	31
3.2.6	<i>Team Foundation Service</i>	33
3.2.7	<i>Star Team Express</i>	35
3.3	<i>Selección de la Herramienta</i>	36
3.3.1	<i>Principales Características y Criterios de Evaluación</i>	36
3.3.1.1	<i>Tuleap</i>	36
3.3.1.2	<i>Mylyn</i>	37
3.3.1.3	<i>Endeavour Agile</i>	38
3.3.1.4	<i>Hansoft</i>	39
3.3.1.5	<i>Team Foundation Service</i>	40
3.3.1.6	<i>Star Team Express</i>	40
3.3.1.7	<i>Topcased o Polaris</i>	41
3.4	<i>Criterios a Tomar en Cuenta en el Proceso de Selección.</i>	41
4.	<i>CAPÍTULO IV: Análisis de las Herramientas a Utilizar</i>	43
4.1	<i>Pentaho.</i>	43
4.2	<i>Metodologías Tradicionales y Web.</i>	45
e.	<i>METODOLOGÍA</i>	47
	<i>MÉTODOS</i>	47
	<i>TÉCNICAS</i>	47
f.	<i>RESULTADOS</i>	49
1.	<i>Criterios de evaluación.</i>	49

2. Características del escenario para la Selección de la Herramienta.....	53
2.1. Casos de Estudio	54
2.2. Selección de Metodología de Prueba	56
3. Análisis de las Herramientas.....	57
3.1. Calidad	57
3.2. Funcionalidad	57
3.3. Usabilidad	58
3.4. Seguridad.....	58
3.5. Parametrización.....	58
3.6. Soporte.....	58
3.7. Licencias	60
3.8. Versionamiento	60
3.9. Uso de metodologías ágiles	61
3.10. Instalación	61
3.11. Valoraciones.....	61
3.12. Sustento Académico Software Libre.....	68
4. Metodología más Usada.	71
4.1. Levantamiento de información	71
4.2. Creación de la base de datos	71
4.3. Herramienta Pentaho	71
4.4. Metodología	74
4.5. Parametrización.....	75
4.6. Propuesta.....	79
4.7. Guía.....	80
1. INSTALACIÓN.....	80

2. ADMIN / ADMINISTRADOR	81
2.1 CREAR UN NUEVO PROYECTO	81
2.2 AGREGAR MIEMBROS AL PROYECTO	83
3. PROJECT / EL PROYECTO.....	86
3.1 PLANNING / PLANIFICACIÓN	86
3.1.1 ETAPAS / FASES	86
3.1.2 ETAPAS BACKLOG ITEMS / TAREAS	88
3.1.3 RELEASES	90
3.1.4 REPORTES	90
3.1.5 VISUALIZACIÓN.....	92
3.1.6 PIPELINES, KANBANS Y WORKFLOWS	97
3.1.7 INFORMES	99
3.2 QUALITY ASSURANCE / ASEGURAMIENTO DE LA CALIDAD	100
3.3 PRODUCT BACKLOG	101
4. DASHBOARDS / TABLERO DE INSTRUMENTOS.....	102
4.1 CREAR UN NUEVO GRÁFICO.....	102
5. TO DO LIST / LISTA DE QUEHACERES.....	103
6. PORTFOLIO FIND / BÚSQUEDA DE CARTERA	104
7. PORTFOLIO ALLOCATIONS / ASIGNACIONES DE CARTERA.....	104
8. DOCUMENTS / DOCUMENTOS	105
9. CHAT.....	106
4.8. Difusión.....	108
g. DISCUSIÓN	112
h. CONCLUSIONES	115
i. RECOMENDACIONES.....	116
j. BIBLIOGRAFÍA	117
k. ANEXOS.....	121
Anexo 1: Entrevista a empresas de Desarrollo de Software de la ciudad de Loja.....	121

Anexo 2: Entrevista 1	122
Anexo 3: Entrevista 2	127
Anexo 4: Entrevista 3	131
Anexo 5: Entrevista 4	135
Anexo 6: Análisis de Entrevistas.....	139
Anexo 7: Evaluación de Herramientas.....	148
Endeavour	148
Hansoft	164
Tuleap.....	177

ÍNDICE DE FIGURAS

Figura 1: Clasificación por colores de la familia Crystal Methods.....	10
Figura 2: Características de Tuleap.....	26
Figura 3: Arquitectura Pentaho Analysis.....	44
Figura 4: Resultados estadísticos del cumplimiento de requerimientos de Herramientas Privadas.....	51
Figura 5: Resultados estadísticos del cumplimiento de requerimientos de Herramientas Libres.....	53
Figura 6: Metodología de prueba resultante.....	56
Figura 7: Resultados estadísticos de las herramientas con mayor valoración y cumplimiento de requerimientos.....	63
Figura 8: Resultados estadísticos de la herramienta con mayor promedio.....	68
Figura 9: Características de una Herramienta ALM.....	70
Figura 10: Interfaz de Pentaho.....	72
Figura 11: Análisis estadístico resultante.....	73
Figura 12: Gráfica del análisis resultante.....	73
Figura 13: Inicio de la parametrización de una nueva metodología.....	76
Figura 14: Parametrización de los datos de la metodología.....	77
Figura 15: Interfaz de la metodología parametrizada en Hansoft ALM.....	78
Figura 16: Modelado de UWE en la fase de Diseño del sistema.....	78
Figura 17: Opciones a disponibles para descargar la herramienta.....	80
Figura 18: Creación de un nuevo proyecto.....	81
Figura 19: Listado de datos a parametrizar según la metodología del proyecto.....	82
Figura 20: Creación de un nuevo usuario.....	83
Figura 21: Ingreso de la información del usuario.....	84
Figura 22: Asignación de los miembros del proyecto.....	84
Figura 23: Asignación del tipo de visibilidad al miembro del proyecto.....	85
Figura 24: Selección del proyecto creado.....	86
Figura 25: Ingreso de etapas o fases del proyecto.....	87
Figura 26: Miembro a cargo del proyecto.....	87
Figura 27: Status de las tareas.....	88
Figura 28: Características adicionales de las tareas.....	89
Figura 29: Creación de un hyperlink.....	89
Figura 30: Creación de releases.....	90
Figura 31: Creación de carpetas.....	91
Figura 32: Creación de reportes.....	91
Figura 33: Visualización y guardado del reporte.....	92
Figura 34: Visualización del status de cada tarea.....	93
Figura 35: Visualización por etapas.....	93
Figura 36: Creación de historias del proyecto.....	94
Figura 37: Visualización de una historia del proyecto.....	94
Figura 38: Visualización del proyecto en base al status.....	95
Figura 39: Creación de tareas programadas.....	96

Figura 40: Visualización de la duración del proyecto.....	96
Figura 41: Creación de workflows.....	97
Figura 42: Selección del tipo de gráfica a realizar.....	98
Figura 43: Visualización del workflow terminado.....	99
Figura 44: Visualización del informe.....	99
Figura 45: Diferentes extensiones para realizar un informe.....	100
Figura 46: Bugs del proyecto.....	100
Figura 47: Ingreso de requisitos funcionales.....	101
Figura 48: Creación de un gráfico.....	102
Figura 49: Visualización de los gráficos generados.....	103
Figura 50: Visualización de bugs pendientes.....	103
Figura 51: Visualización de todos los proyectos creados.....	104
Figura 52: Visualización de los trabajos realizados por los miembros de los proyectos.	105
Figura 53: Creación de una carpeta.....	105
Figura 54: Ingreso de documentos.....	106
Figura 55: Visualización de miembros del equipo.....	107
Figura 56: Visualización de chats con los diferente miembros del equipo.....	107
Figura 57: Página de difusión de Hansoft ALM.....	108
Figura 58: Introducción referente a herramientas ALM.....	109
Figura 59: Especificaciones de Hansoft ALM.....	109
Figura 60: Funciones de Hansoft ALM.....	110
Figura 61: Video explicativo de Hansoft ALM.....	110
Figura 62: Guía de Hansoft ALM.....	111
Figura 63: Creación de un nuevo proyecto.....	148
Figura 64: Asignación de miembros al proyecto.....	149
Figura 65: Creación de iteraciones del proyecto.....	149
Figura 66: Asignación de casos de uso y defectos a las iteraciones del proyecto.....	150
Figura 67: Creación de actores del proyecto.....	150
Figura 68: Creación de casos de uso del proyecto.....	151
Figura 69: Visualización de todos los casos de uso del proyecto.....	151
Figura 70: Cambio de requerimientos del proyecto.....	152
Figura 71: Creación de tareas del proyecto.....	152
Figura 72: Asignación de documentos y stakeholders a las tareas del proyecto.....	153
Figura 73: Asignación de dependencias a las tareas del proyecto.....	153
Figura 74: Visualización de todas las tareas del proyecto.....	154
Figura 75: Ingreso de las palabras del glosario del proyecto.....	154
Figura 76: Creación de bugs del proyecto.....	155
Figura 77: Visualización de los bugs del proyecto.....	155
Figura 78: Creación de Pruebas del proyecto.....	156
Figura 79: Documentación del proyecto.....	156
Figura 80: Creación de Wikis del proyecto.....	157
Figura 81: Visualización total del proyecto.....	157

Figura 82: Visualización del cronograma del proyecto.	158
Figura 83: Vista previa de impresión del proyecto.	158
Figura 84: Reporte de asignación de miembros al proyecto.	159
Figura 85: Reporte de defectos del proyecto.	159
Figura 86: Reporte de iteración de flujo acumulado del proyecto.	160
Figura 87: Reporte de defectos de iteración por prioridad del proyecto.	160
Figura 88: Reporte de defectos de iteración por status del proyecto.	161
Figura 89: Reporte del plan de ejecución del proyecto.	161
Figura 90: Creación de usuarios del proyecto.	162
Figura 91: Soporte de Endeavour en su página oficial.	162
Figura 92: Soporte a través del envío de email.	163
Figura 93: Versionamiento a través Hudson.	163
Figura 94: Creación de un nuevo proyecto.	164
Figura 95: Creación de Usuarios.	164
Figura 96: Creación fases o iteraciones del proyecto.	165
Figura 97: Creación de releases del proyecto.	165
Figura 98: Backlog del proyecto.	166
Figura 99: Creación de subproyectos o tareas del proyecto.	166
Figura 100: Visualización de backlog y tareas de la Fase 1 del proyecto.	167
Figura 101: Visualización de las fases y su estado en el proyecto.	167
Figura 102: Visualización de los tiempos de las fases del proyecto.	168
Figura 103: Creación de la página de dashboards del proyecto.	168
Figura 104: Creación de diagramas del proyecto.	169
Figura 105: Visualización de diagramas del proyecto.	169
Figura 106: Creación de tareas programadas del proyecto.	170
Figura 107: Creación de diagramas del proyecto.	170
Figura 108: Creación de reportes del proyecto.	171
Figura 109: Visualización del reporte de la Fase 1 del proyecto.	171
Figura 110: Creación de Workflow del proyecto.	172
Figura 111: Creación y visualización bugs del proyecto.	172
Figura 112: Visualización de las asignaciones de la cartera del proyecto.	173
Figura 113: Visualización de todos los proyectos.	173
Figura 114: Visualización del tablero de actividades del proyecto.	174
Figura 115: Creación de documentos del proyecto.	174
Figura 116: Visualización de documentos del proyecto.	175
Figura 117: Comunicación / Chat.	175
Figura 118: Soporte de Hansoft en su página oficial.	176
Figura 119: Soporte a través del envío de email.	176
Figura 120: Versionamiento a través Git.	177
Figura 121: Creación de un nuevo proyecto.	178
Figura 122: Correo de aprobación del nuevo proyecto.	178
Figura 123: Selección de la plantilla para el proyecto.	179
Figura 124: Creación de sprints del proyecto.	179

Figura 125: Adición de tareas al sprint del proyecto.	180
Figura 126: Visualización de los artefactos añadidos al sprint del proyecto.	180
Figura 127: Ingreso de la información restante del sprint del proyecto.....	181
Figura 128: Visualización del avance del sprint del proyecto.	181
Figura 129: Creación de diagramas del proyecto.	182
Figura 130: Creación de sprints del proyecto.....	182
Figura 131: Creación de tarjetas en el tablero del proyecto.	183
Figura 132: Visualización del cambio de estado de forma manual de las tarjetas en el tablero del proyecto.	183
Figura 133: Creación de tareas del proyecto.	184
Figura 134: Asignación de datos a las tareas del proyecto.....	184
Figura 135: Creación de tarjetas en las tareas del proyecto.....	185
Figura 136: Visualización de los artefactos y gráficos de las tareas del proyecto.	185
Figura 137: Creación de historias de usuario del proyecto.	186
Figura 138: Referencia al sprint que pertenece la historia del proyecto.	186
Figura 139: Visualización de los gráficos de las historias de usuario del proyecto....	187
Figura 140: Creación de bugs del proyecto.....	187
Figura 141: Ingreso de datos del bug del proyecto.	188
Figura 142: Visualización de los bugs del proyecto.	188
Figura 143: Visualización de las tareas cumplidas, en curso y por cumplirse del proyecto.	189
Figura 144: Creación de carpetas y/o documentos del proyecto, seleccionando su ubicación.	189
Figura 145: Selección entre la creación de una carpeta o documento en el proyecto.	190
Figura 146: Ingreso de datos del documento del proyecto.	190
Figura 147: Visualización de carpetas y documentos del proyecto.	191
Figura 148: Creación de Wikis del proyecto.	191
Figura 149: Visualización de datos de la Wiki del proyecto.	192
Figura 150: Creación de Listas del proyecto.	192
Figura 151: Visualización de las Listas del proyecto.....	193
Figura 152: Visualización de los Foros del proyecto.....	193
Figura 153: Visualización de los Foros del grupo de Desarrolladores del proyecto. .	194
Figura 154: Creación de Archivos del proyecto.	194
Figura 155: Visualización de los Archivos del proyecto.....	195
Figura 156: Visualización de los Rastreadores del proyecto.	195
Figura 157: Asignación de miembros al proyecto.....	196
Figura 158: Visualización de las Notificaciones del proyecto.	196
Figura 159: Soporte de Tuleap en su página oficial.	197
Figura 160: Soporte a través del envío de email.	197
Figura 161: Versionamiento a través Git.....	198

ÍNDICE DE TABLAS

Tabla I: Identificación Herramientas ALM.	17
TABLA II: Identificación Herramientas ALM Libres y Privadas.	18
Tabla III: Contraste entre los Criterios de Evaluación y las Herramientas ALM Privadas.	49
Tabla IV: Contraste entre los Criterios de Evaluación y las Herramientas ALM Libres.	51
Tabla V: Contraste de Información entre los Proyectos de Titulación y las Características del Escenario.	55
Tabla VI: Puntuación de cada uno de los niveles de evaluación.	62
Tabla VII: Valoración de cada una de las Herramientas en base al cumplimiento de los criterios establecidos.	62
Tabla VIII: Valoración de las Herramientas en base al cumplimiento de los requisitos.	63
Tabla IX: Valoración de las Herramientas en base al cumplimiento de los requisitos sin considerar el ámbito social y colaborativo.	66
Tabla X: Extracto del análisis resultante obtenido en Pentaho.	74
Tabla XI: Comparativa entre Metodologías Tradicionales y Web.	74
Tabla XII: Descarte de Metodologías en base a su año y tipo de aplicación.	75

c. INTRODUCCIÓN

La Gestión del Ciclo de Vida de Aplicaciones es un proceso que involucra la implementación, operación, supervisión, soporte y mejora continua del software en desarrollo. Misma que a nivel empresarial proporciona un gran valor al negocio permitiendo aumentar la eficiencia y disminuir los errores; garantizando así un producto software de calidad. Pero es necesario resaltar que son poco conocidas (véase Anexos 2, 3, 4 y 5) tanto en empresas como en instituciones de educación superior. Es así como el Proyecto de Titulación se centra en dar a conocer dichas herramientas a los futuros Ingenieros en Sistemas de la Universidad Nacional de Loja con la finalidad de proporcionar nuevas soluciones tecnológicas encaminadas a elevar el desempeño de los alumnos en el ámbito profesional.

En cuanto a la conformación del documento, está dividido en los siguientes apartados:

El RESUMEN que contiene un extracto del contenido general del Trabajo de Titulación, ÍNDICE que describe la ubicación de los temas abordados, las figuras y tablas que complementan el documento. INTRODUCCIÓN que muestra lo relevante del tema, METODOLOGÍA donde se realiza una descripción de los principales materiales y métodos de investigación empleados en el desarrollo del proyecto, así como también las técnicas aplicadas en el mismo.

La REVISIÓN DE LITERATURA está conformada por capítulos donde se abarcan temas relacionados con Ingeniería de Software (Capítulo I), metodologías ágiles y sus características (Capítulo II), herramientas ALM libres y privadas (Capítulo III) y finalmente las herramientas para predicción o análisis estadísticos (Capítulo IV). Toda esta información previa facilita comprender el desarrollo del presente trabajo proporcionando un contexto claro del mismo.

Seguidamente se tiene los RESULTADOS del proyecto, donde se aprecian tablas y análisis que basadas en la Revisión de Literatura fundamentan la selección de la Herramienta ALM, así como la metodología parametrizada. La herramienta resultante de la evaluación fue Hansoft ALM, misma que se aplicó en un caso de estudio en tiempo real, obteniendo como resultado principal organización y apoyo tecnológico en el desarrollo de software.

La sección de DISCUSIÓN justifica y aclara el porqué de las decisiones tomadas en el desarrollo del proyecto además del hecho de no considerar otras opciones en la ejecución del mismo.

Las CONCLUSIONES del Proyecto de Titulación determinan el éxito del mismo en base a los resultados obtenidos y el cumplimiento de los objetivos planteados. Para finalmente llegar a las RECOMENDACIONES donde se hacen apreciaciones para futuros proyectos, mismas que ayudarán a evitar errores y agilizar su culminación.

d. REVISIÓN DE LITERATURA

El siguiente apartado introduce el contexto de la temática del Proyecto de Titulación, facilitando la comprensión y realización de las tareas que se van a desarrollar.

1. CAPÍTULO I: La Ingeniería de Software en ALM

1.1 Ingeniería de Software.

Disciplina de ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después de que se utiliza. [1] Posee elementos de soporte como métodos, técnicas, estándares y herramientas.

1.2 Procesos de Ingeniería de Software

Son aquellos que identifican el conjunto de tareas y actividades a realizar para la correcta ejecución de un proyecto de software, [2] con la finalidad de desarrollar, innovar y extender el mismo.

1.3 Ingeniería de Software y el Ciclo de Vida del Software

El proceso de desarrollo de software implica lo que se conoce como ciclo de vida del software, que está formado por cuatro etapas:

- Concepción o alcance del proyecto.
- Elaboración que comprende el plan del proyecto y arquitectura.
- Construcción que es el desarrollo del producto.
- Transición que es la transferencia del producto terminado a los usuarios. [3]

Existen varios modelos, paradigmas y filosofías de desarrollo, en los cuales se apoya la ingeniería de software para la construcción del software, entre ellas se tiene a las metodologías de desarrollo ágiles, tradicionales y web. Pero estas pueden complementarse con soluciones tecnológicas conocidas como Herramientas ALM, mismas que se encargan de gestionar el ciclo de vida de las aplicaciones, proporcionando un mayor control y supervisión de los trabajos en desarrollo así como de los miembros del equipo.

2. CAPÍTULO II: Metodologías de Desarrollo

Son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno. [4]

Esta se basa en dos aspectos fundamentales y puntuales, el retrasar las decisiones (permite siempre mantener una satisfacción en el cliente y el éxito del producto) y la planificación adaptativa (permite estar preparado para el cambio y tomar decisiones a lo largo del proyecto); permitiendo potenciar aún más el desarrollo de software a gran escala, logrando un producto final de calidad.

Como resultado se crea un Manifiesto Ágil cuyas principales ideas son:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan. [5]

Algunos de los principales métodos ágiles son:

- XP (eXtreme Programming)
- Iconix
- Cristal Methods
- FDD
- ASD
- DSDM
- Scrum

2.1 XP (eXtreme Programming).

2.1.1 Introducción

Es la metodología más destacada de los procesos ágiles de desarrollo de software. La programación extrema se diferencia de las metodologías tradicionales en que esta pone más énfasis en la adaptabilidad que en la previsibilidad. [5]

XP es una metodología muy utilizada en proyectos con requisitos imprecisos y cambiantes, mismos que poseen un alto riesgo técnico. Se centra especialmente en potenciar las relaciones interpersonales, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo; aspectos muy importantes que constituyen la clave para el éxito en desarrollo de software. [5]

Esta metodología se basa principalmente en la realimentación continua entre el cliente y el equipo de desarrollo, así como la comunicación fluida que debe existir entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios, todo esto con la finalidad de obtener un excelente producto.

Las características fundamentales del método son:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- **Pruebas unitarias continuas:** repetidas y automatizadas, incluyendo pruebas de regresión.
- **Programación por parejas:** tareas de desarrollo realizadas por dos personas en un mismo puesto.
- **Frecuente interacción:** entre el equipo de programación y el cliente o usuario.
- **Corrección:** de errores antes de añadir nueva funcionalidad y realizar entregas frecuentes.
- **Refactorización del código:** reescribir ciertas partes del código con la finalidad de aumentar su legibilidad y mantenibilidad.
- **Propiedad del código compartida:** todo el personal puede corregir y extender cualquier parte del proyecto de ser necesario.
- **Simplicidad en el código:** hacer algo simple y fácil de corregir.

2.1.2 Características

- **Historias de Usuario:** técnica utilizada para especificar los requisitos del software. Son tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Cada una es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.
- **Roles:**
 - **Programador:** escribe las pruebas unitarias y produce el código del sistema.
 - **Cliente:** escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración.
 - **Encargado de pruebas (Tester):** ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
 - **Encargado de seguimiento (Tracker):** proporciona realimentación al equipo en el proceso, verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.
 - **Entrenador (Coach):** Es responsable del proceso global.
 - **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto.
 - **Gestor (Big boss):** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.
- **Proceso**

Este comprende las fases de la metodología.
- **Prácticas**

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. XP apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que describiremos a continuación. [5]

2.1.3 Fases

2.1.3.1 Fase I: Exploración

Los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.

2.1.3.2 Fase II: Planificación de la Entrega

El cliente establece la prioridad de cada historia de usuario, y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

2.1.3.3 Fase III: Iteraciones

Incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Al final de la última iteración el sistema estará listo para entrar en producción.

- Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas
- velocidad del proyecto
- pruebas de aceptación no superadas en la iteración anterior
- tareas no terminadas en la iteración anterior

2.1.3.4 Fase IV: Producción

Requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación.

2.1.3.5 Fase V: Mantenimiento

Mientras la primera versión se encuentra en producción, se debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

2.1.3.6 Fase VI: Muerte del Proyecto

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

2.1.4 Ventajas

- Apropiado para entornos volátiles
- Estar preparados para el cambio
- Planificación más transparente para los clientes
- Permitirá definir en cada iteración cuales son los objetivos de la siguiente
- Permite tener realimentación de los usuarios muy útil
- La presión está a lo largo de todo el proyecto y no en una entrega final

2.1.5 Desventajas

Delimitar el alcance del proyecto con el cliente [5]

2.2 Iconix

2.2.1 Introducción

Metodología que consiste en un lenguaje de modelamiento y un proceso de desarrollo de software práctico. Es un proceso dirigido a la utilización de casos de usos

entendibles por los integrantes del proyecto similar al que utiliza RUP, pero relativamente pequeño y ligero, como XP.

Es un Proceso simplificado ya que unifica un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto, presenta de forma clara las actividades de cada fase y la secuencia que debe ser seguida. Está adaptado a los patrones y ofrece el soporte de UML. [6]

2.2.2 Características

- **Iterativo e incremental:** existen iteraciones entre el desarrollo de modelo del dominio y la identificación de los casos de uso que intervienen en el negocio.
- **Trazabilidad:** Cada paso está referenciado por algún requisito. Siguiendo una relación entre los diferentes artefactos del producto.
- **Dinámica del UML:** Uso dinámico de UML en los diagramas de caso de uso, diagramas de secuencia y de colaboración.

2.2.3 Fases

2.2.3.1 Fase I.- Análisis de requisitos

- Modelo de dominio
- Prototipación rápida
- Modelo de casos de uso

2.2.3.2 Fase II.- Análisis y diseño preliminar

- Descripción de casos de uso
- Diagrama de robustez

2.2.3.3 Fase III.- Diseño

- Diagrama de secuencia
- Completar el modelo estático

2.2.3.4 Fase IV.- Implementación

- Utilizar un diagrama de componentes
- Escribir / Generar código
- Realización de pruebas [6]

2.3 Cristal Methods

2.3.1 Introducción

La familia de metodologías Crystal está basada en los conceptos de Rational Unified Process [RUP] y compuesta o dividida en Crystal Clear, Crystal Yellow, Crystal Orange, Crystal Red, Crystal Maroon, Crystal Blue y Crystal Violet [7]; en donde, esta clasificación por colores está basada en el número de miembros del equipo de trabajo o personas implicadas en el desarrollo, siendo un claro indicador de un tamaño mayor del proyecto y la necesidad de mayor control en el proceso.



Figura 1: Clasificación por colores de la familia Crystal Methods. [7]

Los valores compartidos por los miembros de la familia Crystal se centran en las personas y en la comunicación. [7] En donde el equipo de desarrollo es muy importante, invirtiendo esfuerzos para mejorar sus habilidades y destrezas, fijando políticas de trabajo definidas. Sus principios indican que: el equipo puede reducir trabajo intermedio en la medida que produce código con mayor frecuencia, utilizando mejores canales de comunicación entre las personas; por lo que las convenciones que el equipo adopta tienen que adecuarse y evolucionar.

Existen dos reglas que aplican para toda la familia Crystal:

- 1) Los ciclos donde se crean los incrementos no deben exceder cuatro meses.
- 2) Es necesario realizar un taller de reflexión después de cada entrega para afinar la metodología. [7]

La metodología más documentada es Crystal Clear (CC), que es una optimización de Crystal que puede ser aplicado cuando el equipo consiste en dos a ocho personas que se sientan en el mismo cuarto u oficinas adyacentes [8]. En esta metodología por lo general, los miembros del equipo poseen habilidades o especialidades diferentes. Siendo diseñada para un grupo de personas que quieren construir su propio personal, fuerte y eficaz, capaces de entregar el software frecuentemente.

2.3.2 Características

- Hace menos hincapié en la documentación que otras metodologías más tradicionales.
- Persigue el desarrollo de versiones del proyecto que puedan ser probadas.
- Basada en SCRUM.
- Énfasis especial en pruebas y correcciones al final de cada iteración.

2.3.3 Valores

- Entrega frecuente.
- Comunicación osmótica.
- Mejora reflexiva.
- Seguridad personal.
- Foco.
- Fácil acceso a usuarios expertos.
- Ambiente técnico con prueba automatizada, management de configuración e integración frecuente [9]

2.4 FDD (Feature Driven Development – Desarrollo Basado en Características)

2.4.1 Introducción

Metodología pensada para proyectos con tiempo de desarrollo relativamente cortos. Se basa en un proceso iterativo, con iteraciones cortas (se deciden de acuerdo a las funcionalidades o rasgos) que produce un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar con facilidad. [10]

Las iteraciones o resultados son las que posibilitarán realizar entregas al cliente en tiempos cortos, dando así cumplimiento a los diferentes hitos registrados.

Esta metodología se aplica generalmente con equipos de trabajo pequeños, basándose en la obtención de requisitos. Posibilitando al equipo de desarrollo, trabajar con estos de la manera más conveniente y en caso de que existieran cambios no se vean afectadas las historias de usuarios, ni las descripciones de casos de usos. Además, documenta solo lo que posibilite la integración de desarrolladores fácilmente al proyecto.

FDD se basan en controles propios y una comunicación fluida con el cliente, con la finalidad de poder validar continuamente los resultados y avances de cada iteración, para conseguir una arquitectura sencilla y sin errores.

2.5 ASD (Adaptative Software Development - Desarrollo Adaptable de Software)

2.5.1 Introducción

Es un modelo de implementación de patrones ágiles para desarrollo de software. Su funcionamiento es cíclico y reconoce que en cada iteración se producirán cambios e incluso errores. [11] ASD incorpora el principio de adaptarse al cambio en lugar de luchar contra él.

2.5.2 Características

- Iterativo, ya que se enfoca en un ciclo de vida.
- Orientado en los componentes del software o funcionalidades a desarrollarse.
- Tolerante a cambios.
- Guiado por los riesgos.
- La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

2.5.3 Fases

Se caracterizan por estar enfocadas en la misión, estar basadas en características, ser iterativas, tener marcos de tiempo especificados, ser orientadas por los riesgos y ser tolerantes a los cambios. [7]

2.5.3.1 Especulación

- Inicio para determinar la misión del proyecto.
- Fijación del marco temporal del proyecto.
- Determinación del número de iteraciones y la duración de cada una.
- Definición del objetivo de cada iteración.
- Asignación de funcionalidad a cada iteración

2.5.3.2 Colaboración

Esta fase del llamado ciclo de vida, es donde se construirá la funcionalidad del proyecto, en donde se logre liberar la funcionalidad planificada, explorando nuevas alternativas pudiendo alterar fuertemente el rumbo del proyecto.

2.5.3.3 Aprendizaje

En cada iteración se revisa:

- **Calidad, con criterios de cliente:** Se logra evaluar la calidad desde el punto de vista del cliente, con tal de recoger nuevos requerimientos o cambios.
- **Calidad, con criterios técnicos:** Analizar la calidad del producto revisando el diseño, el código y las pruebas.
- **Funcionalidad desarrollada:** Se discuten los procesos que favorecen el desarrollo del proyecto y se descartan los de influencia negativa que puedan haber afectado el trabajo del grupo y los que no son compatibles con los requerimientos.
- **Estado del proyecto:** Se revisa todo lo efectuado respecto al proyecto en función de lo que inicialmente se había planificado, tanto en la parte técnica como en la humana. [11]

2.6 DSDM (Dynamic Systems Development Method - Método de Desarrollo de Sistema Dinámico)

2.6.1 Introducción

Método que utiliza una serie de conceptos, prácticas y criterios para el desarrollo ágil de software. Su objetivo desarrollar un sistema con las necesidades de la empresa tanto en tiempo como en presupuesto. [9] Es un método ideal para proyectos con restricciones temporales o requerimientos cambiantes; fijándose primero el tiempo y el coste, para poder determinar las funcionalidades que se pueden implementar en el producto.

2.6.2 Características

- El desarrollo es iterativo e incremental.
- Cambios reversibles.
- El equipo toma decisiones sin autorización de superiores.

- Se realizan entregas del producto frecuentemente.
- Pruebas de calidad a lo largo del proceso de desarrollo integradas en el ciclo de vida.
- Cooperación entre los desarrolladores, usuarios y Stakeholders.

2.6.3 Fases

- Fase 1 el pre-proyecto.
- Fase 2 el ciclo vital del proyecto.

Etapas en el ciclo vital del proyecto DSDM: Estudio de viabilidad, Estudio de la empresa, Iteración del modelo funcional, Diseño e iteración de la estructura, Implementación,

- Fase 3 el post-proyecto. [9]

Esta metodología se ha convertido gradualmente en el marco de referencia número uno para el desarrollo rápido de aplicaciones (RAD). [12] Además es utilizada tanto en proyectos grandes (dividiéndolo en componentes para ser trabajados en quipos) como en pequeños.

2.7 Scrum

2.7.1 Introducción

Proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. Este se realiza en forma iterativa e incremental y cada ciclo o iteración (ciclo corto de construcción repetitivo) termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las iteraciones en general tienen una duración entre 2 y 4 semanas. Scrum por lo general es utilizado como marco metodológico para otras prácticas de ingeniería de software como RUP o Extreme Programming. [13]

Scrum está diseñado para adaptarse a los cambios que se realicen en los requerimientos, ya que estos por lo general son revisados y ajustados durante el proyecto en intervalos muy cortos y regulares. Logrando de esta manera que el producto se adapte, resuelva y solventa las necesidades del cliente, aumentando así la satisfacción del mismo.

En Scrum, el equipo se focaliza en una única cosa: construir software de calidad. Por el otro lado, la gestión de un proyecto Scrum se focaliza en definir cuáles son las

características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en remover cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. Se busca que los equipos sean lo más efectivos y productivos posible. [13]

Scrum tiene un conjunto de reglas muy pequeño y muy simple, basado en los principios de:

- inspección continua
- adaptación
- auto-gestión
- innovación.

En el cual el cliente se ve entusiasmado, comprometiéndose con el proyecto al ver su evolución iteración a iteración, alineando el desarrollo de este con los objetivos de negocio de su empresa.

Pero al emplear esta metodología no solo los desarrolladores encuentran un ámbito propicio para explotar y desarrollar sus capacidades profesionales, sino que también encuentran la motivación necesaria entre los integrantes del equipo.

SCRUM se aplica a proyectos pequeños como a proyectos grandes y donde los requisitos de entornos complejos cambian con cierta frecuencia. Su principal objetivo es lograr una correcta evaluación de los cambios del entorno, adaptarse constantemente al "caos" de intereses y necesidades. [14]

2.7.2 Características

- Equipos autodirigidos.
- Colaboración estrecha con el cliente.
- Predisposición y respuesta al cambio.
- Prefiere el conocimiento tácito de las personas al explícito de los procesos.
- Desarrollo incremental con entregas funcionales frecuentes.
- Comunicación verbal directa entre los implicados en el proyecto.
- Motivación y responsabilidad de los equipos por la auto-gestión, auto-organización y compromiso.
- Simplicidad. Supresión de artefactos innecesarios en la gestión del proyecto. [9]

2.7.3 Fases

- **Planificación:** Definición de una nueva funcionalidad requerida por el sistema de basado en el sistema del conocimiento como en su conjunto;
- **Desarrollo:** Desarrollo esta nueva función respetando la tiempo asignado, los requisitos y calidad. Estos elementos definen el final del ciclo el desarrollo;
- **Cierre:** Preparación de actividades persistentes de entrega de productos: Caja Blanca pruebas, ensayos, Caja Negra, Documentación del usuario, Formación y Comercialización. [14]

3. CAPÍTULO III: Gestión del Ciclo de Vida del Software

Proceso continuo de la gestión de la vida de una aplicación a través de la gobernanza, el desarrollo y el mantenimiento. ALM es el matrimonio de la gestión empresarial de la ingeniería de software, posible gracias a herramientas que facilitan e integran la gestión de requisitos, arquitectura, codificación, pruebas, seguimiento. [15]

Para obtener la lista de herramientas disponibles primeramente se realizó una la búsqueda de herramientas ALM en la web, utilizando el Método de Búsqueda de Información Secuencial, que consiste en recurrir al uso de un buscador el cual requiere solo introducir palabras claves para dar con la información que se desea obtener [16], de este cúmulo de información obtenida únicamente se utilizó aquella que pertenece a páginas web confiables y pertinentes de internet (paginas oficiales de herramientas) . Es así como esta búsqueda arrojó los siguientes resultados:

1. Tuleap
2. Mylyn
3. Endeavour Agile ALM
4. TOPCASED ALM
5. Polarion
6. Hansoft(buscar)
7. MantisBT
8. Subversion
9. TOMOS
10. TraceCloud
11. Team Foundation Service
12. StarTeam Express

13. Jenkins
14. OrangeScrum
15. Taiga gestión de proyectos
16. Rally gestión de proyectos
17. Atlassian JIRA
18. IBM Rational Team Concert
19. HP Application Lifecycle Management
20. SAP Solution Manager
21. TraceCloud ALM

Una vez que se obtuvo la lista de posibles herramientas, se realizó la búsqueda de cada una de ellas para verificar si pertenecen al grupo de herramientas ALM, cuyo resultado se muestra a continuación:

Tabla I: Identificación Herramientas ALM.

Hallazgos		
Herramientas	ALM	Otros
Tuleap	✓	
Mylyn	✓	
Endeavour Agile ALM	✓	
Topcased ALM	✓	
Polarion	✓	
Hansoft	✓	
MantisBT		Gestor de tareas
Subversion		Plugin
TOMOS	✓	
TraceCloud	✓	
Team Foundation Service	✓	
StarTeam Express	✓	
Jenkins		Plugin
OrangeScrum		Gestor de proyectos
Taiga		Gestor de proyectos
Rally		Gestor de proyectos
Atlassian JIRA	✓	
IBM Rational Team Concert	✓	
HP Application Lifecycle Management	✓	
TraceCloud ALM	✓	
SAP Solution Manager	✓	

TABLA II: Identificación Herramientas ALM Libres y Privadas.

Hallazgos		
Herramientas ALM	Libres	Privadas
Tuleap	✓	
Mylyn	✓	
Endeavour Agile ALM	✓	
Topcased ALM	✓	
Polarion		✓
Hansoft	✓	
TOMOS		✓
TraceCloud		✓
Team Foundation Service	✓	
StarTeam Express	✓	
Atlassian JIRA		✓
IBM Rational Team Concert		✓
HP Application Lifecycle Management		✓
TraceCloud ALM		✓
SAP Solution Manager		✓

3.1 Herramientas ALM Privadas.

3.1.1 Atlassian JIRA

Esta herramienta es diseñada para que todos los miembros del equipo de software puedan planificar, supervisar y publicar un excelente producto software.

Características

- **Planificar:** Crear historias de usuario e incidencias, planificar sprints y distribuir tareas entre el equipo de software.
- **Supervisar:** Priorizar y analizar el trabajo del equipo con total visibilidad.
- **Lanzamiento:** Con confianza y seguridad, sabiendo que la información es siempre la más actualizada.
- **Crear informes:** Mejora el rendimiento del equipo con datos visuales en tiempo real.
- **Seleccionar o diseñar un flujo de trabajo propio**
- **Delegar gestión de prints**
- **Integración**
- **7 días de prueba**

Costos

- Equipos pequeños \$10 dólares al mes hasta 10 usuarios.
- Equipos en crecimiento \$75 dólares al mes para más de 15 usuarios. [17]

3.1.2 IBM Rational solution for Collaborative Lifecycle Management

Solución de gestión de ciclo de vida que incluye a IBM Rational Team Concert TM, IBM Rational DOORS® Siguiendo Generación y productos de IBM Rational Quality Manager. La inclusión y la integración de estos productos en una solución ofrecen:

- gestión de requisitos
- gestión de calidad
- gestión de cambios y configuración
- proyecto de planificación
- capacidades de seguimiento en una sola plataforma

Esta herramienta permite hacer frente a las necesidades inmediatas de un proyecto y añadir capacidades a medida que evolucionan sus necesidades.

Rational Collaborative Lifecycle Management incluye una plantilla integrada Scaled marco ágil (SAFE) para acelerar la adopción de las prácticas de una caja de seguridad codificado, base de conocimiento público utilizado para escalar el desarrollo Lean-Agile en toda la empresa.

Características

- Planificación en tiempo real diseñado para vincular los planes de trabajo real y mejorar el tiempo de entrega.
- Trazabilidad del ciclo de vida para obtener información, tomar mejores decisiones y mejorar la gestión de entrega de software y previsibilidad.
- En contexto de colaboración sobre los diseños, requisitos, planes, código y tareas a tomar mejores decisiones a partir de una única fuente de la verdad.
- Desarrollo de inteligencia para establecer objetivos medibles y rastrear fácilmente.
- Mejora continua para proporcionar una mejora gradual y la mejora avance a través de la automatización y la reutilización de las mejores prácticas. [18]

3.1.3 IBM Rational Team Concert

Es una solución de gestión del ciclo de vida del software que permite colaboración contextual en tiempo real para equipos distribuidos. Basado en la plataforma de IBM Rational Jazz, Rational. Proporciona y ofrece:

- configuración del proceso
- marco de guía y obligatoriedad que da soporte a todo el entorno de entrega de software
- beneficios de gestión de cambios de colaboración
- modelos de precios flexibles
- software de servidor de forma gratuita
- soporte para diversas plataformas
- licencias basadas en rol en un solo lanzamiento, lo que permite la implementación de componentes individuales ahora y en el futuro.

Rational Team Concert facilita la colaboración de los equipos y permite una entrega de software más rápida.

- Mejora la colaboración del equipo con características integradas, que incluye la gestión de la configuración de software, la creación y los elementos de trabajo.
- Proporciona una gran visibilidad de actividades de proyectos y progreso del equipo con tableros de instrumentos de varios niveles y características de creación de informes.
- Facilita la planificación y ejecución de proyectos ágiles o formales con plantillas y herramientas de planificación. La existencia de procesos coherentes mejoran la calidad del software.
- Ayuda a mejorar la productividad con control de origen avanzado para equipos distribuidos geográficamente. [19]

3.1.4 HP Application Lifecycle Management

Permite a los equipos proporcionar aplicaciones de gran calidad con una velocidad y agilidad superiores. Combina productos, integraciones de herramientas de desarrollo, un ecosistema de partners y un enfoque independiente de la tecnología para mejorar

la calidad de las aplicaciones, la colaboración y la productividad para evitar retrasos en los proyectos.

Características

- Proporciona un seguimiento integral de los dispositivos ALM para conjuntos de cambio del código fuente a los equipos de entrega de aplicaciones.
- Gestión de proyectos Agile
- Gestión de calidad
- Inteligencia del ciclo de vida de la aplicación
- Integraciones de código abierto [20]

3.1.5 StarTeam

Proporciona una ágil gestión del cambio para las organizaciones con múltiples repositorios. Y además permite agilizar los flujos de trabajo mediante la conexión de herramientas, procesos y equipos a través de un único repositorio.

Características

- mantiene y sigue los cambios sobre cualquier activo de software
- control sobre el desarrollo distribuido y colaborativo
- visibilidad
- trazabilidad
- gestión de la configuración

Además esta herramienta se asegura que el o los cambios realizados se gestionen, implementen, analicen y sea puesto en libertad adecuadamente durante todo el ciclo de vida de desarrollo de software (SDLC). Proporciona una fuente única de la gestión del cambio de equipos, herramientas y procesos.

StarTeam se puede integrar en el entorno actual y entregar inmediatamente una vista conectada de los activos SDLC de otras herramientas. Esto permite una amplia gama de capacidades a través de todos los activos integrados

- **Interesados:** apalancamiento de StarTeam completa de versiones, la línea de base y la notificación de todos sus activos para ganar visibilidad sobre los cambios que definen cada entregable.

- **Desarrolladores:** acceder a una vista consolidada para poder colaborar, revisar y entregar sus cambios al cliente de manera rápida.
- **Todo el equipo:** lograr la visibilidad y conectividad necesaria de todo su ciclo de vida del software. [21]

3.1.6 SAP Solution Manager

Es una plataforma estándar para la gestión del ciclo de vida de aplicaciones (ALM) y el eje central de la gestión del ciclo de vida de los sistemas SAP y las soluciones en la nube. Está incluido en el contrato de mantenimiento de SAP proporcionando todo el contenido necesario para implementar de manera eficiente, operar, supervisar y apoyar a todas las aplicaciones de SAP y no SAP, asegurando la entrega de servicios de alta calidad y soporte para los procesos de negocio de misión crítica, permitiendo así un mejor funcionamiento y una aplicación más rápida a un menor costo. Encargada de la innovación, gestiona el ciclo de vida de aplicaciones, soluciones y se ejecuta, de extremo a extremo, en una plataforma de negocio y de TI integrada.

Además de proporcionar la funcionalidad de SAP en sí, SAP Solution Manager integra las otras herramientas para garantizar un enfoque global, facilitando la colaboración eficiente entre la empresa y la organización de soporte de SAP. [22]

3.1.7 TraceCloud ALM

TraceCloud es un AWS -basado en herramienta de ALM que incluye un sistema integrado de gestión de requisitos, defectos y pruebas. Cuenta con un cambio en una función de la tarjeta de control para el seguimiento de la aprobación y cuadros de mando configurables, métricas para la gestión a nivel de proyecto, la liberación, la línea de base, carpeta y usuario.

Debido a que TraceCloud es un plug-and-play SaaS servicio web REST construida en notación de objetos JavaScript, se integra bien con los sistemas existentes. Como oferta SaaS ALM, soporte y mantenimiento se incluyen en la cuota de suscripción. TraceCloud ofrece seguridad con SSL de 128 bits, así como el cifrado y la escalabilidad de la infraestructura de la nube elástica de Amazon. [23].

Características

- **Facilidad de uso:** trazabilidad, atributos, estado de prueba, finalización, registro, información de versión, estado de aprobación y la información de línea de base se encuentra convenientemente en una sola página.
- **Control de cambios:** definir las funciones y controlar los permisos de estas funciones en un nivel sub-carpeta.
- **Flexibilidad:** se puede utilizar su existente proceso de gestión de requisitos y configurar TraceCloud para adaptarse a sus necesidades.
- **Ágil o una cascada:** si está utilizando ágil Scrum o un modelo de desarrollo iterativo de la cascada, las necesidades de gestión de requisitos no desaparecen.
- **Trazabilidad completa:** sistema diseñado con la facilidad de uso, escalabilidad y el rendimiento en mente.
- **CIA:** Análisis del Impacto del Cambio [24]

3.1.8 Tomos

Es la web de primera clase 2.0, solución SaaS para la gestión de calidad de la aplicación del ciclo de vida, que desemboca en una base de datos centralizada. Lo que facilita la trazabilidad completa de los activos de los requisitos para poner a prueba a los casos de defectos y proporciona una pista de auditoría completa de los cambios a nada. [25] Esta herramienta es dirigida a los analistas, desarrolladores y probadores, cuya interfaz ha sido diseñada para proporcionar un semejante aspecto y la sensación de Microsoft Office, lo que minimiza el tiempo de aceleración.

Al ser una aplicación SaaS, no hay costos de instalación o mantenimiento y las actualizaciones se realizan a una base de código de central, lo que permite despliegues rápidos de nuevas características y correcciones.

Categorías

- Ágil
- Gestión del Ciclo de Vida
- Presupuestos y previsiones
- Gestión del Cambio (Versión Control)
- Colaboración
- Gestión de documentos
- Gestión de Empresa Portafolio
- Gestión de programas
- Contabilidad de proyectos
- Gestión de proyectos
- Gestión de la Cartera de Proyectos
- Informes
- Gestión de requisitos
- Administración de recursos

- Seguimiento de gastos
- Compartición de archivos
- Seguimiento de problemas
- Seguimiento de las métricas
- Gestión de proceso
- Automatización de servicios profesionales
- Gestión de riesgos
- Programación
- Gestión De Servicios
- el tiempo de mantenimiento
- Gestión de flujo de trabajo
- Administración de personal

Características

- Seguimiento de Punto de Acción
- Aprobación de gestión / Flujo de trabajo
- los procesos de archivo
- Plantillas de archivado
- Seguimiento de asignación
- Beneficios Documentación y Seguimiento
- Mejor Gestión de Prácticas
- Facturación y Seguimiento de cargo
- Presupuestación & Forecasting
- / Cambio de gestión de versiones
- Colaboración
- Interfaz de usuario personalizada
- Los campos personalizables
- Funcionalidad personalizable
- Informes personalizable
- Importar / Exportar datos a MS Excel
- Importar / Exportar datos de MS Project
- Gestión de documentos
- Integración de correo electrónico
- Estimación
- Dashboarding Ejecutivo
- Seguimiento de gastos
- Gestión de fallos
- Compartición de archivos
- diagramas de Gantt
- Gestión de Problemas
- Gestión del conocimiento
- La integración de sistemas heredados
- Administración de personal
- Metodología / Gestión de Procesos
- Gestión de métricas
- Seguimiento de Milestone
- Acceso móvil
- Multi moneda
- Multi lenguaje
- Seguimiento ciento-Completa
- Pert Cartografía
- Gestión de la cartera
- Seguimiento de los progresos
- Contabilidad de proyectos
- proyecto de Estimación
- Personalización de plantillas de proyecto
- Integración plantilla de proyecto
- Desarrollo del proyecto
- Gestión de la calidad
- Gestión de registros
- Informes (Financiera)
- Informes (Progreso)
- Gestión de requisitos
- Administración de recursos
- Gestión de riesgos
- Programación
- Kit de desarrollo de software
- Administración de tareas
- Soporte personalización de plantillas
- Tiempo
- Seguimiento Costo Total

Plataformas

- Linux / Unix
- Mac
- Fuente abierta
- Otro
- Software como servicio
- Windows [25]

3.2 Herramientas ALM Libres.

3.2.1 Tuleap

Es una suite Fuente libre y abierto para mejorar la gestión de proyectos de software y conectar con los miembros del equipo. Con una única solución basada en web, CTO, gestores de proyectos, desarrolladores, gerentes de calidad y otros usuarios pueden construir, implementar proyectos inteligentes y de calidad. [26]

Tuleap proporciona herramientas para:

- gestión de proyectos
- tareas
- cambios
- defectos
- documentos
- control de versiones
- integración continua
- colaboración social

Esta herramienta posee alta capacidad de personalización, permitiendo al usuario adaptar cada herramienta a sus procesos y requerimientos del negocio. Además se proporciona bajo la licencia GPLv2, Tuleap sirve o puede utilizarse para un número ilimitado de usuarios, tiempo y proyectos.

Con Tuleap, usted será capaz de:

- proyectos del plan y el monitor,
- gestión del ciclo de vida de desarrollo de software: código fuente, construye ...,
- requisitos de la pista, tareas, incidencias, etc.
- producir documentos y comunicados,
- favorecer la colaboración entre los miembros del proyecto. [26]

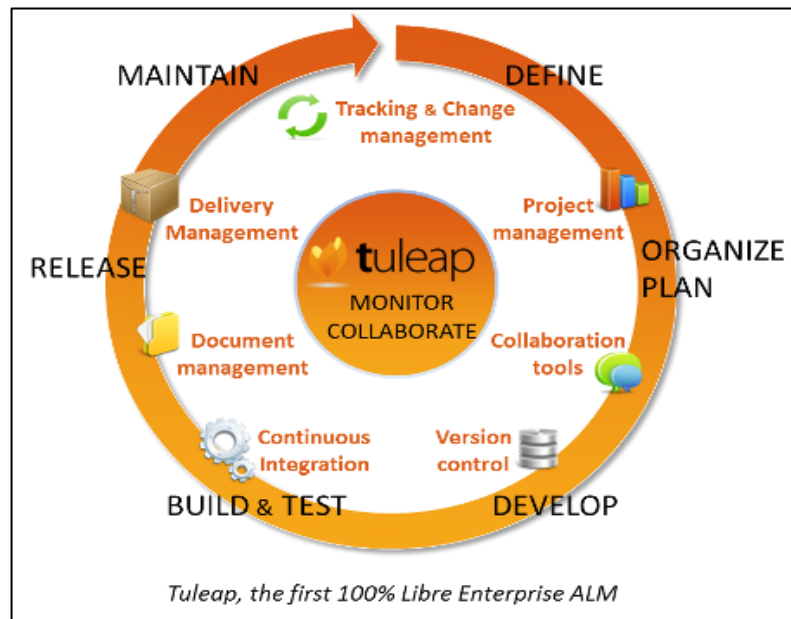


Figura 2: Características de Tuleap. [27]

Con Tuleap, se puede:

- Gestionar todo el ciclo de vida de los proyectos de software con las dos herramientas de gestión de proyectos de desarrollo
- Integración con herramientas de código abierto como Git, Gerrit o Jenkins
- Establecer nuevos proyectos en unos pocos clics : el espacio de trabajo está listo y planificar su proyecto, entradas de pista, administrar el código fuente y colaborar
- Manejo de diferentes metodologías de software: ágil, cascada, etc.
- Personalizar las herramientas a sus necesidades específicas. [27]

Organizar y Planificar

- Recoger y clasificar los requisitos de los clientes y las necesidades
- Planear lanzamientos o sprints
- Asignar tareas a los miembros del proyecto

Monitorear el avance del proyecto

- Realizar un seguimiento de los avances y el trabajo restante
- Obtener estado en tiempo real sobre los proyectos
- Saber exactamente lo que tiene que ver con su escritorio personal
- Crear gráficos y los informes sobre la marcha [28]

3.2.2 Mylyn

Es el marco de gestión de tareas y la aplicación del ciclo de vida (ALM) para Eclipse. Se ofrece:

- La interfaz enfocada al objetivo revolucionario se vuelve a alinear el IDE torno a las tareas de manera que sólo se ve el código que es relevante.
- Una herramienta de gestión de tareas para los desarrolladores de promediado 1 millón de descargas / mes, Mylyn es la herramienta más popular para IDE ALM.
- Un amplio ecosistema de Agile ALM y de interacción con decenas de extensiones integrar Mylyn con herramientas de colaboración de ALM y desarrolladores. [29]

El objetivo de Mylyn es reducir la sobrecarga de información y hacer que la multitarea sea fácil.[29] Además se encarga de poner la información que el usuario necesita a su alcance, aumentando así la productividad y facilitando el intercambio de conocimientos. Mylyn integra la edición rica y fuera de línea para disfrutar de herramientas ALM, y supervisa su actividad de programación para crear un "contexto de la tarea" que se centra su espacio de trabajo y automáticamente vincula todos los objetos relevantes para la tarea-en-mano. [29]

La misión de Mylyn proyecto es proporcionar:

1. marcos y APIs para tareas basadas en Eclipse y Gestión del ciclo de vida de aplicaciones (ALM)
2. herramientas ejemplares para la programación enfocado al objetivo en el IDE de Eclipse.
3. implementaciones de referencia para las herramientas de ALM de código abierto usados por la comunidad Eclipse y los estándares abiertos como ALM OSLC

El proyecto está estructurado en subproyectos, cada uno representando una categoría de ALM y proporcionar APIs comunes para disfrutar de herramientas ALM específicos. Los consumidores principales de este proyecto son ALM ISV y otros en adoptar marcos Eclipse ALM.

Mylyn hace que las tareas de una parte de la primera clase Eclipse, e integra la edición rica y fuera de línea para los repositorios, como Bugzilla, Trac, y JIRA. Una vez que sus tareas se integran, Mylyn supervisa su actividad de trabajo para identificar la información relevante para la tarea-ahí, y utiliza este contexto tarea de enfocar la interfaz de usuario de Eclipse en la información interesante, ocultar el poco interesante, y encontrar lo que está relacionado de forma automática. Esto pone a la información que necesita para realizar su trabajo a su alcance y mejora la productividad mediante la reducción de la búsqueda, el desplazamiento y la navegación. Al hacer contexto de la tarea explícita Mylyn también facilita la multitarea, la planificación, la reutilización de los esfuerzos del pasado, y compartir experiencia. [30]

3.2.3 Endeavour Agile

Es una solución de código abierto liberado bajo la versión Licencia Pública General de GNU 3, impulsa la creación de soluciones de sistemas de empresas a gran escala mediante el apoyo a los procesos de desarrollo ágil de software iterativo e incremental, con especial énfasis en las prácticas impulsadas de casos de uso. [31]

Correlacionar los artefactos de ciclo de vida con los esfuerzos de desarrollo

Endeavour Agile ALM ha sido diseñado de tal forma que sea fácil de usar, intuitiva y realista, evitando características innecesarias que podrían aumentar la complejidad, dificultando su explotación y comprensión por parte del usuario. Esta herramienta cuenta con soporte para los casos de uso, iteraciones, plan del proyecto, las solicitudes de cambio, rastreo de defectos, casos de prueba, planes de prueba, tareas, gestión de documentos, informes y muchos otros artefactos de proceso. Ha sido diseñado con el objetivo de facilitar su uso, ser intuitiva y sobre todo realista, evitando características innecesarias que podrían aumentar la complejidad. [31]

Seguir el progreso de su equipo de desarrollo de software ágil

Endeavour Agile ALM proporciona informes y navegación del repositorio de subversión que permitirá seguir fácilmente el progreso del equipo de desarrollo de software y soporte para la integración continua que mantendrá hasta la fecha conjunto de aplicación construye listo para la prueba y demostración. [31]

Solución basada en la web a través del navegador con capacidades de cliente enriquecido

Endeavour Agile ALM se ejecuta en cualquiera de estos navegadores:

- Mozilla Firefox
- Internet Explorer
- Apple Safari
- Google Chrome
- Opera

Funcionalidad principal

Agile Project Management, Presupuesto, colaboración, seguimiento de problemas, aprendizaje y apoyo, notificaciones, informes, gestión de recursos, gestión de tareas y gestión de proyectos tradicional.

Usuario previsto

De pequeña o mediana empresa.

Mejor zona funcional

Seguimiento de problemas.

Mejor zona funcional

Gestión de tareas. [32]

Además proporciona una interfaz gráfica de usuario con una apariencia similar a las aplicaciones de escritorio tradicionales, facilitando de esta manera su uso y adaptabilidad del usuario.

Características

- Casos de uso
- Iteraciones
- Plan de proyecto
- Solicitudes de cambio
- Dependencias de tareas
- Seguimiento de defectos
- Casos de prueba
- Buscando en la subversión
- proyecto Wiki
- Notificaciones de Correo Electrónico
- Glosario del proyecto
- Gráfico de Gantt del proyecto
- Proyectos
- Asignaciones de miembros del

- Los planes de prueba
- Tareas
- Foros
- Gestión de documentos
- Integración continua
- proyecto
- Integración de LDAP
- Los defectos de informe
- Iteración informe de flujo acumulativo
- Los defectos de iteración por estatus
- Los defectos de iteración por prioridad

Endeavour Agile ALM está disponible en los siguientes idiomas: Inglés, español, alemán, chino, holandés, francés y polaco. [33]

Pros:

- **Los flujos de trabajo mejores Fit:** Permite a los usuarios resolver los problemas a través de múltiples flujos de trabajo permitiendo manejar los errores de manera diferente.

Contras:

- **Sin opciones de personalización**
- **Seguimiento duro:** Endeavour puede manejar el seguimiento, pero no puede dar los datos en horas del proyecto.

Revisión de expertos

Endeavour está construido específicamente para las empresas de tamaño pequeño a medio. Es basado en la nube; mantiene los gestores de proyectos conectados siempre, incluso si no tienen su equipo de trabajo con ellos.

3.2.4 Topcased o Polaris

Es un Grupo de Trabajo de la Industria Eclipse creado por los grandes grupos industriales y por los proveedores de herramientas para colaborar en la creación y el apoyo de herramientas de código abierto para el desarrollo de sistemas embebidos (diseñados para realizar una o algunas pocas funciones dedicadas).

Metas

- Proporcionar medios de colaboración entre empresas de usuarios finales
- La organización de servicios comerciales sostenibles y ecosistemas alrededor de componentes de código abierto
- El fomento de los intercambios entre académicos y socios de la industria
- La gestión de la calidad y madurez de herramientas y componentes de prototipos tempranos de la investigación a través de la obsolescencia
- Proporcionar los documentos y kits de cualificación requeridos para la certificación
- Reconociendo la madurez del proyecto y la empresa de know-how y el compromiso a través de un proceso de marca.

Los dominios centrales

- Modelado
- Compilación, depuración, trazando
- Proceso de aplicación del ciclo de vida y las herramientas de gestión - gestión de configuración, gestión de cambios, seguimiento de problemas, gestión de requisitos, los informes del proyecto
- De prueba y verificación marcos y herramientas dirigidas a los métodos integrados de software, análisis estático, simulación y validación de componentes tempranos Embedded (RTOS, tiempo de ejecución, middleware, etc.)
- SoC (System on Chip) simulación y lógica de hardware (VHDL, SystemC, etc.).
[34]

3.2.5 Hansoft

Es una herramienta para la colaboración en equipo y la gestión en el desarrollo de software ágil de productos y servicios. Tiene un conjunto de características que abarca más ágil ALM, PPM (cartera de productos y gestión de programas) y Colaboración Social.

Hansoft es aplicado en industrias como: electrónica, aeroespacial, desarrollo de juegos, Telecom y servicios en la nube, además es muy utilizado en equipos de Europa, América del Norte, Asia y Australia. [35] Permitiendo ejecutar tareas, metodologías y proyectos en paralelo, haciendo más fácil la transferencia de datos

evitando perderlos. Posee una interfaz intuitiva que simplifica la complejidad del trabajo para el equipo, ajustando los procesos a las necesidades e informando de los cambios a cada uno de los miembros.

Posee un completo sistema de chat para la comunicación instantánea con los miembros del equipo, individual o de grupos, donde los registros de chat son recogidos en el servidor. [36]

Características

- **Métodos Side-by-Side:** Independientemente del método que se emplee, los equipos pueden trabajar hacia un objetivo común, permitiendo rastrear el progreso.
- **Gran Escala:** Permite ver el panorama general y profundizar en las actualizaciones en tiempo real tanto de equipos como de personas. Además se adapta a la forma de trabajar de la empresa.
- **Las métricas accionables:** realiza un seguimiento de las métricas para tomar mejores decisiones, cabe destacar que esta herramienta faculta a todos los miembros del proyecto con la finalidad de obtener mejores resultados.
- **Mejora continua:** Permite adaptar y mejorar de forma continua gracias a su flexibilidad, respondiendo así a cambios de forma adecuada y oportuna a futuros y repentinos cambios.
- **Seguridad a nivel de empresa:** con el protocolo de cifrado TLS 1.2 estándar entre servidores y clientes.
- **Construido para la velocidad:** En tiempo real a través de miles de usuarios que disfrutaban de esta experiencia de trabajo.
- **Actualizaciones y soporte incluido:** las actualizaciones sólo son necesaria en el servidor, además se lanzan nuevas versiones del software para el final de (casi) todos los sprints. [37]

Características

- | | |
|--|---------------------------------|
| • Aprobación de gestión / Flujo de trabajo | • Multi-Lenguaje |
| • Seguimiento de asignación | • Gestión de la cartera |
| • Colaboración | • Seguimiento de los progresos |
| • Los campos personalizables | • proyecto de Estimación |
| • Importar / Exportar datos a Excel | • Personalización de plantillas |

- Importar / Exportar datos de MS Project
- Gestión de documentos
- Integración de correo electrónico
- Estimación
- Dashboarding Ejecutivo
- Gestión de fallos
- Compartición de archivos
- diagramas de Gantt
- Gestión de Problemas
- Metodología / Gestión de Procesos
- Gestión de métricas
- Seguimiento de Milestone
- Integración plantilla de proyecto
- Desarrollo del proyecto
- Gestión de la calidad
- Informes (Progreso)
- Gestión de requisitos
- Administración de recursos
- Gestión de riesgos
- programación
- Kit de desarrollo de software
- Administración de tareas
- tiempo de seguimiento
- Administración de personal

Categorías

- Ágil
- Gestión del Ciclo de Vida
- Colaboración
- Gestión de documentos
- Gestión de Empresa Portafolio
- Seguimiento de problemas
- Gestión de programas
- Gestión de proyectos
- Gestión de la Cartera de Proyectos
- Administración de recursos
- Programación
- Gestión de flujo de trabajo

Plataformas

- Linux / Unix
- Mac [35]
- Windows

3.2.6 Team Foundation Service

Es un servicio de Microsoft en el que se proporciona la herramienta de gestión de ciclo de vida de aplicaciones TFS bajo la modalidad de SaaS (Software as a Service). Para poder usar esta herramienta, bastará con darse de alta en una página web. [38] Las actualizaciones de esta herramienta serán gratuitas e inmediatas, para que así la empresa pueda dedicarse a lo que es realmente importante, como la creación del software y la metodología. Además es compatible con los navegadores más modernos y posee tres plantillas de procesos: Scrum, Agile o CMMI.

Características

- **Gestión del equipo y su trabajo**

- **Gestión del backlog:** conjunto de vistas que permitirán organizar los requerimientos e ir asignándolos en cada iteración.
- **Sprint actual:** para el equipo de trabajo se pone a su disposición un *task board* donde se pueden gestionar las tareas de la iteración actual y gestionar los *post-it* virtuales, para marcar el trabajo como en progreso o terminado.
- **Gestión de los elementos de trabajo:** herramienta para crear y gestionar, desde el portal web, consultas sobre todos los elementos de trabajo como tareas, requerimientos, bugs, impedimentos, etc. [38]
- **Control de código fuente**

Se puede emplear cualquier lenguaje de programación: C#, C++, Java o Python. Obteniendo de esta manera un control total, tanto online como en modo local, del código fuente de los proyectos en desarrollo.
- **Automatización de builds, pruebas y despliegues**

La gestión de builds y el estudio de sus resultados se pueden realizar directamente desde el portal web, o bien desde Team Explorer. Beneficiándose de una máquina en la nube para realizar estas operaciones de forma automática o bajo demanda. [38]
- **Control de versiones**

Almacenar y colaborar en código de cualquier lugar con proyectos de equipos privados, respaldado por el control de versiones. Es un repositorio masivamente escalable o varios repositorios Git para conseguir la máxima flexibilidad.
- **Lenguajes y herramientas**

El control de versiones admite cualquier lenguaje, así como cualquier cliente Git (incluido Xcode). Los equipos de Java pueden obtener acceso al código y a los elementos de trabajo a través de un complemento gratuito para Eclipse y ejecutar versiones de integración continua basadas en archivos de configuración de Ant o Maven.
- **Integración continua**

Permite detectar problemas de calidad pronto con versiones de integración continua (CI) que compilan y prueban la aplicación automáticamente después de cualquier cambio de código. Además implementar automáticamente aplicaciones o sitios web que superen las pruebas o modele sus canalizaciones de lanzamiento para que coincidan con los procesos de lanzamiento existentes. [39]

Limitaciones

- No se puede crear plantillas de procesos propias.
- No se puede integrar con otros servicios empresariales o de federación.
- No se soportan las funciones de generar informes en el producto on-premises.

3.2.7 Star Team Express

Es la única fuente de la verdad para la gestión del cambio a lo largo del ciclo de vida de desarrollo de software. Ofrece cambios a través de cualquier herramienta y el repositorio, su gestión de configuración de software funciona para los equipos de desarrollo de software centralizados y distribuidos geográficamente, manteniendo la visibilidad y trazabilidad en todos sus activos de software, que actúa como su única fuente de verdad. [40] Promueve la colaboración y la comunicación entre el equipo a través de la centralización de las actividades del proyecto y de los recursos digitales. [41] Esta herramienta se encuentra disponible para su descarga inmediata, ofreciendo capacidades y funcionalidad para grupos de desarrollo de hasta 10 usuarios. Asegurando que los miembros del equipo estén siempre al día con los cambios del proyecto en desarrollo en el momento en el que estos ocurren.

Características

- **Versiones de archivos y el seguimiento de cambios:** versiones completas, ramificación y seguimiento de cambios para todo el ciclo de vida de desarrollo de software, abarcando todo el paisaje de herramientas ALM.
- **Flujo de trabajo personalizable:** proporciona un motor de flujo de trabajo flexible e integrado desde los más simples a los más complejos, mejorando así la colaboración.
- **Liberar y Construcción de Gestión:** es el estado de cambios para cada proyecto de construcción y liberar en un solo vistazo, permitiendo la generación de informes de liberación más eficientemente, proporcionando una visibilidad constante en todos los artefactos de liberación.
- **End-to-end análisis de impacto:** es una función de análisis de datos, que ofrece análisis inteligentes y presentación de informes con la finalidad de realizar el análisis y evaluación de tendencias. Brindando apoyo a las decisiones basándose en los hechos, facilitando la realización de cambios.

Esta herramienta permite a los desarrolladores centrarse en el desarrollo de software y colaborar con otros miembros del equipo. Además permite a los administradores acceder a la información, facilitando la toma de aquellas decisiones que son críticas con rapidez y confianza.

3.3 Selección de la Herramienta.

3.3.1 Principales Características y Criterios de Evaluación

Basándose en las funcionalidades de cada una de las herramientas y las entrevistas realizadas a las empresas desarrolladoras de software GOOTLEK, IOET, PRENDHO y UTI de la ciudad de Loja (véase Anexos 2, 3, 4 y 5), se pudo obtener la información necesaria para establecer los criterios de evaluación, mismos que permitirán realizar la selección de la herramienta más idónea que indicará el cumplimiento del último objetivo de este Proyecto de Titulación.

A continuación se presenta el listado de herramientas donde cada una consta de sus funcionalidades; según las entrevistas realizadas se procedió a contrastar la información y seleccionar aquellas que cumplan con la mayor parte de requerimientos previos a la selección de una herramienta que permita gestionar el ciclo de vida de las aplicaciones a desarrollarse. Como se puede observar hay ciertos requerimientos que ninguna de las herramientas posee, en este caso se puede mencionar la gestión de pruebas unitarias, y algunas más, que aunque no sean numerosas forman parte importante de los requerimientos previos considerados por los grupos de desarrollo, pero antes de establecer la inexistencia de estos en las herramientas a evaluar, se las aplicará a un caso de estudio, el cual permitirá constatar su existencia o ausencia, así mismo encontrar funcionalidades no mencionadas que puedan ser valiosas o de gran ayuda al gestionar el ciclo de vida de una aplicación.

3.3.1.1 Tuleap

Funcionalidad principal

- Gestión de proyectos
- Tareas
- Cambios
- Defectos
- Documentos

- Control de versiones
- Integración continua (herramientas de código abierto)
- Colaboración social
- Alta capacidad de personalización
- Adaptabilidad a procesos y requerimientos
- Sirve para un número ilimitado de usuarios, tiempo y proyectos.
- Proyectos del plan y el monitor
- Gestión del ciclo de vida de desarrollo de software: código fuente, construye ...,
- Requisitos de la pista, tareas, incidencias, etc.
- Producir documentos y comunicados,
- Favorecer la colaboración entre los miembros del proyecto
- Gestión de todo el ciclo de vida de los proyectos de software
- Manejo de diferentes metodologías de software
- Personalizar las herramientas a necesidades específicas
- Recoger y clasificar los requisitos de los clientes y las necesidades
- Planear lanzamientos o sprints
- Asignar tareas a los miembros del proyecto
- Seguimiento de avances y trabajo restante
- Estado en tiempo real sobre los proyectos
- Gráficos e informes sobre la marcha

3.3.1.2 Mylyn

Funcionalidad principal

- Interfaz enfocada a objetivos
- Gestión de tareas
- Amplio ecosistema de Agile ALM
- Interacción con decenas de extensiones con herramientas de colaboración de ALM y desarrolladores
- Reduce la sobrecarga de información
- Multitarea fácil
- Información accesible y al alcance
- Marcos y APIs para tareas basadas en Eclipse y ALM

- Herramientas ejemplares para la programación
- Implementaciones de referencia para las herramientas de ALM y estándares abiertos
- Estructuración en subproyectos
- Tareas
- Integra la edición rica y fuera de línea para los repositorios
- Facilita la multitarea, la planificación, la reutilización de los esfuerzos del pasado y compartir experiencia

3.3.1.3 Endeavour Agile

Funcionalidad principal

- Agile Project Management
- Presupuesto
- Colaboración
- Seguimiento de problemas
- Aprendizaje y apoyo
- Notificaciones
- Informes
- Gestión de recursos
- Gestión de tareas
- Gestión de documentos
- Gestión de proyectos tradicional
- Interfaz gráfica con apariencia similar a las aplicaciones de escritorio tradicionales
- Fácil uso y adaptabilidad
- Glosario del proyecto
- Integración continua
- Asignaciones de miembros del proyecto
- Seguimiento de defectos
- Basado en la nube
- Construido para empresas de tamaño pequeño a medio
- Se ejecuta en los principales navegadores (Mozilla Firefox, Internet Explorer, Apple Safari, Google Chrome, Opera)

3.3.1.4 Hansoft

Funcionalidad principal

- Colaboración en equipo
- Gestión en el desarrollo de software ágil de productos y servicios
- Gestión de programas
- Gestión de documentos
- Gestión de Empresa Portafolio
- Gestión de proyectos
- Gestión de la Cartera de Proyectos
- Gestión de métricas
- Gestión de flujo de trabajo
- Gestión de fallos
- Gestión de problemas
- Gestión de riesgos
- Gestión de calidad
- Gestión de requisitos
- Gestión de procesos
- Cartera de productos
- Colaboración Social
- Ejecución de tareas, metodologías y proyectos en paralelo
- Facilita la transferencia de datos
- Interfaz intuitiva que simplifica la complejidad del trabajo
- Informes de cambios
- Completo sistema de chat
- Métodos Side-by-Side
- Gran Escala
- Métricas accionables
- Mejora continua
- Seguridad a nivel de empresa
- Construido para la velocidad (tiempo real)
- Actualizaciones y soporte incluido
- Ágil

- Seguimiento de problemas
- Seguimiento de asignación
- Personalización de plantillas de proyecto
- Administración de personal, tareas y recursos
- Se ejecuta plataformas como: Linux, Mac y Windows

3.3.1.5 Team Foundation Service

Funcionalidad principal

- ALM de aplicaciones TFS bajo la modalidad de SaaS
- Actualizaciones gratuitas e inmediatas
- Compatible con los navegadores más modernos
- Plantillas de procesos: Scrum, Agile o CMMI.
- Gestión del equipo y su trabajo
- Gestión del backlog
- Sprint actual
- Gestión de los elementos de trabajo
- Control de código fuente
- Automatización de builds, pruebas y despliegues
- Control de versiones
- Seguimiento del trabajo
- Integración continua
- Lenguaje y herramientas

3.3.1.6 Star Team Express

Funcionalidad principal

- Cambios a través de cualquier herramienta y repositorio
- Gestión de configuración de software
- Mantiene la visibilidad y trazabilidad en todos sus activos de software
- Colaboración y comunicación entre el equipo
- Centralización de las actividades del proyecto y recursos digitales
- Informes de cambios
- Versiones de archivos

- Seguimiento de cambios
- Flujo de trabajo personalizable
- Liberar y Construcción de Gestión
- End-to-end análisis de impacto
- Apoyo a las decisiones
- Generación de informes de liberación
- Visibilidad constante en todos los artefactos de liberación
- Análisis y evaluación de tendencias
- Facilita la realización de cambios
- Para grupos de desarrollo de hasta 10 usuarios

3.3.1.7 Topcased o Polaris

Funcionalidad principal

- Gestión de la calidad
- Gestión de madurez
- Componentes de prototipos
- Documentos y kits de cualificación
- Gestión de configuración
- Gestión de cambios
- Gestión de requisitos
- Seguimiento de problemas
- Informes
- Simulación y lógica de hardware

3.4 Criterios a Tomar en Cuenta en el Proceso de Selección.

Existen varios criterios que deben considerarse al momento de elegir entre herramientas ALM, ya que estas se encargan de brindar mayor productividad y soluciones, no sólo al servicio de los desarrolladores, sino de toda una compañía.

En esta comparativa no se va a considerar el costo, ya que se realizará en base a herramientas libres. Se tomarán en cuenta criterios importantes a la hora de decidirse en la utilización de una herramienta ALM considerando estas para una predominante eficiencia, productividad y calidad del software.

Criterios

Conjuntamente con los criterios se citará 3 características que formarán parte de la evaluación, cuyo puntaje será proporcional al cumplimiento de las mismas.

- **Calidad:** grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. En donde es muy importante considerar las opiniones de la comunidad de desarrolladores y se evaluará tomando en cuenta: el tipo de empresas al que está enfocada la herramienta, cumplimiento de necesidades de los clientes y conjunto de características para la gestión.
- **Funcionalidad:** conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas, donde se evaluará tomando en cuenta: monitorización de los avances del proyecto, seguimiento de bugs, tareas y proyectos y uso de metodologías ágiles.
- **Usabilidad:** conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, aquí se evaluará: interfaz amigable al usuario, presentación de forma ordenada y conocimientos del usuario para el uso de la herramienta (con o sin conocimientos técnicos). [42]
- **Seguridad:** controles administrativos que se establecen de acceso al sistema y las regulaciones legales que afecta o determina el uso del sistema y que serán tenidas en cuenta si se incumple, donde se evaluará: login para el acceso a la herramienta, diferentes niveles de acceso y visibilidad. [43]
- **Parametrización:** configurar el sistema de una manera específica o en aspectos puntuales de su funcionamiento, teniendo en cuenta los parámetros de seguridad, dándole los permisos necesarios, pero sin exceso de los mismos y realizar todas las actividades necesarias para que quede funcionando de forma adecuada, adaptándose así a las necesidades de la empresa. La parametrización exige un gran conocimiento tanto del producto como de las necesidades de la empresa y requiere ser evaluado mediante su forma de: ajustarse a las necesidades del proyecto (rastreadores), uso de plantillas y libre configuración de características (metodologías, gráficos, comunicación entre miembros, reportes).
- **Soporte:** brinda al usuario la facilidad de instalación, facilidad de mantenimiento, lo que requiere código y diseño documentado y facilidad de actualización hacia

versiones más modernas. Aquí se evaluará: manual o guía de usuario, seguimiento de fallos y soporte profesional (contacto a través de correos). [44]

- **Licencias:** contrato entre el autor del programa y el usuario, que comprenden una serie de términos y cláusulas que el usuario deberá cumplir para usar el mismo. En los programas gratuitos, las condiciones siempre están a favor del usuario final, donde se evaluará el tipo de licencia e información sobre la misma.
- **Versionamiento:** registra todos los cambios hechos en uno o más proyectos, guardando así versiones del producto en todas sus fases del desarrollo. Las versiones son como fotografías que registran su estado en ese momento del tiempo y se van guardando a medida que se hacen modificaciones al código fuente. De este factor se evaluará únicamente si la herramienta proporciona o no el versionamiento. [45]
- **Uso de metodologías ágiles:** método que permite incorporar cambios con rapidez en el desarrollo de software, donde se evaluará: uso de plantillas ágiles, selección de diferentes metodologías y configuración de nuevas metodologías. [46]
- **Instalación:** proceso por medio del cual una aplicación pasa a formar parte del conjunto de programas listos para ser ejecutados en el ordenador. [47] considerando adicionalmente la arquitectura de la aplicación a instalar. En este último factor se evaluará únicamente la facilidad de instalación de la herramienta.

4. CAPÍTULO IV: Análisis de las Herramientas a Utilizar

4.1 Pentaho.

Herramienta de Business Intelligence (permite crear conocimiento a partir de la información de la empresa) desarrollada bajo la filosofía del software libre para la gestión y toma de decisiones empresariales. Ofreciendo soluciones para la gestión y análisis de la información, incluyendo el análisis multidimensional OLAP (procesamiento analítico en línea), presentación de informes, minería de datos y creación de cuadros de mando para el usuario.

- **Pentaho Analysis:** Ayuda a operar con máxima efectividad para ganar perspicacia y entender lo necesario para tomar optimas decisiones. Permite, de una forma interactiva, analizar los datos del Data Warehouse a través de una

interfaz de tabla cruzada donde se puede navegar por las diferentes dimensiones definidas en el modelo dimensional. [48]

Está basado en Mondrian, que es el corazón de este (servidor de código abierto OLAP que responde a las preguntas lo suficientemente rápido como para permitir una exploración interactiva de los datos). [5] Y en Jpivot, que es la herramienta de análisis de usuario, con el que se realiza la navegación dimensional sobre los cubos desde la plataforma BI y son visualizados los resultados de las consultas. Estas son ejecutadas por Mondrian, que traduce los resultados relacionales a resultados dimensionales, que a su vez son mostrados al usuario en formato Html por Jpivot.

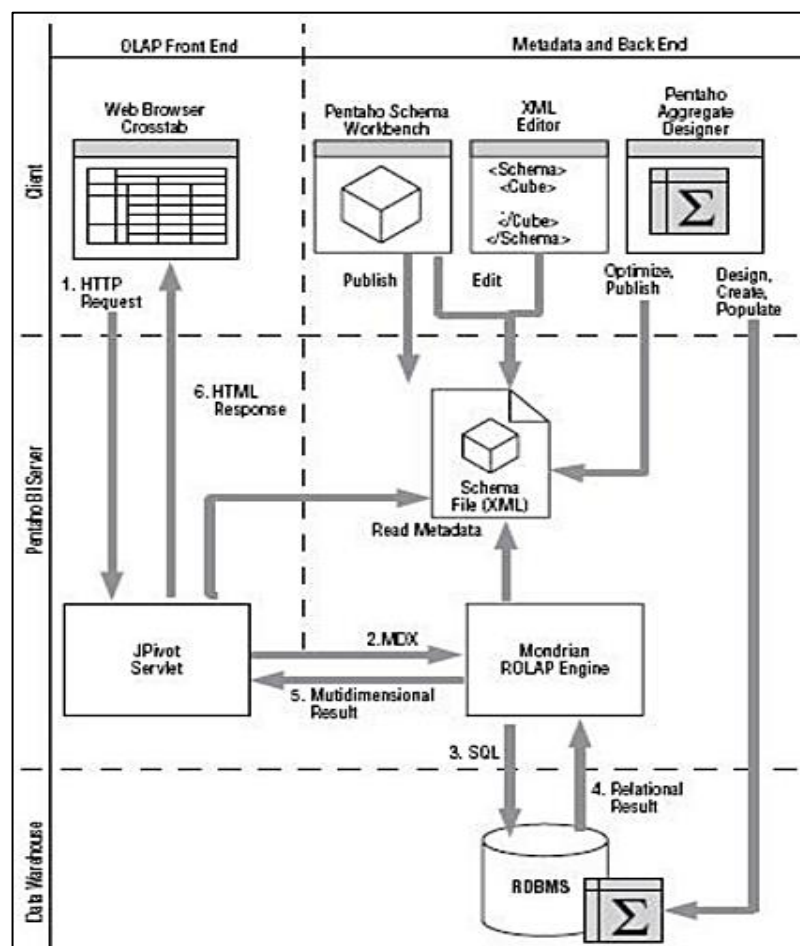


Figura 3: Arquitectura Pentaho Analysis. [49]

- **Pentaho Dashboards:** Provee inmediata perspicacia en un rendimiento individual, departamental o empresarial.
- **Pentaho Data Integration:** permite una poderosa ETL (Extracción, Transformación y Carga) de información.

- **Data Mining:** Proceso de correr datos en algoritmos completamente sofisticados, relevando significantes patrones y correlaciones que pueden estar escondidos.
- **Pentaho BI Plataforma:** Provee servicios críticos incluyendo programación, seguridad, integración automatización y flujo de trabajo. [49]

4.2 Metodologías Tradicionales y Web.

Tradicionalmente los aplicativos de software eran construidos con la finalidad de que sean instalados en el computador del cliente, y así estos puedan funcionar de forma independiente. Este modelo de distribución se acentuó a finales de los 70 debido a la popularización de las computadoras personales. [50]

Pero con el avance tecnológico surgieron las aplicaciones web, mismas que gracias al lenguaje PHP dieron inicio. [51] El auge de estas aplicaciones se debe a las ventajas que ofrece sobre las aplicaciones tradicionales, que son:

- Capacidad de despliegue rápido.
- Acceso a los datos desde cualquier lugar.
- Se obtiene siempre la última versión de la aplicación.
- Cero instalación. Solo se requiere un navegador web para acceder.
- Independencia de la plataforma.
- Fácilmente adaptables.
- Alcance global (requiere publicación en internet).
- Entorno familiar para los usuarios.
- Gran importancia al contenido y aspecto visual. [50]

En base a esta información se puede apreciar que muchas metodologías tradicionales (para el desarrollo de aplicaciones) no consideran la adaptabilidad como una prioridad dentro de su proceso y potencian la planificación detallada de prácticamente todo el desarrollo software. [52] La constante innovación tecnológica hace que cada vez sea necesaria la aplicación de nuevas metodologías adaptadas a los nuevos tiempos y, sin embargo, siguen figurando en los libros de texto viejas metodologías pensadas para viejos problemas. [53]

Entre estas tenemos una lista de las metodologías más utilizadas a nivel mundial en orden cronológico:

Década de los 70s

- En Cascada desde 1970
- Programación Estructurada Jackson desde 1975

Década de los 80s

- Structured Systems Analysis and Design Methodology (SSADM) desde 1980
- Structured Analysis and Design Technique (SADT) desde 1980
- Scrum a principios de los 80
- Ingeniería de la Información (IE/IEM) desde 1981

Década de los 90s

- Rapid Application Development (RAD) desde 1991
- Programación Orientada a Objetos (OOP) a lo largo de la década de los 90's
- Virtual Finite State Machine (VFSM) desde 1990s
- Dynamic Systems Development Method desarrollado en UK desde 1995
- Iconix desde 1993
- Rational Unified Process (RUP) desde 1999

Año 2000 en adelante

- Extreme Programming (XP) desde 2000
- GAIA desde 2000
- UWE desde 2000
- Enterprise Unified Process (EUP) extensiones RUP desde 2002
- Constructionist Design Methodology (CDM) desde 2004 por Kristinn R. Thórisson
- Mobile D desde 2004
- Agile Unified Process (AUP) desde 2005 por Scott Ambler [54]

Es por esta razón que surgen metodologías que se adaptan a las nuevas necesidades y en especial al desarrollo de aplicaciones web (EORM, OOHDM, SOHDM, WSDM RNA, UWE); [55] quedando así desplazadas de cierta forma las metodologías tradicionales. Que si bien es cierto siguen siendo utilizadas (por lo general en aplicaciones de escritorio) pero con una frecuencia menor con el paso del tiempo.

e. METODOLOGÍA

La aplicación de metodologías constituye una parte fundamental en el estudio que se realizará para responder al problema planteado, garantizando la validez de las afirmaciones que se realicen. Es así como en el presente Proyecto de Titulación se utilizarán los siguientes métodos y técnicas:

MÉTODOS

Método Deductivo: permite conocer acerca de la gestión de ciclo de vida de una aplicación de forma general, hasta el tipo de herramientas que ayudan a la misma; para que a través del razonamiento establecer cuál de ellas es la más adecuada para el empleo de los estudiantes.

Método Descriptivo: permite describir y evaluar las características de las herramientas open source ALM, con la finalidad de facilitar el proceso de escogitamiento.

Método Analítico: que apoya a la revisión de cada una de las herramientas objeto de análisis en el presente proyecto.

Método Comparativo: permite precisar “el mejor” entre las alternativas estudiadas, en este caso el conjunto de herramientas ALM.

TÉCNICAS

Observación: necesaria para poder obtener información de forma general o superficial acerca de los problemas que rodean al desarrollo de proyectos software. Así como determinar las características más relevantes de las herramientas y su interacción con el usuario.

Encuesta: sirve para poder ratificar o validar la situación problemática y poder adentrarse más al establecimiento del problema central de investigación. Aplicada en forma general con la finalidad de obtener problemas globales como lo es el desconocimiento de las Herramientas ALM.

Entrevista: permite validar el problema de investigación, por lo general es aplicada a personas específicas; en este caso a los representantes de las empresas de desarrollo

de software de la ciudad de Loja, ya que su experiencia en el área permitió establecer los criterios de evaluación de las herramientas.

Investigación Bibliográfica: sirve para la recopilación información relevante y específica de un tema en general, en este caso las herramientas ALM tanto libres como privadas. Cuyo objetivo es adentrarse al mundo que rodea la solución del problema, accediendo a fuentes fidedignas que realcen el desarrollo del Proyecto de Titulación.

Demostrativa: permitió mostrar cómo debería realizarse el correcto uso de la herramienta ALM resultante a través del modelamiento, la observación y la demostración.

f. RESULTADOS

1. Criterios de evaluación.

Los criterios de evaluación de las *Tablas 3 y 4* se obtuvieron en base a la información obtenida de las entrevistas a empresas de la ciudad (véase Anexo 2, 3, 4 y 5) y fueron contrastadas con las características de cada una de las herramientas ALM privadas y libres (véase Capítulo III – Sección 3.3.1). Con la finalidad de conocer cuál de ellas cumple con la mayor parte de los criterios establecidos.

Tabla III: Contraste entre los Criterios de Evaluación y las Herramientas ALM Privadas.

Requisitos	Criterios de Evaluación							
	Herramientas							
	Atlassian JIRA	IBM Rational solution	IBM Rational Team Concert	HP	Star Team	SAP Solution Manager	Trace Cloud ALM	Tomos
Tipo distribuida			✓		✓		✓	✓
Servicios en la nube						✓	✓	✓
Control de versiones	✓	✓	✓	✓	✓	✓	✓	✓
Unificar el trabajo								
Notificaciones					✓			
Gestión de tareas	✓	✓	✓	✓	✓	✓	✓	✓
Gestión de proyectos	✓	✓	✓	✓	✓	✓	✓	✓
Gestión de procesos			✓	✓	✓	✓		✓
Gestión de productos								
Gestión de requisitos	✓	✓	✓		✓	✓	✓	✓
Gestión de riesgos	✓							✓
Gestión de tiempos	✓	✓		✓				✓
Gestión de costos								✓
Gestión de código	✓	✓	✓	✓	✓	✓	✓	✓
Colaborativa		✓	✓	✓		✓		✓
Social								✓
Aseguramiento de calidad		✓		✓		✓	✓	✓
Control de pruebas							✓	

Control de diagramas	✓	✓	✓		✓		✓	✓
Multiusuario								
Multitarea	✓	✓	✓	✓	✓	✓	✓	✓
A nivel de servicio							✓	
Ingeniería de software			✓	✓			✓	✓
Uso de metodologías ágiles			✓	✓			✓	✓
Estadísticas del proyecto	✓	✓	✓		✓		✓	✓
Adaptabilidad		✓	✓	✓	✓	✓	✓	✓
Uso de plantillas			✓	✓			✓	✓
Interacción con el usuario								

La *Figura 4* muestra el cumplimiento de los requerimientos de cada una de las herramientas, permitiendo identificar visualmente los resultados.

- **Atlassian JIRA**, con 10 características cumplidas.
- **IBM Rational solution**, con 12 características cumplidas.
- **IBM Rational Team Concert**, con 8 características cumplidas.
- **HP**, con 13 características cumplidas.
- **Star Team**, con 12 características cumplidas.
- **SAP Solution Manager**, con 11 características cumplidas.
- **TraceCloud ALM**, con 17 características cumplidas.
- **Tomos**, con 21 características cumplidas

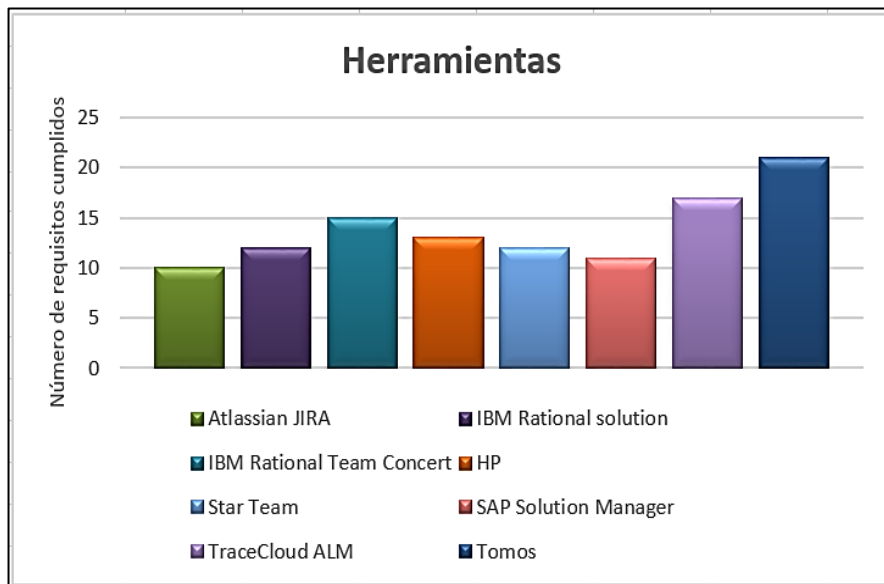


Figura 4: Resultados estadísticos del cumplimiento de requerimientos de Herramientas Privadas.

Como se puede apreciar la herramienta ALM privada que más requerimientos cumple, en base a sus características es:

- **Tomos**

Tabla IV: Contraste entre los Criterios de Evaluación y las Herramientas ALM Libres.

Requisitos	Criterios de Evaluación						
	Herramientas						
	Tuleap	Mylyn	Endeavour Agile	Hansoft	Team Foundation Service	Star Team Express	Topcased
Tipo distribuida							
Servicios en la nube			✓				
Control de versiones	✓	✓		✓	✓	✓	✓
Unificar el trabajo							
Notificaciones			✓				
Gestión de tareas	✓	✓	✓	✓	✓		
Gestión de proyectos	✓	✓	✓	✓			
Gestión de procesos				✓			

Gestión de productos				✓			
Gestión de requisitos	✓			✓			✓
Gestión de riesgos				✓			
Gestión de tiempos							
Gestión de costos			✓				
Gestión de código	✓				✓		
Colaborativa	✓	✓	✓	✓		✓	
Social	✓		✓	✓		✓	
Aseguramiento de calidad				✓			✓
Control de pruebas					✓		
Control de diagramas							
Multiusuario							
Multitarea		✓					
A nivel de servicio							
Ingeniería de software							
Uso de metodologías ágiles	✓			✓	✓		
Estadísticas del proyecto	✓						
Adaptabilidad	✓	✓	✓	✓	✓	✓	✓
Uso de plantillas				✓	✓		
Interacción con el usuario							

La *Figura 5* muestra el cumplimiento de los requerimientos de cada una de las herramientas, permitiendo identificar visualmente los resultados.

- **Tuleap**, con 10 características cumplidas.
- **Mylyn**, con 6 características cumplidas.
- **Endeavour Agile**, con 8 características cumplidas.
- **Hansoft**, con 13 características cumplidas.
- **Team Foundation Service**, con 7 características cumplidas.
- **Star Team Express**, con 4 características cumplidas.
- **Topcased**, con 4 características cumplidas.

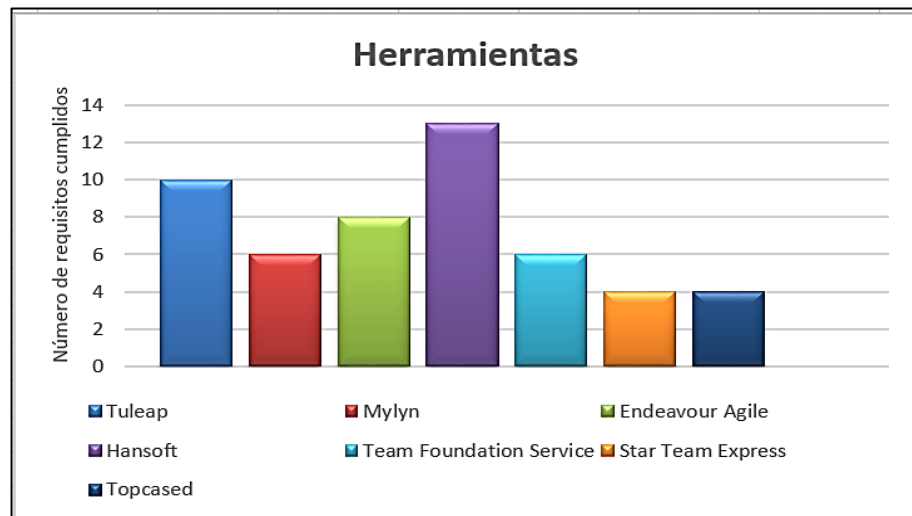


Figura 5: Resultados estadísticos del cumplimiento de requerimientos de Herramientas Libres. Como se puede apreciar las herramientas ALM libres que más requerimientos cumplen, en base a sus características son:

- **Hansoft**
- **Tuleap**
- **Endeavour Agile**

Mismas que serán empleadas en un caso de estudio en particular y poder establecer con cuál de estas se obtiene mejores resultados al momento de Gestionar el ciclo de vida de la aplicación o caso de estudio.

2. Características del escenario para la Selección de la Herramienta.

Las características del escenario se han establecido mediante la técnica de observación, basándose en los datos que requieren cada una de las herramientas para iniciar con la gestión del ciclo de vida de una aplicación. Primeramente se analizó Endeavour, para usar esta herramienta se necesita tener establecido los casos de uso, las fases o iteraciones, tareas y finalmente los bugs o errores que surgirán en el desarrollo del software. Seguidamente se analizó Hansoft, esta herramienta requiere tener establecido las fases o iteraciones, submódulos, metodologías, backlog o requerimientos y por último los bugs o errores que surgirán en el desarrollo del software. Finalmente se analizó Tuleap, misma que requiere sprints, tareas, historias de usuario, y bugs, cabe resaltar que todos ellos necesitan de un programador, administrador y tester.

Grupo de trabajo

El equipo debe estar conformado mínimo por:

- Programador
- Administrador
- Tester

Avance del proyecto

Debe estar definido o establecido lo siguiente:

- Backlog o requerimientos
- Casos de uso o historias de usuario
- Fases o iteraciones

Estructura del proyecto

- Metodología
- Iteraciones
- Submódulos (opcional)

2.1. Casos de Estudio

Los casos de estudio que se presentan a continuación han sido seleccionados basándose en los proyectos de titulación que se han desarrollado el ciclo pasado en la Escuela de Ingeniería en Sistemas de la Universidad Nacional de Loja.

Para poder establecer el caso de estudio se utilizó la técnica de Observación Estructurada en la cual se tiene que delimitar que aspectos se va a observar y que tipo de información se requiere. Para lo cual se tiene una lista de aspectos a observar:

- Año de aprobación de la tesis (2015)
- Tema enfocado al desarrollo de software
- Bugs
- Características de escenario (detalladas)

Una vez establecida la lista se ingresó al Repositorio Digital de la universidad y se buscó los temas de tesis relacionados con desarrollo de software dentro del año 2015. De estas se seleccionó 5 en base al tema de las mismas.

1. Creación e implantación de una herramienta web para la supervisión control y monitoreo de presupuestos en los proyectos impulsados por el Ministerio de Inclusión Económica y Social MIES de Zamora Chinchipe. (Iconix)
2. Diseño y construcción de un Kiosko electrónico prototipado para la impresión automática de certificados académicos para los estudiantes del Área de la Energía, las Industrias y los Recursos Naturales no Renovables. (Iconix)
3. Software inteligente para el procesamiento musical de guitarra melódica aplicando una red neuronal supervisada. (XP)
4. Desarrollo del portal web de la Universidad Nacional de Loja aplicando técnicas de web semántica en el gestor de contenidos DRUPAL. (RAD)
5. Realidad aumentada para la visualización de la infraestructura interna y externa de edificios de la Universidad Nacional de Loja a través de entornos móviles. (Mobile D)

Estos proyectos fueron ubicados en la *Tabla 5* con la finalidad de establecer cuáles de ellos cumplen con las características del escenario ya establecidas. Y de esta manera obtener el caso de estudio previo a la selección de la herramienta.

Tabla V: Contraste de Información entre los Proyectos de Titulación y las Características del Escenario.

Caso de Estudio					
Características del Escenario	Proyectos				
	Proyecto 1	Proyecto 2	Proyecto 3	Proyecto 4	Proyecto 5
Programador	✓	✓	✓	✓	✓
Administrador	✓	✓	✓	✓	✓
Tester	✓	✓	✓	✓	✓
Requisitos o Backlog	✓	✓			✓
Casos de uso o Historias de usuario	✓	✓	✓		
Fases o iteraciones	✓	✓	✓	✓	✓
Tareas	✓	✓	✓	✓	✓
Metodología	✓	✓	✓	✓	✓
Bugs					

2.2. Selección de Metodología de Prueba

La metodología Iconix ha sido seleccionada para el caso de estudio debido a las siguientes características:

- **Iterativo e incremental:** existen iteraciones entre el desarrollo de modelo del dominio y la identificación de los casos de uso que intervienen en el negocio.
- **Trazabilidad:** Cada paso está referenciado por algún requisito. Siguiendo una relación entre los diferentes artefactos del producto.
- **Dinámica del UML:** Uso dinámico de UML en los diagramas de caso de uso, diagramas de secuencia y de colaboración. [6]

De estos casos de estudio dos utilizaron la metodología Iconix, y los 3 restantes XP, RAD y Mobile D respectivamente como lo muestra la *Figura 6*. Lo que nos indica que el caso de estudio a seleccionar será uno de aquellos que utilizó Iconix, debido a su empleo en la mayoría de proyectos.

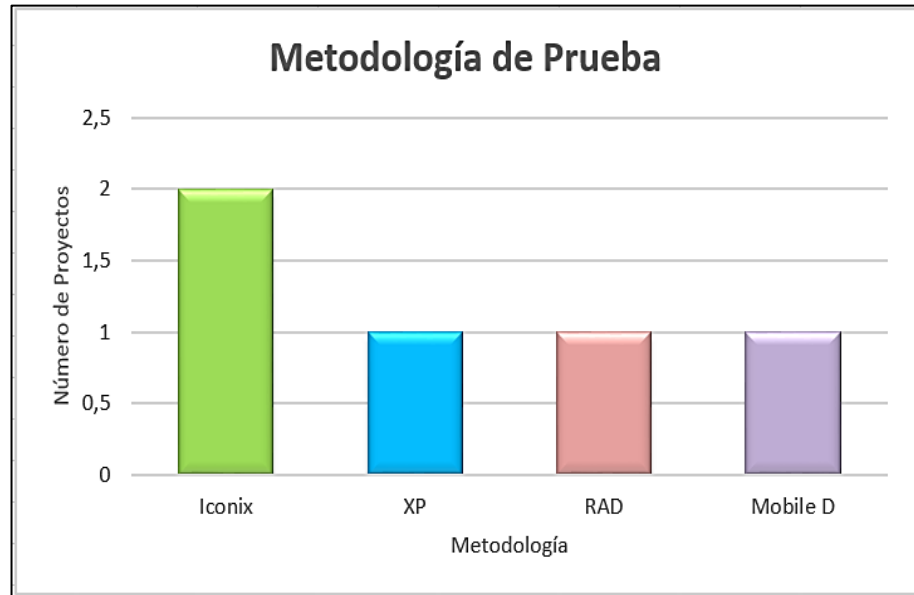


Figura 6: Metodología de prueba resultante.

3. Análisis de las Herramientas.

En base a los criterios a comparar (véase Capítulo III – Sección 3.4) se realizó la siguiente evaluación de las tres herramientas:

3.1. Calidad

- Endeavour siendo una herramienta enfocada a pequeñas y medianas empresas cumple con las expectativas de calidad, brindando satisfacción al cliente, supliendo sus necesidades. Al estar enfocado a este tipo de empresas por lo general no requieren de la realización de gestiones detalladas, siéndoles de gran utilidad únicamente el llevar registro de los casos de uso, cambio de requerimientos, bugs, tareas y asignaciones de las mismas, planes de pruebas, añadir documentos al proyecto, crear una wiki, realizar la integración con subversión o la visualización de reportes, estadísticas y proyectos. Características que satisfacen las necesidades de gestión del ciclo de vida de sus proyectos.
- Hansoft al estar enfocado o dirigido a todo tipo de empresas, también ha sabido cumplir y suplir con las necesidades de sus clientes. Siendo una herramienta mucho más completa ofrece la selección de la metodología de programación, releases, producto backlog, iteraciones, tareas, subtareas, tareas programadas y asignación de las mismas, creación de reportes, historias de usuario, bugs, workflows, diagramas, hyperlinks, chat de grupo y estadísticas del proyecto o proyectos que se tienen. Ofreciendo un control y completa visibilidad de los mismos incluyendo a los miembros del proyecto y sus avances.
- Tuleap está enfocado a todo tipo de empresa; puede construir, implementar proyectos inteligentes y de calidad, gracias a sus características, cubriendo de igual manera las necesidades de sus clientes. Esta herramienta también es muy completa, permite la creación de proyectos mediante una plantilla ágil, creación de sprints, tareas, historias de usuario, bugs, releases, epics o personalizar su propio tracker, visualización de estadísticas o creación de diagramas, añadir documentos al proyecto, wikis, listas, foros, notificaciones, integración con Git y Subversion.

3.2. Funcionalidad

- Endeavour proporciona soporte a varias características, artefactos de ciclo de vida (especialmente el seguimiento de problemas y tareas) y los informes.

- Hansoft tiene un conjunto de características que abarca más ágil ALM, permite ejecutar tareas, metodologías y proyectos en paralelo, además posee un completo sistema de chat para la comunicación instantánea con los miembros del equipo.
- Tuleap permite monitorizar el avance del proyecto, gestionar el ciclo de vida de desarrollo de software y manejar diferentes metodologías de software favoreciendo la colaboración.

3.3. Usabilidad

- Endeavour proporciona una interfaz gráfica de usuario con una apariencia similar a las aplicaciones de escritorio tradicionales, facilitando de esta manera su uso y adaptabilidad del usuario.
- Hansoft posee una interfaz intuitiva que simplifica la complejidad del trabajo para el equipo, ajustando los procesos a las necesidades e informando de los cambios a cada uno de los miembros, su uso requiere de personal con conocimientos técnicos.
- Tuleap posee una interfaz intuitiva y alta capacidad de personalización, permitiendo al usuario adaptar cada herramienta a sus procesos y requerimientos del negocio.

3.4. Seguridad

- Endeavour posee diferentes niveles de acceso.
- Hansoft posee diferentes niveles de acceso, control de visibilidad y el protocolo de cifrado TLS 1.2 estándar entre servidores y clientes.
- Tuleap posee diferentes niveles de acceso y control de visibilidad.

3.5. Parametrización

- Endeavour no tiene opciones de personalización.
- Hansoft posee una capacidad de personalización no muy alta.
- Tuleap posee alta capacidad de personalización, permitiendo al usuario adaptar cada herramienta a sus procesos y requerimientos del negocio.

3.6. Soporte

- Endeavour en su página oficial ofrece:
 - “Encontrar Ayuda en nuestros foros”

- Posee foros Endeavour proyecto ALM, para facilitar el uso de la herramienta en donde se realizan preguntas según una clasificación determinada.
- “Consulta también nuestra Wiki”
- Endeavour ALM Wiki del Proyecto, que ofrece información acerca de la herramienta
- “Informe de problemas en nuestro seguimiento de fallos”
- Endeavour ALM Bug Tracker, donde se ingresan los defectos y se realiza el seguimiento de los mismos
- “Se ponen en contacto con nosotros por e-mail”
- Permite contactarse a través del correo electrónico
- “Consulta nuestra opiniones de los usuarios”
- Endeavour ALM Los comentarios de los usuarios, en donde se visualiza las valoraciones y opiniones acerca de la herramienta [56]
- Hansoft en su interfaz posee la sección “Contact Support” que ofrece:
 - Manual de usuario Hansoft, posee una guía sobre instalación, administración, tutoriales, documentos, entre otras opciones.
 - Guía del administrador del sistema, dirigida a las personas responsables de la instalación, configuración y mantenimiento de servidores y clientes Hansoft.
 - SDK y las integraciones, contiene la puesta en marcha y guías de administración para las integraciones de terceros ya hechos de Hansoft.
 - Hansoft guía de gestión de acciones, como su nombre lo dice contiene la guía de gestión de acciones y la migración a acciones.
 - Bienvenido a la prueba Hansoft, es un paquete que contiene el servidor Hansoft, una licencia libre para nueve usuarios y el cliente Hansoft. El servidor, que es alojado en sí mismo, contiene datos de ejemplo de proyecto y la información de licencia.
 - Cómo, contiene artículos sobre cómo sacar el máximo provecho de Hansoft.
 - Se puede contactar con Hansoft support a través de support@hansoft.com [57]
- Tuleap en su página oficial ofrece:
 - CONSULTE LA DOCUMENTACIÓN PRIMERA, que recomienda echar un vistazo a la guía para el uso de Tuleap de la mejor manera.
 - SOPORTE COMUNITARIO, para lo que se requiere crear una nueva cuenta o entrada a su cuenta en el espacio de trabajo del proyecto Tuleap y enviar su pregunta a la lista de correo de la Comunidad

- SOPORTE PROFESIONAL, que trabaja con Enalean (principal compañía de desarrollador de software de Tuleap) y esta ofrece:
 - Tuleap Empresa Planes para espacios de trabajo de la empresa: más estabilidad, más seguridad, apoyo de respuesta.
 - Entrenamientos: ¿está seguro que ahorrar tiempo a aprender por sí mismo?
 - Plan de trabajo abierto: lo que los usuarios necesitan próxima en Tuleap? [58]

3.7. Licencias

- Endeavour liberado bajo la versión Licencia Pública General de GNU 3. Esta es la última versión de la GPL de GNU: una licencia de software libre con copyleft. Recomendada para la mayoría de los paquetes de software. [59] Es una de las licencias más importantes dentro del software libre y la que rige software como el núcleo de Linux. Esta garantiza una serie de libertades al usuario (libre uso y adaptación del código fuente del programa), obligando a que las modificaciones que se realicen en el código de un programa estén disponibles para el resto de usuarios si este se redistribuye. Además intenta solucionar los problemas de patentes de software (ofrece permiso gratuito para utilizar las patentes que lo cubren al resto de usuarios) y tivoización (evitar que el usuario no pueda modificar el software y ejecutarlo a causa del DRM que se utiliza). [60]
- Hansoft no ofrece información sobre su licencia.
- Tuleap se proporciona bajo la licencia GPLv2. Que es la versión previa de la GPL de GNU: una licencia de software libre con copyleft. [59] Esta pretende garantizar la libertad de compartir y modificar software libre, para asegurar que el software es libre para todos sus usuarios. Se aplica a la mayor parte del software de la Free Software Foundation y a cualquier otro programa si sus autores se comprometen a utilizarla. [61]

3.8. Versionamiento

- Endeavour proporciona este factor a través de la Integración continua con Subversion.
- Hansoft en su versión libre no permite el uso de versionamiento.
- Tuleap permite realizar la Integración continua con Git y Subversion.

3.9. Uso de metodologías ágiles

- Endeavour no proporciona las opciones para seleccionar o establecer una metodología ágil en particular y poder trabajar en base a una plantilla; pero al no constar de esta característica se adapta sin ningún problema a la metodología de trabajo de la empresa, al permitir el ingreso de iteraciones y tareas.
- Hansoft si proporciona las opciones de selección de metodologías ágiles y poder trabajar en base a una plantilla específica para cada una de estas (Scrum, XP e Iconix).
- Tuleap no permite realizar la selección de metodología a utilizar, ya que por defecto provee dos plantillas y en cualquiera de estas el trabajo se ve limitado al uso de XP como metodología de desarrollo y Scrum como marco de trabajo.

3.10. Instalación

- Endeavour no presenta problemas en su instalación, ya que este funciona por medio del navegador, inicializando la aplicación en el servidor de Tomcat.
- Hansoft en su versión para Windows se instala sin problemas, únicamente descargándolo de su página oficial, y de igual manera en Linux no hay dificultades.
- Tuleap presenta problemas al realizar la instalación, ya que solicita paquetes o archivos que no se encuentran disponibles, por esta razón lo más recomendable es emplear su versión Tuleap-campus.

3.11. Valoraciones

Para establecer las valoraciones se usará el Método Comparativo que, al igual que cualquier otro método de análisis empírico requiere, una serie de decisiones previas referidas al diseño de investigación. [62] Donde básicamente uno de los criterios principales son evaluativo como la "satisfacción", la "utilidad" etc., y la puntería del estudio es precisar el mejor (en este respecto) entre las alternativas que se estudian. [63]

Mediante el método comparativo como se puede apreciar en la *Tabla 6*, se establece una puntuación de 1 al 3, en donde 3 será la máxima valoración, mismos que se asignarán de acuerdo a la información de cada herramienta.

Tabla VI: Puntuación de cada uno de los niveles de evaluación.

Nivel	Puntuación
Alta	3
Media	2
Baja	1

La *Tabla 7* contiene la valoración de las herramientas, realizada en base a los criterios establecidos. De la cual se obtendrá la herramienta con mayor puntuación.

Tabla VII: Valoración de cada una de las Herramientas en base al cumplimiento de los criterios establecidos.

Cuadro comparativo						
Criterios	Herramientas					
	Endeavour Agile	Nivel	Hansoft	Nivel	Tuleap	Nivel
Calidad	2	Media	3	Alta	3	Alta
Funcionalidad	2	Media	3	Alta	3	Alta
Usabilidad	3	Alta	2		2	Media
Seguridad	3	Alta	3	Alta	3	Alta
Parametrización	1	Baja	3	Alta	2	Media
Soporte	2	Media	3	Alta	3	Alta
Licencias	3	Alta	1		3	Alta
Versionamiento	3	Alta	3	Alta	3	Alta
Uso de metodologías ágiles	1	Baja	3	Alta	1	Baja
Instalación	3	Alta	3	Alta	2	Media
Total	23		27		25	
Promedio	2,3		2,7		2,5	

Basándose en la *Tabla 7* se graficó los resultados, en los cuales se puede apreciar que Hansoft es la herramienta con mayor valoración, tal como lo muestra la *Figura 7*.

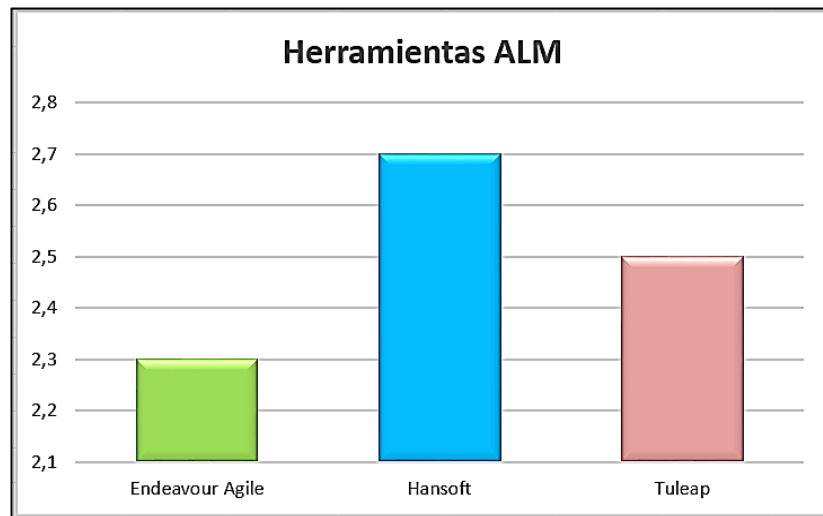


Figura 7: Resultados estadísticos de las herramientas con mayor valoración y cumplimiento de requerimientos.

La valoración de la *Tabla 8* se realizó en base a los requisitos obtenidos de las entrevistas (véase Anexos 2, 3, 4 y 5) y el empleo de las herramientas en el caso de estudio, mismos que fundamentan la valoración que tendrá de igual manera 3 como su máxima puntuación. Cabe destacar que en caso de que alguna característica no sea cumplida se asignara la letra “N”, que indicará la ausencia o incumplimiento total de esta.

Tabla VIII: Valoración de las Herramientas en base al cumplimiento de los requisitos.

Valoración						
Requerimientos	Endeavour Agile	Nivel	Hansoft	Nivel	Tuleap	Nivel
Tipo distribuida: ordenadores organizados en clústeres incrustados en una infraestructura de telecomunicaciones distribuida	1	Baja N	1	Baja N	1	Baja N
Servicios en la nube: provisión de servicios de hospedaje a través de Internet	2	Media A8F63	1	Baja N	2	Media A8F122
Control de versiones: registrar todos los cambios hechos en uno o más proyectos	3	Alta A8F93	3	Alta A8F120	3	Alta A8F161
Unificar el trabajo: unir los diferentes trabajos asignados a los	3	Alta A8F93	3	Alta A8F120	3	Alta A8F161

desarrolladores (código)						
Notificaciones: alertas sobre cambios en el proyecto	1	Baja N	2	Media A8F112	3	Alta A8F158
Gestión de tareas: asignación de responsables y visualización de estadísticas	3	Alta A8F71	3	Alta A8F99	3	Alta A8F143
Gestión de proyectos: actividades para alcanzar los objetivos y metas del proyecto	3	Alta A8F63	3	Alta A8F113	3	Alta A8F121
Gestión de procesos: conversión de requerimientos en características del producto	1	Baja A8F71	2	Media A8F99	2	Media A8F124
Gestión de productos: planificación y desarrollo de productos software	2	Media A8F63	2	Media A8F113	2	Media A8F121
Gestión de requisitos: identificación, asignación y seguimiento de los requisitos para la creación de un proyecto	1	Baja A8F70	3	Alta A8F98	3	Alta A8F124
Gestión de riesgos: minimización o evitación de riesgos	3	Alta A8F76	3	Alta A8F111	3	Alta A8F140
Gestión de tiempos: procesos necesarios para que el proyecto termine en el tiempo requerido	3	Alta A8F81	3	Alta A8F101	3	Alta A8F142
Gestión de costos: costo de los recursos que se utilizarán en las actividades desarrolladas en el proyecto	1	Baja N	1	Baja N	1	Baja N
Gestión de código: gestión de los cambios en el código fuente	3	Alta A8F93	3	Alta A8F120	3	Alta A8F161
Colaborativa: facilitan a los usuarios comunicarse y trabajar conjuntamente	2	Media A8F80	2	Media A8F117	3	Alta A8F148
Social: crecimiento y desarrollo facilitando la comunicación	2	Media A8F80	2	Media A8F117	3	Alta A8F148
Aseguramiento de calidad: los requisitos de calidad de un producto o servicio sean satisfechos	3	Alta A8F70	3	Alta A8F98	3	Alta A8F124
Control de pruebas: documentación adecuada para realizar correcciones	3	Alta A8F78	2	Media A8F96	2	Media A8F130
Control de diagramas: realización y modificación de diagramas	1	Baja A8F85	3	Alta A8F103	3	Alta A8F129

Multiusuario: servicio y procesamiento a múltiples usuarios simultáneamente	3	Alta A8F90	3	Alta A8F95	3	Alta A8F122
Multitarea: cumplen simultáneamente las necesidades de dos o más usuarios	3	Alta A8F64	3	Alta A8F100	3	Alta A8F157
A nivel de servicio: servicios de una herramienta vía web	1	Baja N	1	Baja N	2	Media A8F122
Ingeniería de software: conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de sw	2	Media A8F65	3	Alta A8F94	2	Media A8F123
Uso de metodologías ágiles: técnicas para la gestión de proyectos	1	Baja N	3	Alta A8F94	1	Baja A8F123
Estadísticas del proyecto: datos o representaciones mediante gráficos	2	Media A8F85	3	Alta A8F103	3	Alta A8F129
Adaptabilidad: capacidad de adaptación del sw a la empresa	1	Baja A8F63	3	Alta A8F94	2	Media A8F121
Uso de plantillas: para facilitar el uso de la herramienta (metodologías o proyectos)	1	Baja N	3	Alta A8F94	1	Baja A8F123
Interacción con el usuario: hacer más productivas las tareas que rodean a las personas y los computadores	3	Alta A8F64	3	Alta A8F95	3	Alta A8F153
Total	58		70		69	
Promedio	2,0714		2,5		2,4642	

Como se puede observar en esta valoración Tuleap posee un puntaje menor a Hansoft por 1, esto es debido a que Tuleap es superior en el ámbito social y colaborativo, pero en este caso como las herramientas fueron evaluadas en el ámbito educativo, es decir como parte del aprendizaje en la Escuela de Ingeniería en Sistemas, el ámbito social y colaborativo no representan un “peso” significativo al momento de seleccionar a una de estas herramientas, ya que este parámetro es muy utilizado cuando los miembros del equipo de trabajo no se encuentran en un solo lugar (ciudad o país). Pero este no es el caso en la institución de educación superior ya que el lugar de trabajo y desarrollo de proyectos se da en las aulas de la institución, lo que facilita la interacción y colaboración al ser todos alumnos y compañeros en este tipo de proyectos de desarrollo.

Es por esta razón que en la *Tabla 9* la nueva valoración, sin considerar el ámbito social y colaborativo quedaría de la siguiente manera:

Tabla IX: Valoración de las Herramientas en base al cumplimiento de los requisitos sin considerar el ámbito social y colaborativo.

Valoración						
Requerimientos	Endeavour Agile	Nivel	Hansoft	Nivel	Tuleap	Nivel
Tipo distribuida: ordenadores organizados en clústeres incrustados en una infraestructura de telecomunicaciones distribuida	1	Baja N	1	Baja N	1	Baja N
Servicios en la nube: provisión de servicios de hospedaje a través de Internet	2	Media A8F63	1	Baja N	2	Media A8F122
Control de versiones: registrar todos los cambios hechos en uno o más proyectos	3	Alta A8F93	3	Alta A8F120	3	Alta A8F161
Unificar el trabajo: unir los diferentes trabajos asignados a los desarrolladores (código)	3	Alta A8F93	3	Alta A8F120	3	Alta A8F161
Gestión de tareas: asignación de responsables y visualización de estadísticas	3	Alta A8F71	3	Alta A8F99	3	Alta A8F143
Gestión de proyectos: actividades para alcanzar los objetivos y metas del proyecto	3	Alta A8F63	3	Alta A8F113	3	Alta A8F121
Gestión de procesos: conversión de requerimientos en características del producto	1	Baja A8F70	2	Media A8F98	2	Media A8F124
Gestión de productos: planificación y desarrollo de productos software	2	Media A8F63	2	Media A8F113	2	Media A8F121
Gestión de requisitos: identificación, asignación y seguimiento de los requisitos para la creación de un proyecto	1	Baja A8F70	3	Alta A8F98	3	Alta A8F124
Gestión de riesgos: minimización o evitación de riesgos	3	Alta A8F76	3	Alta A8F111	3	Alta A8F140
Gestión de tiempos: procesos necesarios para que el proyecto termine en	3	Alta	3	Alta	3	Alta

el tiempo requerido		A8F81		A8F101		A8F142
Gestión de costos: costo de los recursos que se utilizarán en las actividades desarrolladas en el proyecto	1	Baja N	1	Baja N	1	Baja N
Gestión de código: gestión de los cambios en el código fuente	3	Alta A8F93	3	Alta A8F120	3	Alta A8F161
Aseguramiento de calidad: los requisitos de calidad de un producto o servicio sean satisfechos	3	Alta A8F70	3	Alta A8F98	3	Alta A8F124
Control de pruebas: documentación adecuada para realizar correcciones	3	Alta A8F78	2	Media A8F96	2	Media A8F130
Control de diagramas: realización y modificación de diagramas	1	Baja A8F85	3	Alta A8F103	3	Alta A8F129
Multiusuario: servicio y procesamiento a múltiples usuarios simultáneamente	3	Alta A8F90	3	Alta A8F95	3	Alta A8F122
Multitarea: cumplen simultáneamente las necesidades de dos o más usuarios	3	Alta A8F64	3	Alta A8F100	3	Alta A8F157
A nivel de servicio: servicios de una herramienta vía web	1	Baja N	1	Baja N	2	Media A8F122
Ingeniería de software: conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de sw	2	Media A8F65	3	Alta A8F94	2	Media A8F123
Uso de metodologías ágiles: técnicas para la gestión de proyectos	1	Baja N	3	Alta A8F94	1	Baja A8F123
Estadísticas del proyecto: datos o representaciones mediante gráficos	2	Media A8F85	3	Alta A8F103	3	Alta A8F129
Adaptabilidad: capacidad de adaptación del sw a la empresa	1	Baja A8F63	3	Alta A8F94	2	Media A8F121
Uso de plantillas: para facilitar el uso de la herramienta (metodologías o proyectos)	1	Baja N	3	Alta A8F94	1	Baja A8F123
Interacción con el usuario: hacer más productivas las tareas que rodean a las personas y los computadores	3	Alta A8F64	3	Alta A8F95	3	Alta A8F153
Total	53		64		60	
Promedio	2,12		2,56		2,4	

Conclusión:

En base a los criterios de evaluación y los requisitos obtenidos de las entrevistas, se ha podido establecer en la *Figura 8* que Hansoft es superior a Tuleap y Endeavour. Con un promedio de 2,56.

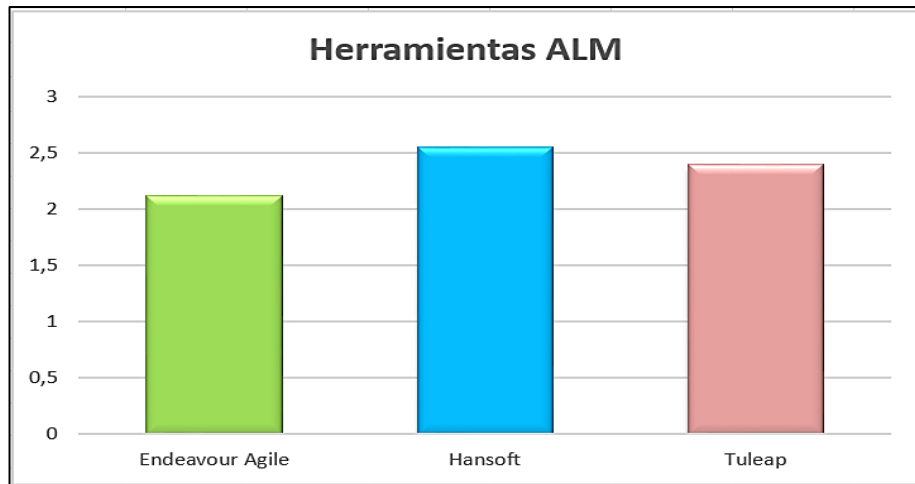


Figura 8: Resultados estadísticos de la herramienta con mayor promedio.

3.12. Sustento Académico Software Libre.

Como es de conocimiento general el Gobierno de la República de Ecuador promueve el uso e implementación de Software Libre en el sector público. Y mediante Decreto Ejecutivo No. 1014 emitido el 10 de Abril de 2008, se dispone el uso de Software Libre en los sistemas y equipamientos informáticos de la Administración Pública de Ecuador. Ya que es interés del Gobierno ecuatoriano alcanzar soberanía y autonomía tecnológica, así como un ahorro de recursos públicos.

La Subsecretaría de Gobierno Electrónico es responsable de elaborar y ejecutar planes, políticas y reglamentos para el uso de Software Libre en el Gobierno Central. Como órgano regulador desarrolla la "Estrategia para la implantación de Software Libre para la Administración Pública Central de Ecuador".

La Estrategia define políticas, objetivos y planes de acción en base a cuatro ejes estratégicos:

1. Promulgación de estándares y normatividad,
2. Formación de masa crítica,

3. Planificación, seguimiento y control y
4. Difusión del Software Libre

Se definen como políticas: la utilización de estándares abiertos, la minimización de compra de licencias propietarias, la contratación de servicios en proyectos informáticos, la reutilización del software y el uso preferencial de programas navegadores como medios de acceso. [64]

Decreto 1014

Art. 1: Establecer como política pública para las entidades de administración Pública central la utilización del Software Libre en sus sistemas y equipamientos informáticos.

Art. 2: Se entiende por software libre, a los programas de computación que se pueden utilizar y distribuir sin restricción alguna, que permitan el acceso a los códigos fuentes y que sus aplicaciones puedan ser mejoradas.

Estos programas de computación tienen las siguientes libertades:

- Utilización de programa con cualquier propósito de uso común.
- Distribución de copias sin restricción alguna
- Estudio y modificación de programa (Requisito: código fuente disponible)
- Publicación del programa mejorado (Requisito: código fuente disponible)

Art. 3: Las entidades de la administración pública central previa a la instalación del software libre en sus equipos, deberán verificar la existencia de capacidad técnica que brinde el soporte necesario para el uso de este tipo de software.

Art. 4: Se faculta la utilización de software propietario (no libre) únicamente cuando no exista una solución de software libre que supla las necesidades requeridas, o cuando esté en riesgo de seguridad nacional, o cuando el proyecto informático se encuentre en un punto de no retorno.

Art. 5: Tanto para software libre como software propietario, siempre y cuando se satisfagan los requerimientos.

Art. 6: La subsecretaría de Informática como órgano regulador y ejecutor de las políticas y proyectos informáticos en las entidades de Gobierno Central deberá realizar el control y seguimiento de este Decreto.

Art. 7: Encargue de la ejecución de este decreto los señores Ministros Coordinadores y el señor Secretario General de la Administración Pública y Comunicación. [65]

Por otra parte se tiene aquellas herramientas que funcionan como plugins y Gestor de Proyectos; mismas que no serán consideradas fundamentándose en la siguiente definición:

ALM es un proceso continuo de la gestión de la vida de una aplicación a través de la gobernanza, desarrollo y mantenimiento. En donde a estos tres aspectos se añade la visión de la infraestructura requerida por muchas de las empresas y comunidades de desarrollo que aplican ALM en su día a día. [66]



Figura 9: Características de una Herramienta ALM.

Es así como la Universidad Nacional de Loja, siendo una entidad pública se acoge al Decreto 1014 emitido por el señor presidente de la república, debido a que los estudiantes no poseen los recursos económicos para solventar el costo de adquisición de software privativo. Que si bien es cierto ofrece una gran cantidad de beneficios y funcionalidades, pero como en este caso si existe una solución de software libre que suple las necesidades requeridas en lo concerniente a Herramientas ALM, se justifica el hecho de no considerar las herramientas privativas en el análisis a realizado.

Además con la definición anteriormente citada se demuestra que la Gestión de proyectos es una de las muchas funcionalidades que debe cumplir una herramienta ALM y por esta razón tanto plugins como gestores de proyectos no formarán parte del análisis posterior.

4. Metodología más Usada.

4.1. Levantamiento de información

Antes de emplear la técnica de análisis estadístico o cualesquier otra, se inició con el levantamiento de información, que básicamente consistió en obtener datos específicos de los proyectos de titulación (software) desarrollados en la Escuela de Ingeniería en Sistemas, apoyándose en la técnica de Observación Estructurada en la cual se tiene que delimitar que aspectos se va a observar y que tipo de información se requiere.

4.2. Creación de la base de datos

Para la creación o construcción de la base de datos a utilizarse en la técnica de análisis estadístico se han considerado los siguientes criterios:

- Metodología
- Tipo de empresa (comercial / servicio)
- Tipo de entidad (pública / privada)
- Tipo de aplicación (escritorio / móvil / web / web_móvil)
- Tiempo [67]

4.3. Herramienta Pentaho

En este caso se utilizó Pentaho Analysis (véase Capítulo IV – Sección 4.1), cuyo elemento principal del sistema son los ficheros xml (Schema Workbench) donde se representan los esquemas dimensionales.

Pentaho Schema Workbench es la herramienta gráfica que permite la construcción de los esquemas de Mondrian, y además permite publicarlos al servidor BI para que puedan ser utilizados en los análisis por los usuarios de la plataforma. En los ficheros de esquema XML, se describen las relaciones entre las dimensiones y medidas del cubo (modelo multidimensional) con las tablas y campos de la base de datos, a nivel relacional. Este mapeo se utiliza para ayudar la traducción de las queries MDX (que es

el lenguaje con el que trabaja Mondrian), y para transformar los resultados recibidos de las consultas SQL a un formato dimensional.

Al crear un esquema (contenedor de cubos) se pueden indicar las siguientes propiedades: nombre, descripción, nombre para la dimensión que agrupará a las medidas y un rol por defecto para utilizar en las conexiones de base de datos. Luego se procede a la creación de cubos definiendo la estructura de tabla de hechos, medidas, miembros calculados y dimensiones. Tras construir el esquema, el paso final para poder utilizarlo en los análisis del portal BI de Pentaho es su publicación. [68]

Jpivot es utilizado para visualizar los resultados de los análisis, pero la visualización de estos es poco amigable. Entonces se optó por emplear el Plugin Saiku (de Analytics) en la *Figura 10* y *11* que permite armar consultas y generar gráficos como en la *Figura 12* de forma bastante intuitiva con las consultas al mismo cubo con JPivot, explorar fuentes de datos complejas, utilizando un lastre familiar y soltar interfaz, además de contar con una terminología de negocios fácil de entender, todo dentro de un navegador. Permitiendo seleccionar los datos que interesan, mirarlos desde diferentes perspectivas y perforar en el detalle. [69]

Información: 20:47 / 12 x 24 / 1.57s

metodologia	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
Ciclo de vida interactivo	-	-	-	-	-	-	-	1	-	-	-
Creacion de una metodologia	-	-	-	-	-	-	-	-	1	-	-
De Weiss y Kulkowski	-	-	-	-	-	-	-	-	2	-	-
En Cascada	-	-	-	-	1	-	1	-	-	2	-
GAIA	-	-	-	-	-	-	-	-	-	1	1
GRAPPLE	-	-	1	-	-	-	-	-	-	-	-
IDEAL	-	-	-	-	-	-	-	-	1	-	-
Iconix	2	3	7	8	24	19	14	6	26	15	1
Metodología de la red neuronal	-	-	-	-	-	-	-	-	-	1	-
Metodología híbrida	-	-	-	-	-	-	-	-	1	-	-
Mobile-D	-	-	-	-	-	-	-	-	-	6	2
Modelado Orientado a Objetos con UML	-	-	-	1	-	-	-	-	-	-	-
Prometheus	-	-	-	-	-	-	-	-	-	1	-
RAD (Rapid Application Development)	-	-	-	-	-	-	-	-	2	1	-
RUP	-	-	-	-	1	-	1	-	1	-	2
Rapid Application Development (RAD)	-	-	-	-	-	-	-	-	1	1	-
SCRUM	-	-	-	-	-	-	-	-	1	3	-
SCRUM y XP (Extreme Programming)	-	-	-	-	-	1	-	-	-	-	-
SUM	-	-	-	-	-	-	-	-	-	-	1
UWE	-	-	-	-	1	-	-	-	1	-	2
XP (Extreme Programming)	-	-	3	-	3	5	5	4	5	12	2

Figura 10: Interfaz de Pentaho.

metodologia	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
Ciclo de vida interactivo	-	-	-	-	-	-	-	1	-	-	-
Creacion de una metodologia	-	-	-	-	-	-	-	-	1	-	-
De Weiss y Kulikowski	-	-	-	-	-	-	-	-	2	-	-
En Cascada	-	-	-	-	1	-	1	-	-	2	-
GAIA	-	-	-	-	-	-	-	-	-	1	1
GRAPPLE	-	-	1	-	-	-	-	-	-	-	-
IDEAL	-	-	-	-	-	-	-	-	1	-	-
Iconix	2	3	7	8	24	19	14	6	28	15	1
Metodología de la red neuronal	-	-	-	-	-	-	-	-	-	1	-
Metodología híbrida	-	-	-	-	-	-	-	-	1	-	-
Mobile-D	-	-	-	-	-	-	-	-	-	6	2
Modelado Orientado a Objetos con UML	-	-	-	1	-	-	-	-	-	-	-
Prometheus	-	-	-	-	-	-	-	-	-	1	-
RAD (Rapid Application Development)	-	-	-	-	-	-	-	-	2	1	-
RUP	-	-	-	-	1	-	1	-	1	-	2
Rapid Aplication Development (RAD)	-	-	-	-	-	-	-	-	1	1	-
SCRUM	-	-	-	-	-	-	-	-	1	3	-
SCRUM y XP (Extreme Programming)	-	-	-	-	-	1	-	-	-	-	-
SUM	-	-	-	-	-	-	-	-	-	-	1
UWE	-	-	-	-	-	1	-	-	1	-	2
XP (Extreme Programming)	-	-	3	-	3	5	5	4	5	12	2
ffff	-	-	1	-	-	1	-	-	-	-	-
método para el desarrollo de una distribución	-	-	-	1	-	-	-	-	-	-	-

Figura 11: Análisis estadístico resultante.

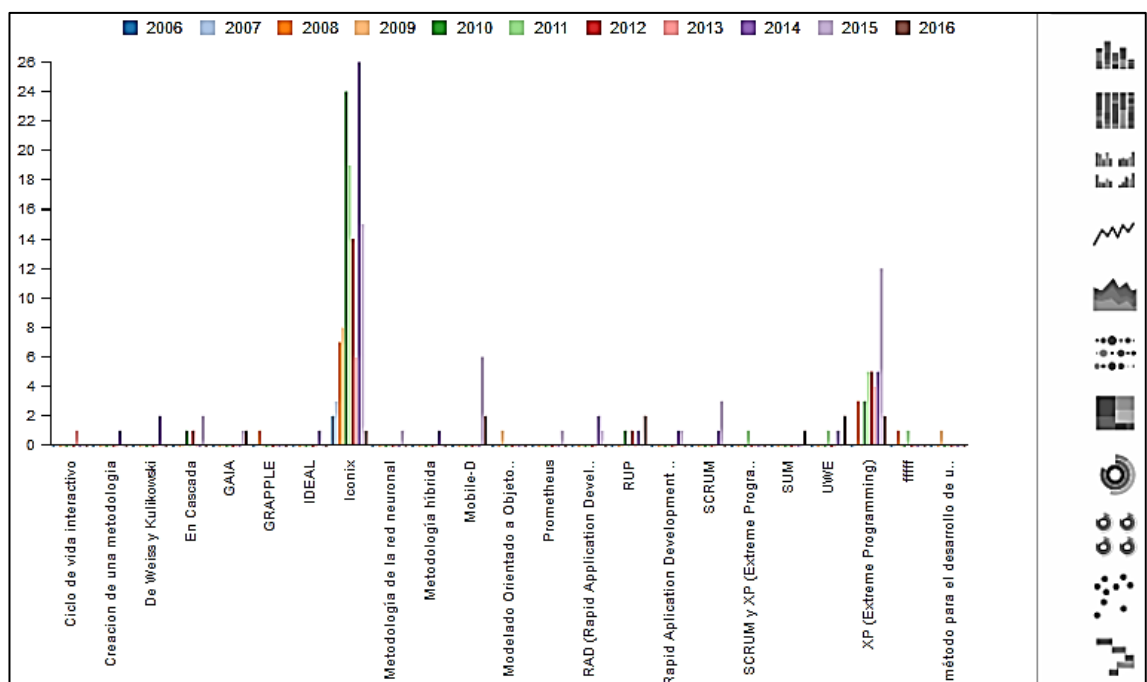


Figura 12: Gráfica del análisis resultante.

4.4. Metodología

Haciendo referencia a la información de las metodologías (véase Capítulo IV – Sección 4.2); es importante resaltar que en base al cubo OLAP utilizado para obtener la información de la metodología de desarrollo más empleada en la CIS, el resultado es la metodología ICONIX. Pero esta es predominante debido al auge de la misma en los inicios de la escuela (2006) y al desarrollo de aplicaciones de escritorio de ese tiempo. Es por esta razón que se realizó una nueva evaluación en la *Tabla 10* considerando las metodologías empleadas desde el año 2013; ya que a partir de este año las aplicaciones desarrolladas son orientadas a la web.

Tabla X: Extracto del análisis resultante obtenido en Pentaho.

Metodología	2013	2014	2015	2016
En Cascada	-	-	2	-
GAIA	-	-	1	1
Iconix	6	26	15	1
Mobile D	-	-	6	2
RAD	-	3	2	-
RUP	-	1	-	2
Scrum	-	1	3	-
UWE	-	1	-	2
XP	4	5	12	2

En base al extracto anterior de las metodologías, se puede observar aquellas que han sido empleadas en los últimos 4 años, teniendo un importante predominio de las metodologías Iconix y XP; siendo estas tradicional y ágil respectivamente.

Seguidamente se realizará una comparativa en la *Tabla 11* que permitirá efectuar un filtro final en un posterior análisis.

Tabla XI: Comparativa entre Metodologías Tradicionales y Web.

Metodologías Tradicionales	Metodologías Web
Imponen una disciplina de trabajo sobre el proceso de desarrollo del software	Buscan el equilibrio en relación al proceso/esfuerzo
No se adaptan adecuadamente a los cambios	Son adaptables y progresan con el cambio
El grado de interacción con el usuario varía según la metodología	Alto grado de interacción con el usuario
La facilidad de uso depende del tipo de aplicación a desarrollar	Facilidad de uso más presente
Todas poseen fases similares (estándar)	En la fase de diseño comprende diferentes análisis (diseño o análisis navegacional)

Periodo de implantación varia	Periodo de implantación mucho más reducido
Menor soporte técnico	Mayor soporte técnico

[50] [70] [71]

Es así que debido al incremento del desarrollo de aplicaciones web en la actualidad, la base de datos que contiene la información de los proyectos de titulación de la CIS (usada en cubo OLAP), la comparativa entre metodologías Tradicionales y Web de la *Tabla 10* y el orden cronológico del uso de las metodologías (véase Capítulo IV – Sección 4.2); no serán consideradas algunas de ellas, principalmente porque hay metodologías que se ven desfasadas con el paso del tiempo (1970 a 1999), algunas son empleadas en el desarrollo móvil y otras en el desarrollo de sistemas expertos, incluyendo combinación o creación de metodologías orientadas a la solución de un determinado problema. Quedando únicamente XP y UWE (originadas a partir del año 2000) en la *Tabla 12* para realizar la evaluación y posterior parametrización de una única metodología en la herramienta Hansoft ALM.

Tabla XII: Descarte de Metodologías en base a su año y tipo de aplicación.

Metodología	Tipo de aplicación	Año	Tipo de metodología
En Cascada	Escritorio	1970	Tradicional
GAIA	Escritorio (sistemas expertos)	2000	Desarrollo de SE
Iconix	Escritorio	1993	Tradicional
Mobile D	Móvil	2004	Ágil
RAD	Escritorio	1991	Tradicional
Scrum	Escritorio y Móvil	1980	Tradicional
RUP	Escritorio	1999	Tradicional
XP	Escritorio y Web	2000	Ágil
UWE	Web	2000	Tradicional

4.5. Parametrización

Se realizó la parametrización de la metodología UWE en Hansoft ALM, ya que esta no consta en las plantillas de la misma. Para lo cual es necesario crear un nuevo proyecto como en la *Figura 13*, en el cual se observará la opción **Method Template**, misma que permite seleccionar la plantilla para las metodologías ya establecidas. Pero en este caso se tiene 5 plantillas listas para ajustarlas a las metodologías que se requiera

(Custom). Se selecciona la opción Custom 2 ya que la 1 esta parametrizada con la metodología Mobile D.

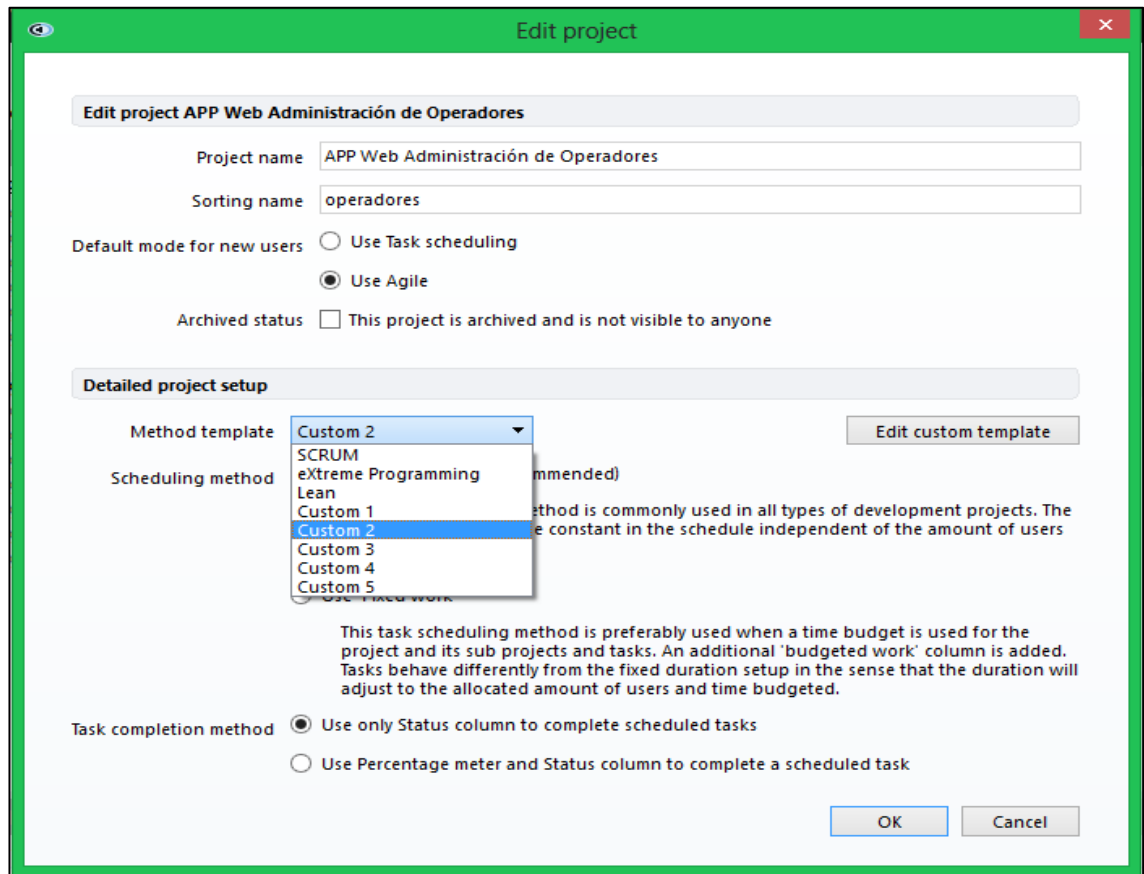


Figura 13: Inicio de la parametrización de una nueva metodología.

Es necesario aclarar que la parametrización consiste básicamente en asignar nombres característicos de la metodología a adaptarse en la opción **Edit custom tamplate** en la *Figura 14* de la siguiente manera:

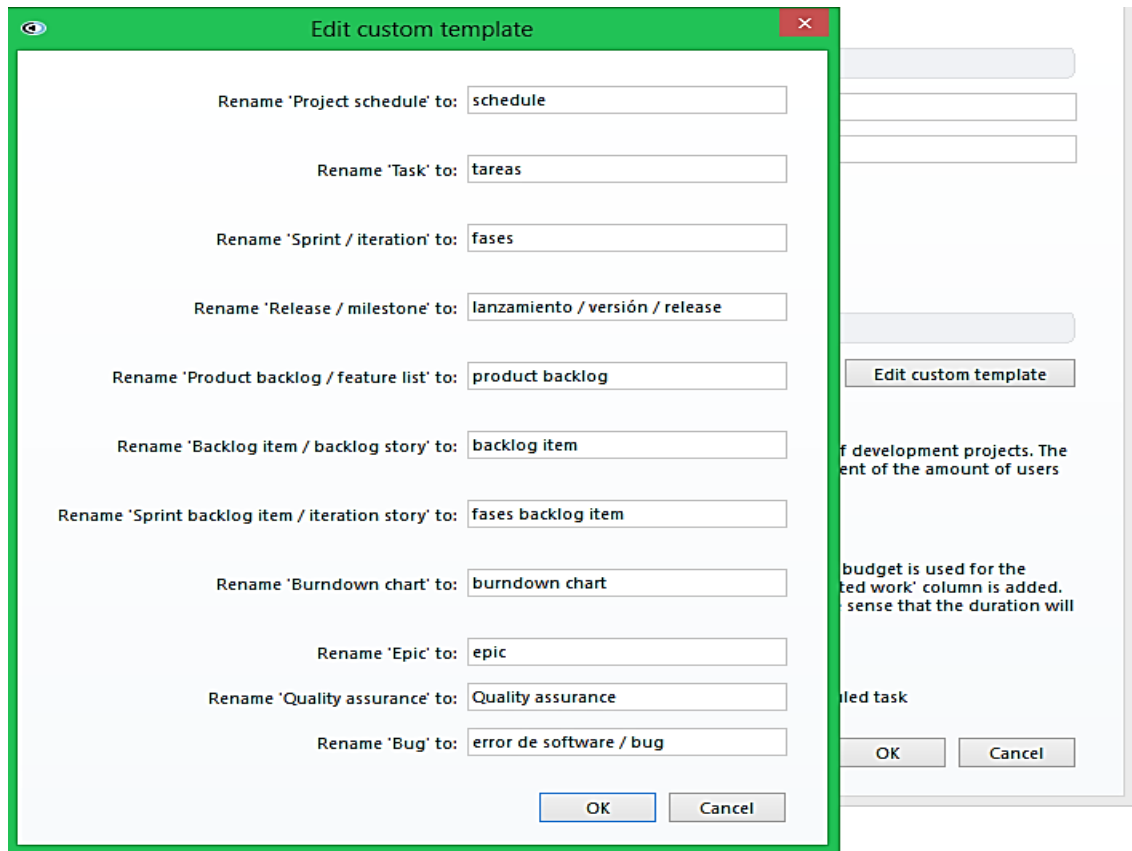


Figura 14: Parametrización de los datos de la metodología.

Antes de asignar los nuevos nombres a los campos, es necesario tener en claro los que corresponden o caracterizan a la metodología a parametrizar. En este caso UWE se caracteriza por estar dividido en fases tal como lo muestra la *Figura 15*.

Product backlog		Quality assurance		Planning										
Agile Schedule		Add fases backlog item		Add fases		Add release		Show		More		My Reports: Status		
Item name	Assigned to	Status	Work remaining	Fases priority	5 February		W06		1		2		3	
5	Ingeniería de Requisitos 2016-02-01 - 2016-03-14	✓ Completed												
1:	Inicio de ingeniería de requerimientos	Nestor	✓ Completed											
1:	Diseño del Sistema 2016-03-15 - 2016-05-16	✓ Completed												
2:	Codificación 2016-05-17 - 2016-07-25	✓ Completed												
2:	Pruebas 2016-07-26 - 2016-08-31	✓ Completed												

Figura 15: Interfaz de la metodología parametrizada en Hansoft ALM.

Tomando en cuenta que UWE está basada en UML y sus fases poseen gran énfasis en el modelo conceptual, navegacional, de presentación y de procesos. Estos se ven reflejados en la parte de diseño del sistema. Donde se pueden añadir o quitar los modelos adicionales a estos como en la *Figura 16*.

1:	<ul style="list-style-type: none"> <ul style="list-style-type: none"> Diseño del Sistema 2016-03-15 - 2016-05-16 <ul style="list-style-type: none"> User allocations Click here to allocate users to this fases Burndown chart Fases backlog items <ul style="list-style-type: none"> Modelado UWE <ul style="list-style-type: none"> Modelo conceptual Nestor ✓ Completed 0h Modelo navegacional Nestor ✓ Completed 0h Modelo de presentación Nestor ✓ Completed 0h Modelo de procesos Nestor ✓ Completed 0h Diagrama de casos de uso Nestor ✓ Completed 0h Modelo entidad relación Nestor ✓ Completed 0h 					
----	---	--	--	--	--	--

Figura 16: Modelado de UWE en la fase de Diseño del sistema.

Como se puede apreciar, la parametrización es sencilla, el éxito de la misma radica en conocer la metodología con la que se va a trabajar y así poder plasmar sus aspectos esenciales en la herramienta.

4.6. Propuesta

Como es de conocimiento general, actualmente los alumnos de la Escuela de Ingeniería en Sistemas de la Universidad Nacional de Loja no usan ningún tipo de software o herramienta que apoye al desarrollo, control y gestión de las aplicaciones que estos desarrollan como parte de su formación en el ámbito educativo.

Primeramente cabe destacar el desconocimiento de las herramientas ALM, no solo por parte de estudiantes sino también de algunos docentes. En este caso, los alumnos como futuros Ingenieros en Sistemas están en la obligación de conocer las diferentes soluciones tecnológicas y formas de apoyo al desarrollo de software que existe en la actualidad. Es aquí donde entran estas herramientas, que facilitan o permiten llevar un control adecuado del desarrollo de proyectos software; y así mismo su uso o explotación proporciona a los alumnos la obtención de conocimientos básicos antes de adentrarse al campo laboral.

Por otra parte, una vez que los alumnos inicien su etapa laboral necesitan realizar constantes controles o supervisiones de los diferentes proyectos a los que sean asignados; estas herramientas facilitan la gestión, obtención de avances y estadísticas de los proyectos en curso.

Por lo cual se propone Hansoft ALM ya que posee un conjunto de características que beneficiarán al alumnado, tales como:

- Realizar una adecuada gestión del ciclo de vida de las aplicaciones.
- Llevar un adecuado control de tiempos y cumplimiento de las aplicaciones.
- Mejora de la calidad del producto.
- Evitar reportes manuales, ya que la información y estadísticas se encuentran en la herramienta.
- Es una herramienta open source.
- Ingreso de documentación del proyecto en desarrollo.
- Seguimiento desde el origen hasta el final del proyecto.

4.7. Guía

1. INSTALACIÓN

Para iniciar con el uso de Hansoft, primeramente se debe ingresar a su página oficial y descargar la herramienta según sea su sistema operativo, tal como se muestra en la *Figura 17*:

<https://hansoft.com/downloads/>

The screenshot displays two sections of the Hansoft website. The top section is titled 'All-in-One Small Team Edition' and includes the text: 'If you want to evaluate Hansoft or for training purposes (all features but limited to 9 users)'. It features three buttons: 'Installer for Windows', 'Installer for OS X', and 'Tutorial'. To the right, there is a diagram showing a monitor icon labeled 'Client', a database icon labeled 'Server', and a document icon labeled 'Example Projects', all connected by plus signs. The bottom section is titled 'Client' and includes the text: 'If you want to access Hansoft servers.' It features three buttons: 'Installer for Windows', 'Installer for OS X', and 'Installers for Linux', followed by a 'Tutorial' button. To the right, there is a similar diagram with 'Client', 'Server', and 'Example Projects' icons. Below the buttons in the 'Client' section, there is a note: 'Please note that the Client will automatically adapt to the correct version when you connect to a Hansoft server so installing the client from this page is something you typically only do once.'

Figura 17: Opciones a disponibles para descargar la herramienta.

Si se va a trabajar como administrador, se debe descargar la versión **All-in-One Small Team Edition**, pero si se va a trabajar como miembro de un proyecto se debe descargar la versión **Client**.

La instalación en Windows es muy sencilla, únicamente se ejecuta el archivo y automáticamente se instala la herramienta. Pero en Ubuntu se debe seguir los comandos que proporciona la página oficial de la herramienta.

¡Y listo!

2. ADMIN / ADMINISTRADOR

2.1 CREAR UN NUEVO PROYECTO

Primeramente se debe seleccionar la pestaña **Projects** como en la *Figura 18*, misma que desplegará una nueva ventana en donde se ingresarán los datos del proyecto a crear. En donde se ingresará el nombre

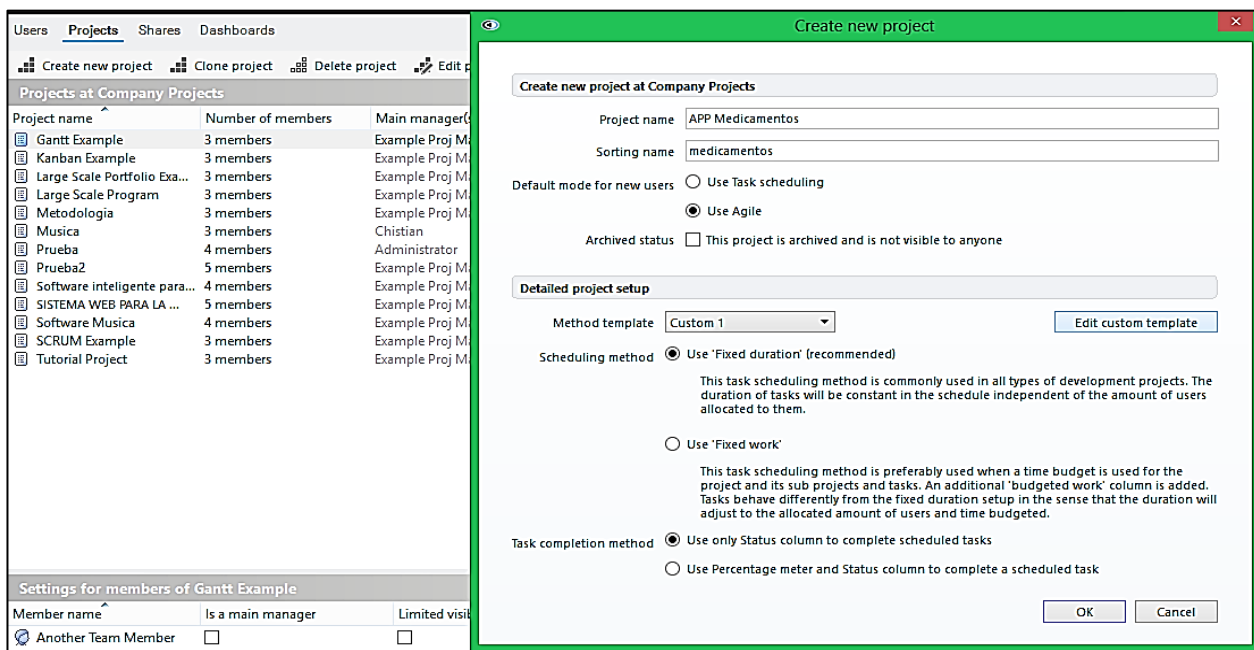


Figura 18: Creación de un nuevo proyecto.

Seguidamente se debe proceder a llenar los datos del proyecto a crear como lo muestra la *Figura 19*, tomando en cuenta las restricciones y características solicitadas para cada uno de los campos.

Hansoft posee plantillas para las metodologías ágiles como: Scrum y XP. Pero en este caso la metodología a usar es Mobile-D, esta herramienta permite ajustar la plantilla que posee de acuerdo a lo que requiere la metodología con la que se va a trabajar, esto se puede realizar en el botón **Edit custom template**. Cabe recalcar que la plantilla que ofrece es la misma para todas las metodologías que se vayan a usar, la diferencia es que no permite asignar un nombre a la misma, pero si se puede cambiar el nombre a los links o pestañas de la interfaz como se muestra a continuación:

The image shows a dialog box titled "Edit custom template" with a green header bar. Inside the dialog, there is a list of items to be renamed, each with a text input field. The items and their corresponding values are:

- Rename 'Project schedule' to:
- Rename 'Task' to:
- Rename 'Sprint / iteration' to:
- Rename 'Release / milestone' to:
- Rename 'Product backlog / feature list' to:
- Rename 'Backlog item / backlog story' to:
- Rename 'Sprint backlog item / iteration story' to:
- Rename 'Burndown chart' to:
- Rename 'Epic' to:
- Rename 'Quality assurance' to:
- Rename 'Bug' to:

At the bottom right of the dialog, there are two buttons: "OK" and "Cancel".

Figura 19: Listado de datos a parametrizar según la metodología del proyecto.

2.2 AGREGAR MIEMBROS AL PROYECTO

Una vez que se ha culminado con el registro del nuevo proyecto, se podrá añadir los usuarios o miembros del mismo, tal como se realizó en la *Figura 20*. Debido a que se está trabajando con la versión libre o comunitaria el límite de usuarios es de 9. Se pueden utilizar los usuarios que vienen por defecto en la herramienta o crear uno nuevo, en la pestaña **Create new user**. En donde se ingresará el nombre, email, clave y datos que el administrador considere necesarios acerca de la misma.

The screenshot shows a 'Create new user' dialog box with the following fields and options:

- Name: Christian
- Sorting Name: chris
- E-mail: cgbustamantef@unl.edu.ec
- Password: [Empty]
- Confirm Password: [Empty]
- Security**
 - User must change password next logon
 - Password never expires
 - User cannot change password
 - User cannot log in using Hansoft password
- Avatar**
 - User can change their avatar

Figura 20: Creación de un nuevo usuario.

Una vez ingresados los datos del nuevo usuario, se lo ingresará a un grupo o se creará uno nuevo en donde ingresarlo, en este caso se creó un grupo de programadores, donde el nuevo usuario en la *Figura 21* formará parte.

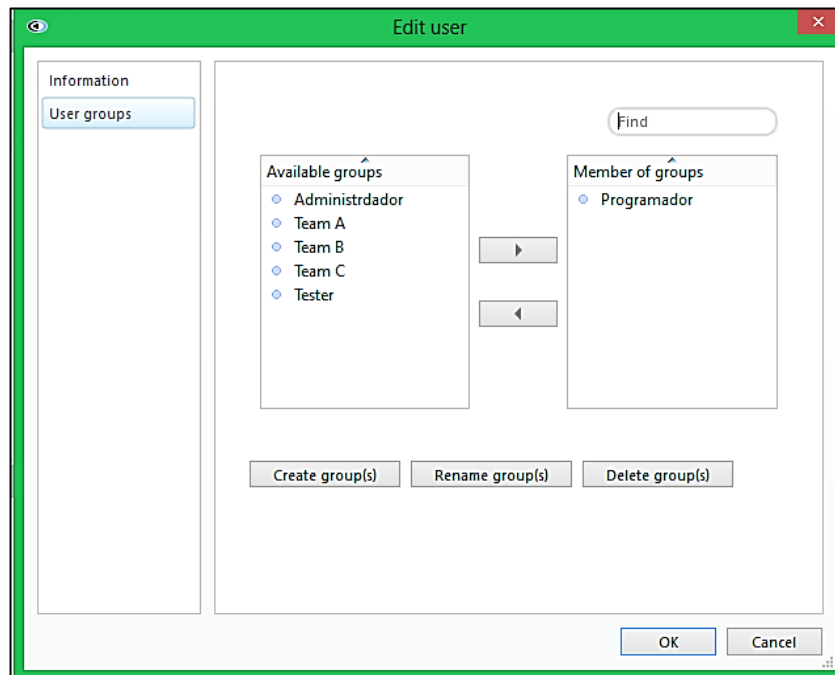


Figura 21: Ingreso de la información del usuario.

Se asignará los miembros al proyecto como en la *Figura 22* pero se recomienda usar el **Example Proj Manager** ya que este hará el papel de administrador y permitirá trabajar con normalidad en las plantillas, para seguidamente incluir al usuario creado en este caso.

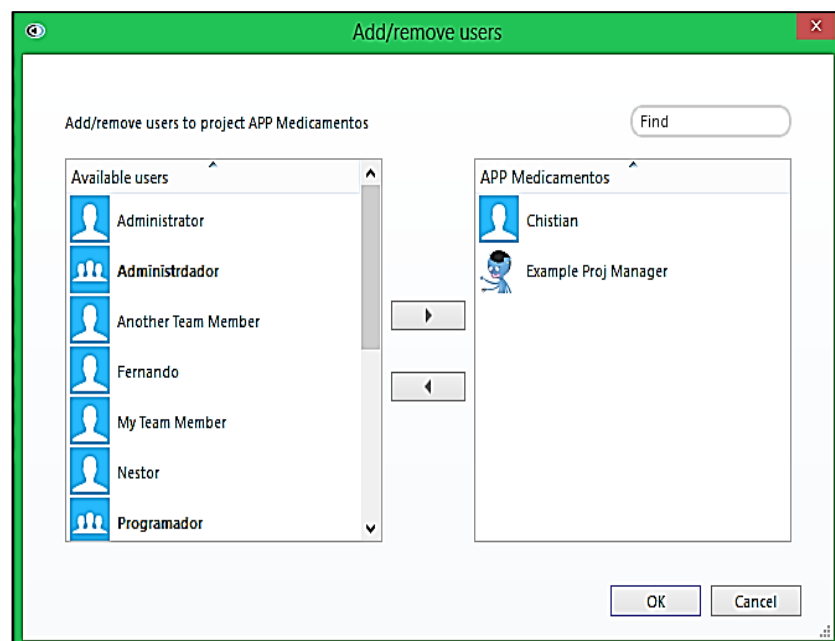


Figura 22: Asignación de los miembros del proyecto.

Como en todo proyecto, a los usuarios se les asignará el tipo de visibilidad tal como está en la *Figura 23*. Esta se encuentra en la parte inferior de la interfaz de Hansoft, y se visualizará al seleccionar el nombre del proyecto.

The screenshot shows the 'ADMIN' section of the Hansoft interface. At the top, there are navigation tabs: 'TO DO LIST', 'DASHBOARDS', 'PROJECT: PRUEBA', and 'ADMIN'. Below these are sub-tabs: 'Users', 'Projects', 'Shares', and 'Dashboards'. A toolbar contains actions: 'Create new project', 'Clone project', 'Delete project', 'Edit project', 'Add/remove users to project', and 'More'. The main content area is titled 'Projects at Company Projects' and contains a table with columns: 'Project name', 'Number of members', and 'Main manager(s)'. The row for 'APP Medicamentos' is highlighted. Below this table is a section titled 'Settings for members of APP Medicamentos' with a table for member settings.

Project name	Number of members	Main manager(s)
<input type="checkbox"/> Gantt Example	3 members	Example Proj Manager
<input type="checkbox"/> Kanban Example	3 members	Example Proj Manager
<input type="checkbox"/> Large Scale Portfolio Exa...	3 members	Example Proj Manager
<input type="checkbox"/> Large Scale Program	3 members	Example Proj Manager
<input checked="" type="checkbox"/> APP Medicamentos	2 members	Example Proj Manager
<input type="checkbox"/> Metodologia	3 members	Example Proj Manager
<input type="checkbox"/> Musica	3 members	Chistian
<input type="checkbox"/> APP Web Administració...	2 members	Example Proj Manager
<input type="checkbox"/> Prueba	4 members	Administrator
<input type="checkbox"/> Prueba2	4 members	Example Proj Manager
<input type="checkbox"/> Software inteligente para...	4 members	Example Proj Manager
<input type="checkbox"/> SISTEMA WEB PARA LA ...	4 members	Example Proj Manager
<input type="checkbox"/> Software Musica	4 members	Example Proj Manager
<input type="checkbox"/> SCRUM Example	3 members	Example Proj Manager
<input type="checkbox"/> Tutorial Project	3 members	Example Proj Manager

Member name	Is a main manager	Limited visibility	Can access project history
<input type="checkbox"/> Christian	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Example Proj Manager	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 23: Asignación del tipo de visibilidad al miembro del proyecto.

3. PROJECT / EL PROYECTO

Ahora, se debe ir a la sección **PROJECT**: donde se encontrará una lista de ejemplos de proyectos y se debe seleccionar el proyecto que se ha creado, en este caso APP Medicamentos como en la *Figura 24*.

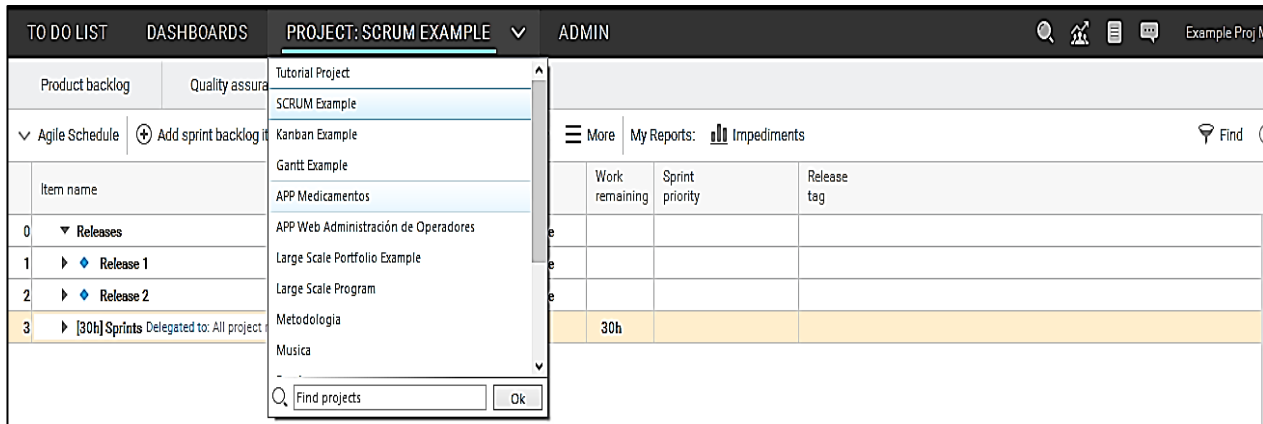


Figura 24: Selección del proyecto creado.

3.1 PLANNING / PLANIFICACIÓN

3.1.1 ETAPAS / FASES

Una vez previamente establecidas las etapas y sus tareas, se procede a ingresarlas en Hansoft, mediante la opción **Add etapas / fases** de la *Figura 25*, misma que aparecerá en la parte inferior lista para asignarle un nombre, una vez que se la ha creado aparecerá en la parte derecha a la cual se la puede mover entre las fechas del calendario hasta ubicarla en la fecha deseada y seguidamente ingresar las tareas. Cabe destacar que en la parte inferior hay una sección para añadir comentario en relación a la etapa o tarea.

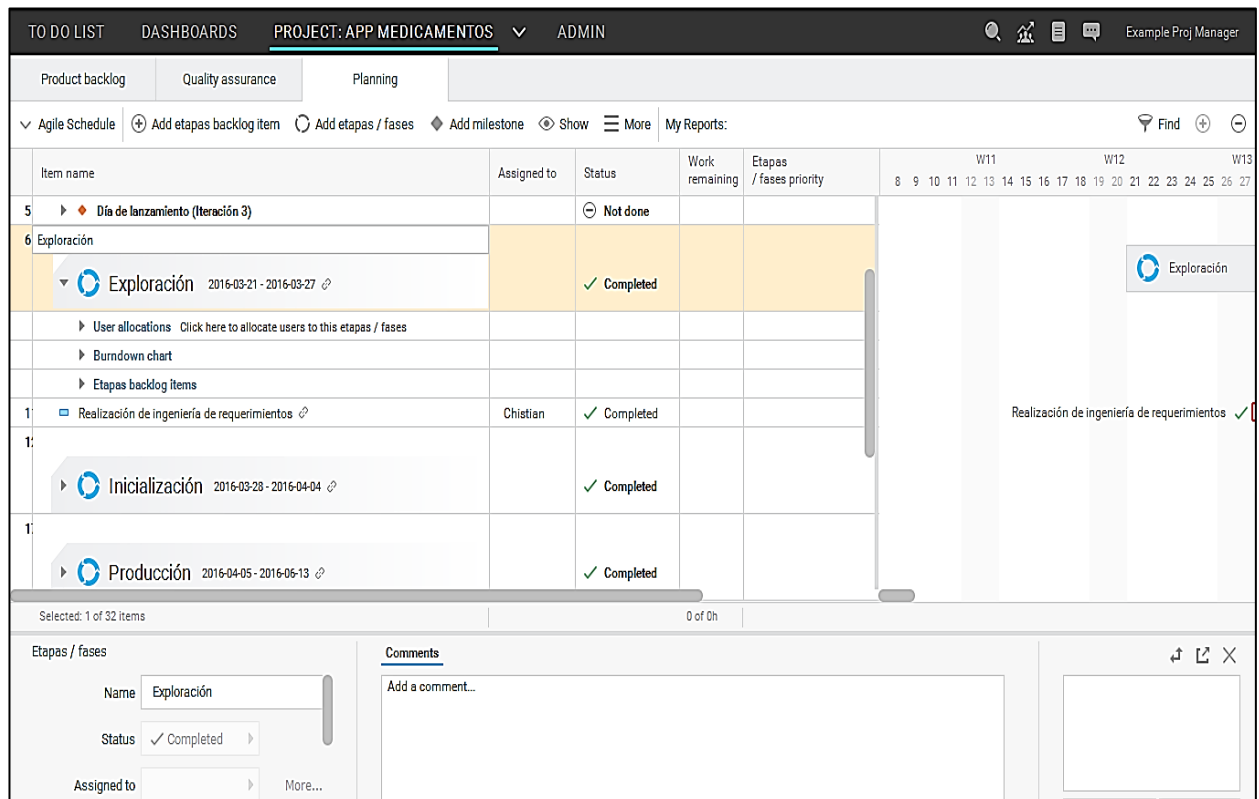


Figura 25: Ingreso de etapas o fases del proyecto.

Mediante la opción **Click here** de **Users allocations** se puede asignar el miembro del proyecto que estará a cargo de esa etapa, tal como lo muestra la *Figura 26*.

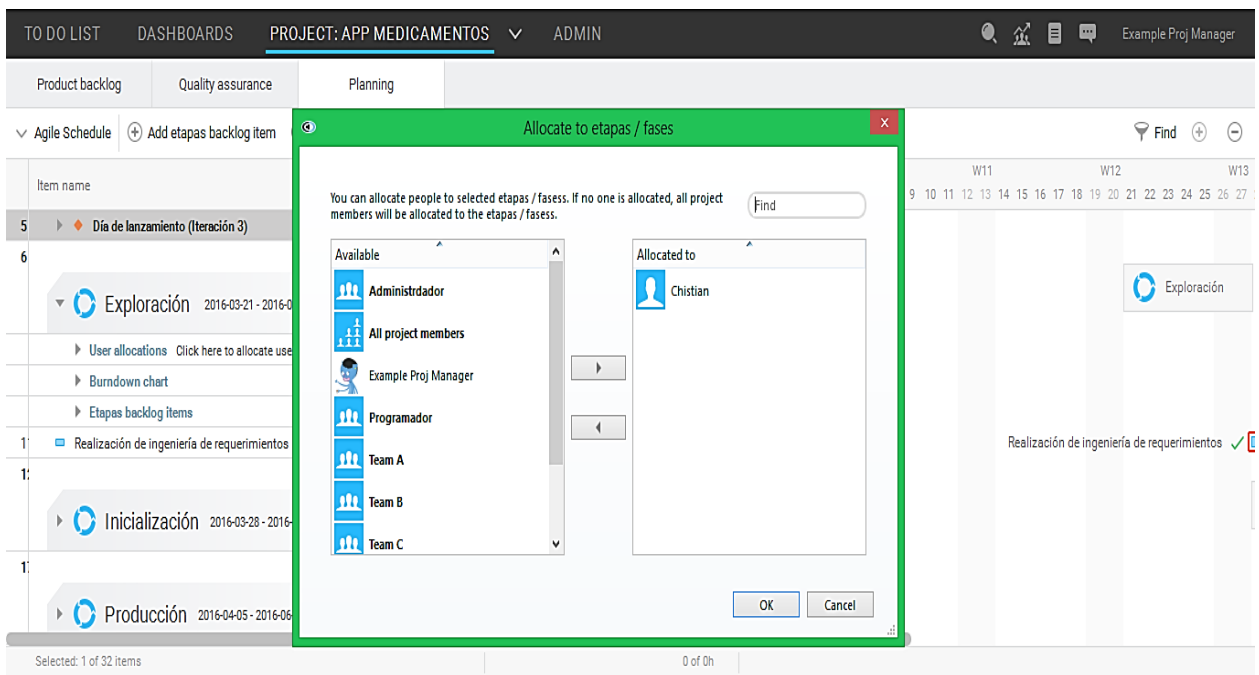


Figura 26: Miembro a cargo del proyecto.

3.1.2 ETAPAS BACKLOG ITEMS / TAREAS

Para ingresar las tareas se va a la opción **Add etapas backlog ítem** y automáticamente aparecerá una nueva tarea lista para asignarle un nombre como se puede apreciar en la *Figura 27*. En caso de que esta posea subtareas se usa la tecla Tab y automáticamente se creará una subclasificación. Adicionalmente a estas tareas se puede asignar un status, prioridad, releases, etc.

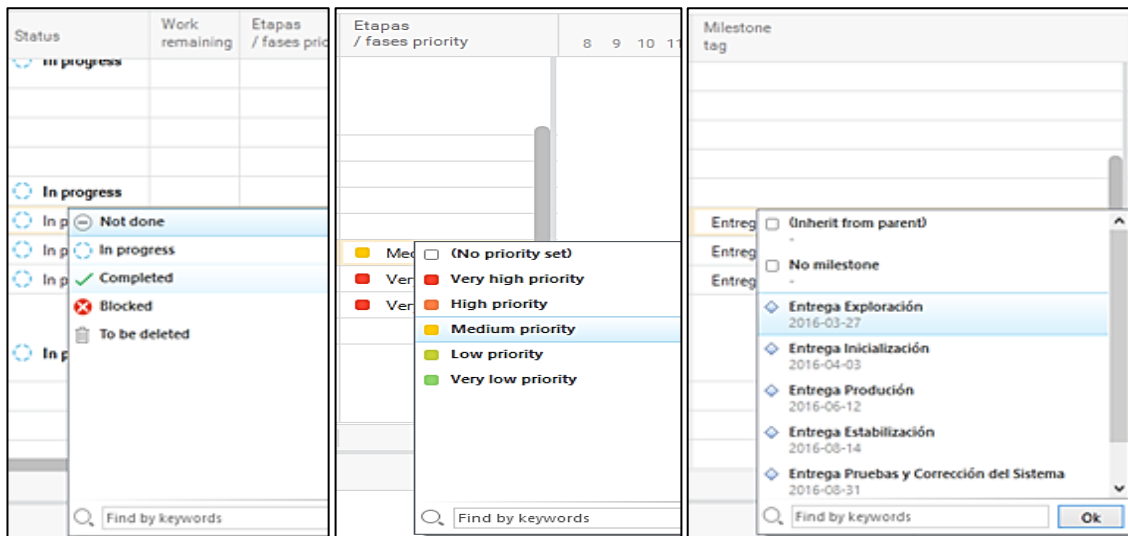


Figura 27: Status de las tareas.

Si se requiere adicionar más características a las tareas se va a la opción **More** y se selecciona **Customise Project...** de la *Figura 28*, misma que presentará un conjunto de opciones, en este caso se seleccionará **Hyperlink**, que permitirá acceder al conjunto de documentos que constan en una carpeta de Drive del proyecto.

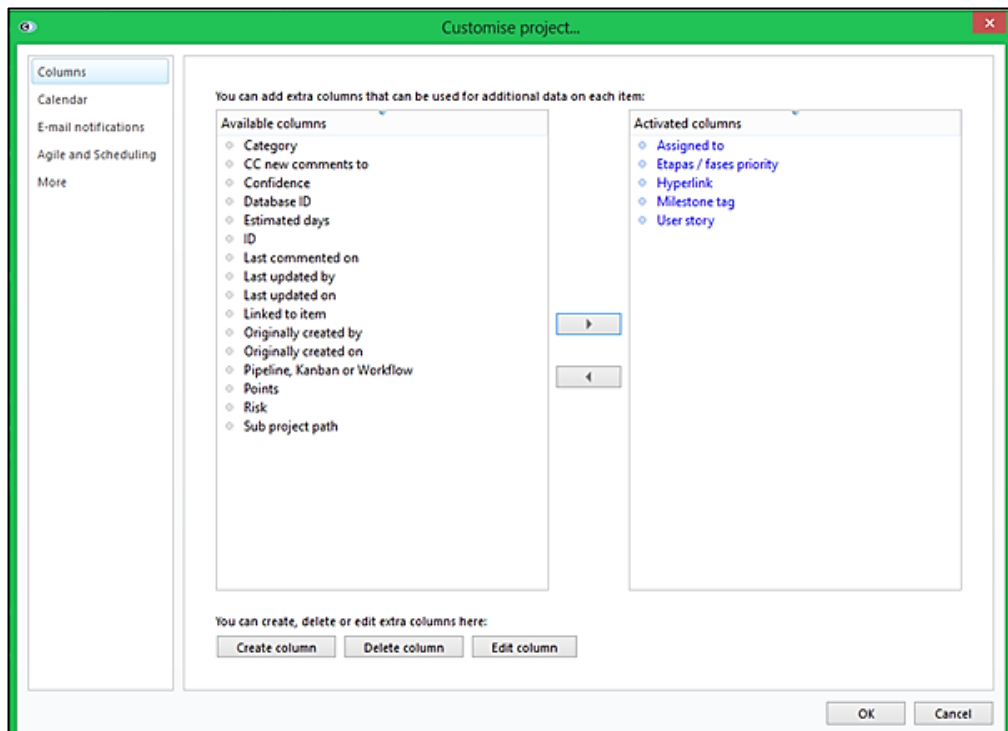


Figura 28: Características adicionales de las tareas.

Así automáticamente aparecerá la opción de **Hyperlink**, donde únicamente se hace doble click y se ingresa la dirección url del documento como en la *Figura 29*.

Item name	Assigned to	Status	Work remaining	Etapas / fases priority	Milestone tag	Hyperlink
<ul style="list-style-type: none"> ▼ Exploración 2016-03-21 - 2016-03-27 🔗 <ul style="list-style-type: none"> ▶ User allocations Click here to allocate users to this etapas / fases ▶ Burndown chart ▼ Etapas backlog items <ul style="list-style-type: none"> ▼ Información Inicial <ul style="list-style-type: none"> ■ Establecimiento de stakeholders Chistian ✓ Completed 0h ■ Medium priority Día de lanzamiento ■ Establecimiento del proyecto Chistian ✓ Completed 0h ■ Very high priority Día de lanzamiento ■ Definición del alcance Chistian ✓ Completed 0h ■ Very high priority Día de lanzamiento ■ Realización de ingeniería de requerimientos 🔗 Chistian ✓ Completed https://drive.google.com/a/unl.edu.ec/fold 		✓ Completed			Día de lanzamiento	https://drive.google.com/a/unl.edu.ec/fold
Selected: 1 of 32 items 0 of 0h						

Figura 29: Creación de un hyperlink.

3.1.3 RELEASES

Para ingresar Releases al proyecto, se hace click derecho y se selecciona la opción Insert sub Project, donde aparecerá una nueva celda en la cual se ingresará el nombre, seguidamente se inserta un nuevo subproyecto y mediante la tecla Tab se crea una subdivisión o clasificación como lo muestra la *Figura 30*. De igual manera que las etapas, los releases también aparecen en la parte derecha y pueden moverse hasta ubicarlos en la fecha deseada.

El status de los releases se actualiza automáticamente al culminar las tareas que dependen de este.

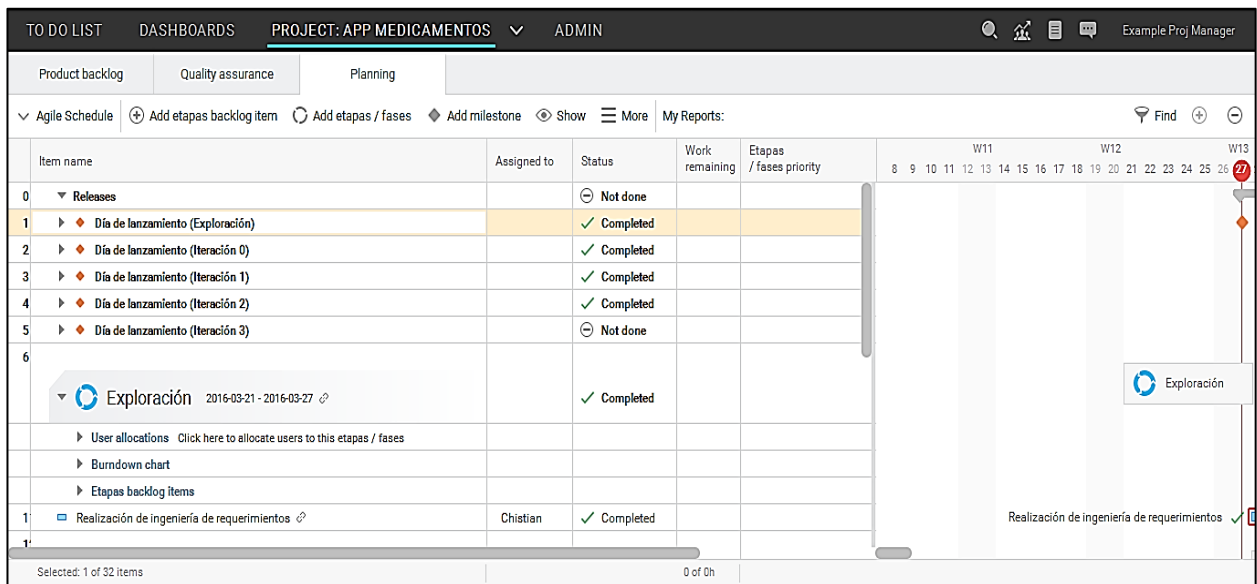


Figura 30: Creación de releases.

3.1.4 REPORTES

Se crean yendo a la sección **My Reports:** de la *Figura 31* donde esta permitirá seleccionar lo que se desea mostrar del proyecto, como el status, fases, releases etc. Proporcionando una visibilidad acorde a las necesidades del administrador o miembro del proyecto, permitiendo la creación de carpetas y reportes como en la *Figura 32* para llevar de forma ordenada los mismos.

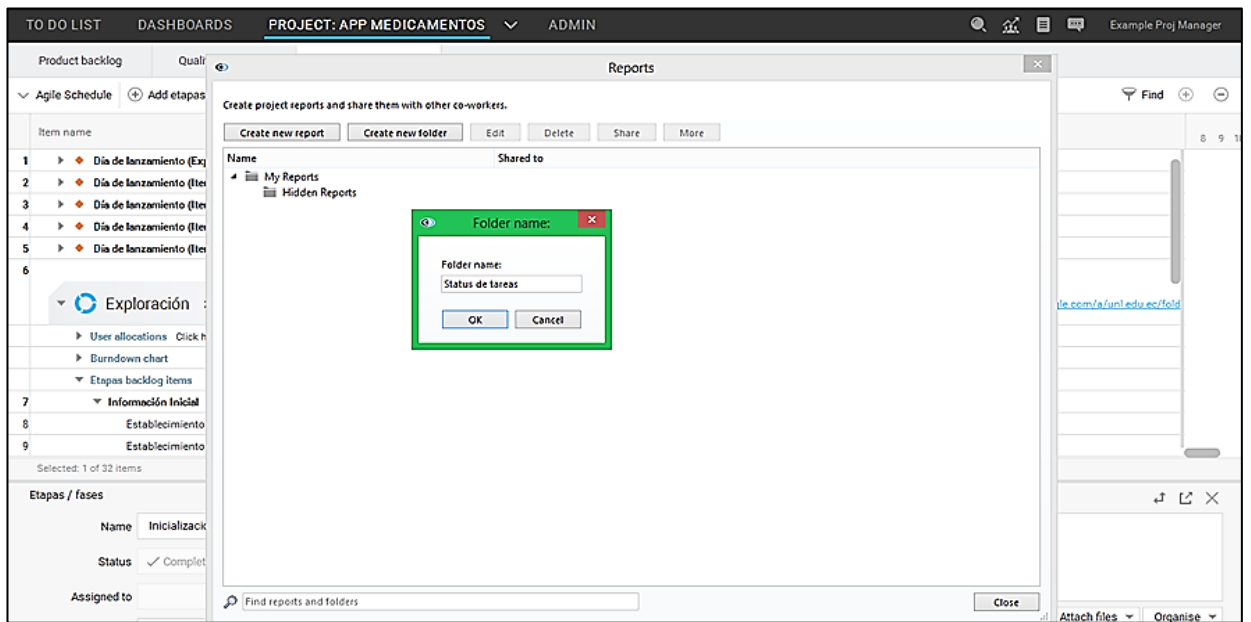


Figura 31: Creación de carpetas.

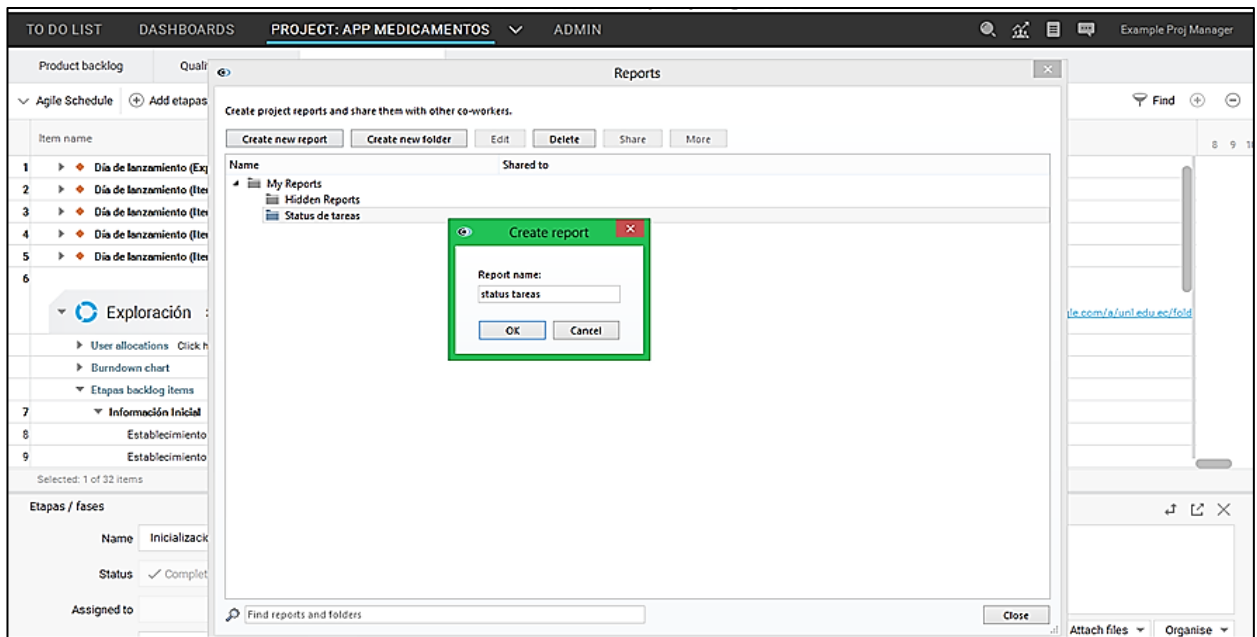


Figura 32: Creación de reportes.

Una vez creada la carpeta y el nuevo reporte, aparecerá una nueva ventana en donde se seleccionará el orden, lo que se desea visualizar y finalmente se guarda (**Save**) el reporte creado como se observa en la *Figura 33*.

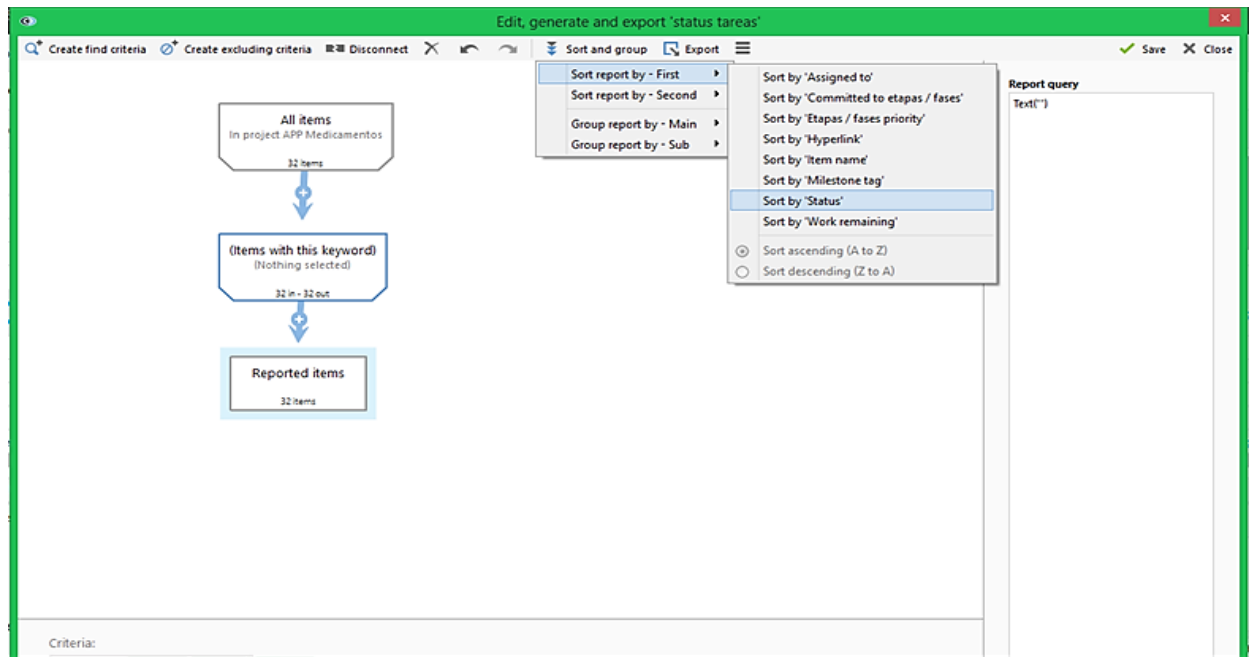


Figura 33: Visualización y guardado del reporte.

3.1.5 VISUALIZACIÓN

Board

Es un tablero de tarjetas, mismas que indican el status de cada tarea de una determinada etapa del proyecto, estas pueden moverse y colocarse en cualquier estado, actualizando automáticamente la interfaz principal del proyecto. La visualización se puede realizar según la etapa seleccionada como lo muestra la *Figura 34*.

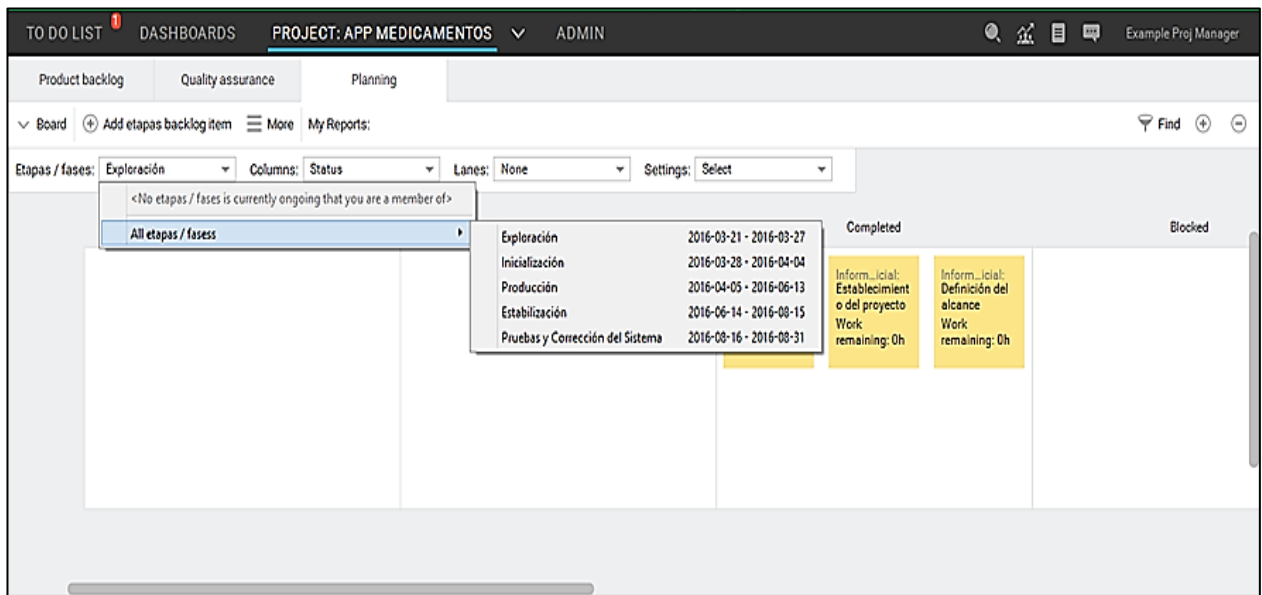


Figura 34: Visualización del status de cada tarea.

Etapas

La visualización también puede realizarse por etapas como en la *Figura 35*, en las cuales se puede observar su status, a que miembro del equipo de trabajo han si asignadas, incluyendo sus respectivos releases.

	Etapas / fases priority	Assigned to	Status	Work remaining	Hyperlink	Milestone tag
Agile Schedule Board No reports created or shared to you 2016-08-15						
1:	Configuraciones Iniciales : Día de planificación (Iteración 2)	High priority	Christian	Completed	0h	Día de lanzamiento (Iteración 2) 2016-08-15
1:	Configuraciones Iniciales : Día de trabajo (Iteración 2)	High priority	Christian	Completed	0h	Día de lanzamiento (Iteración 2) 2016-08-15
1:	Configuraciones Iniciales : Documentación de recapitulación (Iteración 2)	Medium priority	Christian	Completed	0h	Día de lanzamiento (Iteración 2) 2016-08-15
Pruebas y Corrección del Sistema 2016-08-16 - 2016-08-31						
2:	Configuraciones Iniciales : Prueba del sistema	Very high priority	Christian	In progress	(Not set)	Día de lanzamiento (Iteración 3) 2016-08-31
2:	Configuraciones Iniciales : Día de planificación (Iteración 3)	Very high priority	Christian	In progress	(Not set)	Día de lanzamiento (Iteración 3) 2016-08-31
2:	Configuraciones Iniciales : Día de trabajo (Iteración 3)	Very high priority	Christian	In progress	(Not set)	Día de lanzamiento (Iteración 3) 2016-08-31

Figura 35: Visualización por etapas.

Historias

La historia permite almacenar los “estados” o cambios q se realizan en el proyecto, es decir que se puede regresar a un estado anterior que se haya considerado importante en el proyecto. Se pueden crear en la sección **More – Project history – Save a snapshot in the project history** de la *Figura 36* y de igual manera pueden ser accedidas mediante su registro de fechas y nombre asignado en fechas posteriores como en la *Figura 37*.

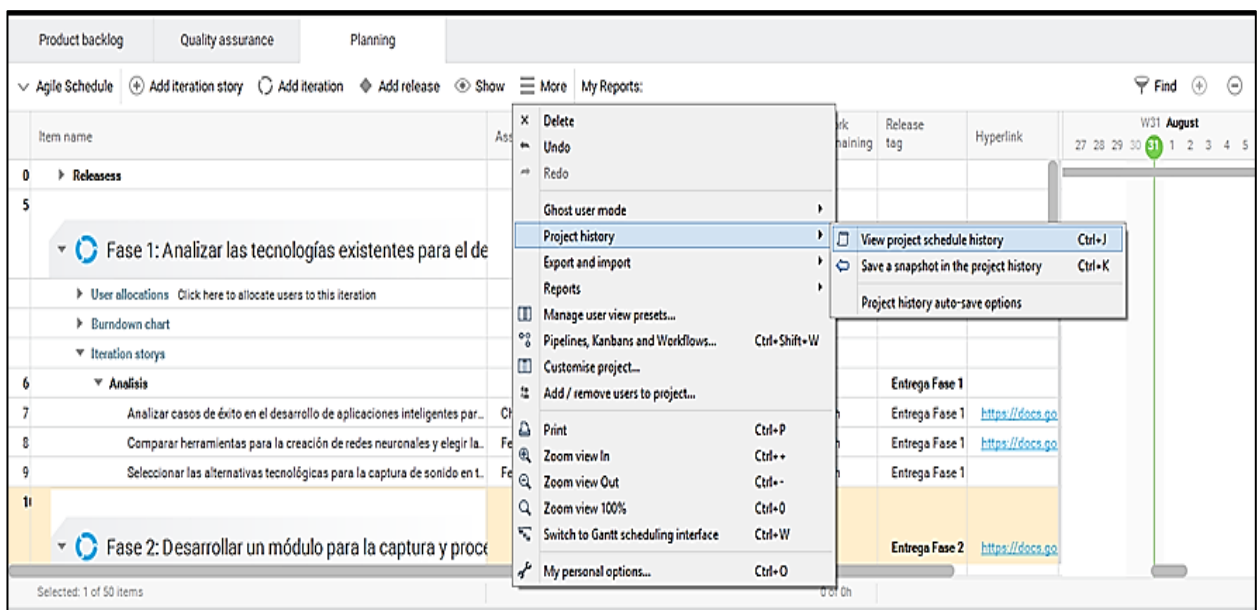


Figura 36: Creación de historias del proyecto.

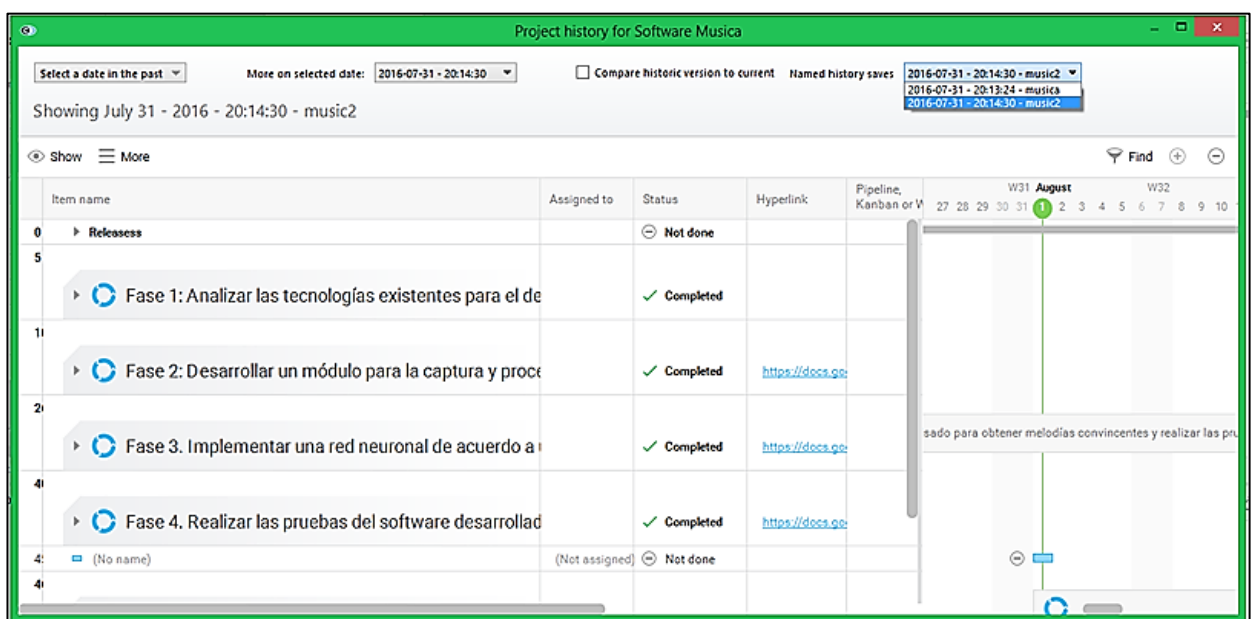


Figura 37: Visualización de una historia del proyecto.

Find

Es una forma de encontrar y visualizar etapas, tareas, workflows y releases del proyecto, según se seleccionen en base a su status tal como lo muestra la *Figura 38*.

Item name	Status	Milestone tag	Hyperlink
1 Releases: Día de lanzamiento (Exploración)	Completed		
2 Releases: Día de lanzamiento (Iteración 0)	Completed		
3 Releases: Día de lanzamiento (Iteración 1)	Completed		
4 Releases: Día de lanzamiento (Iteración 2)	Completed		
7 Exploración : Información Inicial			
8 Información Inicial: Establecimiento de stakeholders			
9 Información Inicial: Establecimiento del proyecto			
11 Información Inicial: Definición del alcance			
1 Realización de ingeniería de requerimientos	Completed		https
1: Inicialización: Configuraciones Iniciales	Completed		
1- Configuraciones Iniciales : Puesta a punto del proyecto	Completed	0h	

Figura 38: Visualización del proyecto en base al status.

Gantt Scheduling

Se puede acceder a esta opción a través de la sección **More – Switch to Gantt scheduling interface** de la *Figura 39*, misma que permitirá la visualización y creación de tareas programadas del proyecto, así como la presentación de fechas y duración en días de las etapas como puede visualizarse en la *Figura 40*. Una vez que se haya culminado con el trabajo en esta sección se puede volver a la interfaz principal mediante los pasos antes indicados.

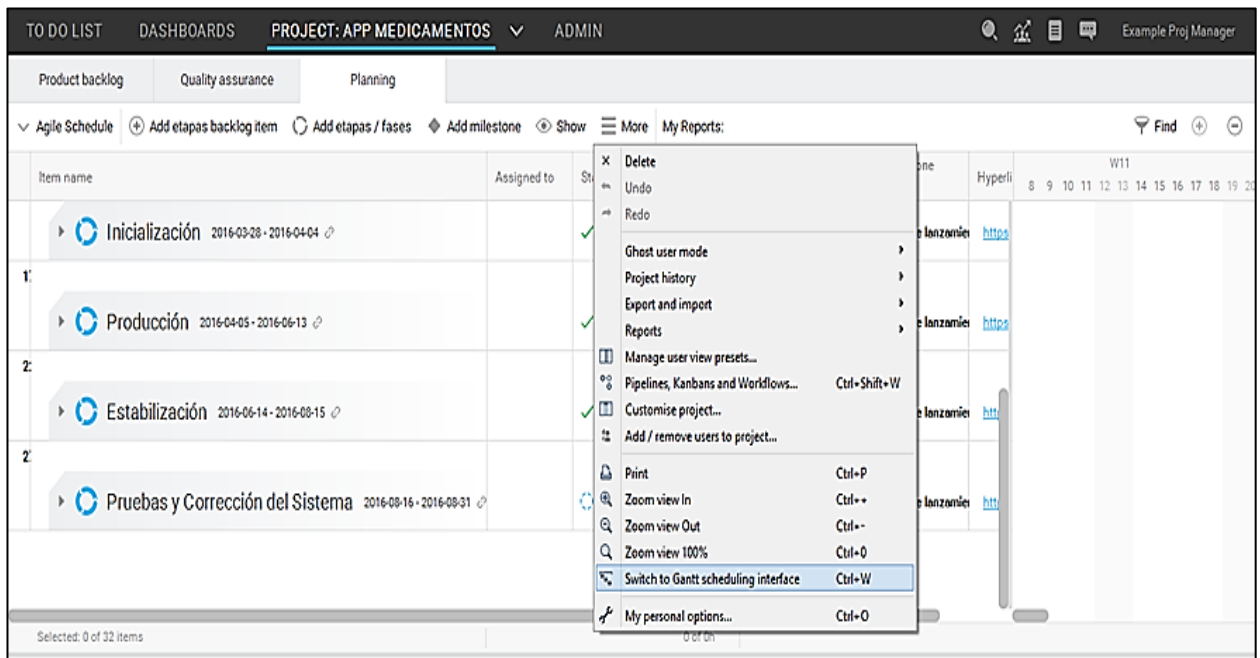


Figura 39: Creación de tareas programadas.

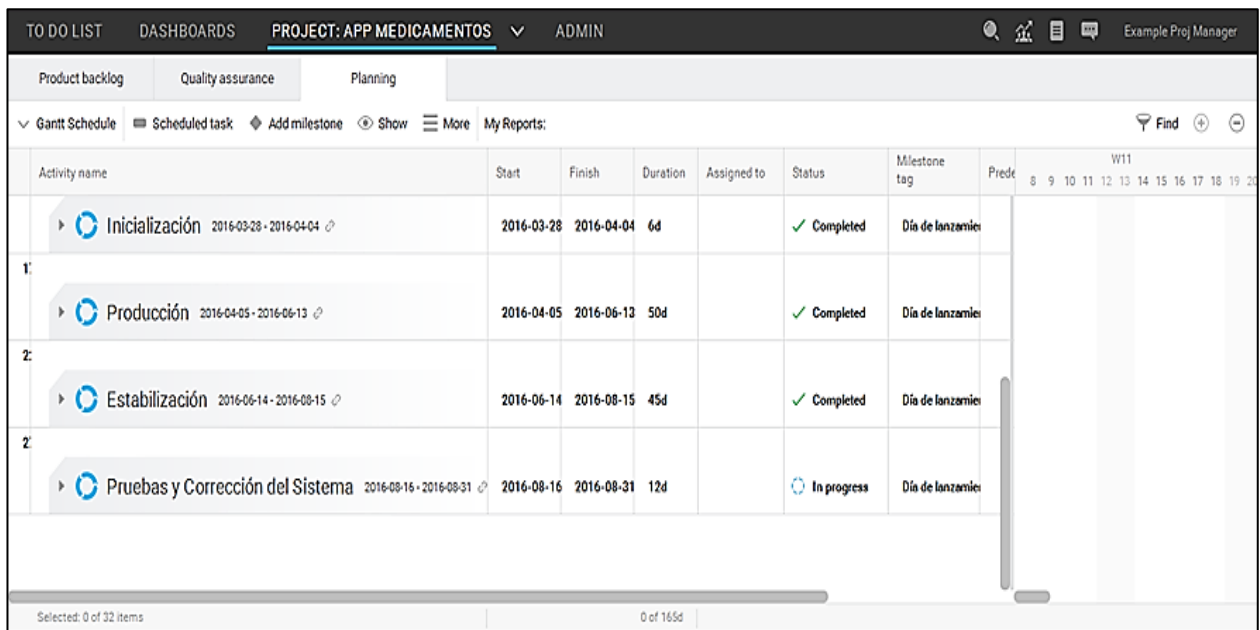


Figura 40: Visualización de la duración del proyecto.

3.1.6 PIPELINES, KANBANS Y WORKFLOWS

Permite la realización de workflows (flujo de trabajo) del proyecto. Para crearlos se debe acceder a la sección **More – Pipelines, Kanbans and Workflows...** de la *Figura 41* donde se desplegara una ventana que permitirá realizar los mismos.

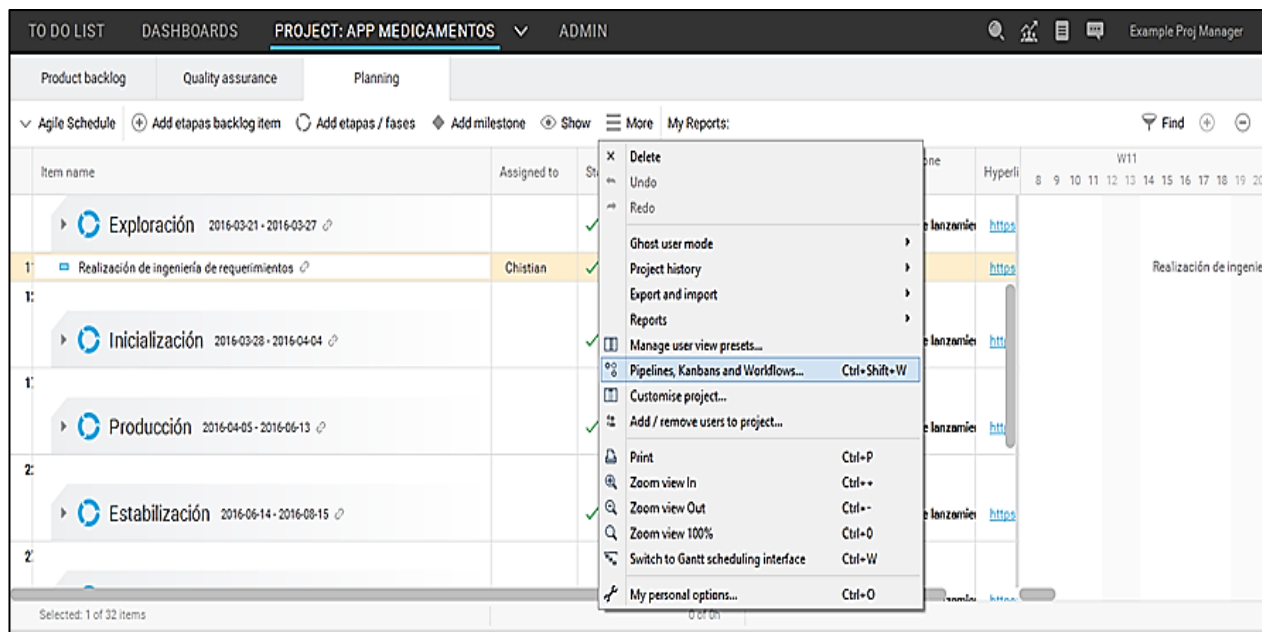


Figura 41: Creación de workflows.

Una vez en la ventana se tiene la opción de crear Workflows o pipeline/Kanbans asignándoles un ícono como en la *Figura 42*.

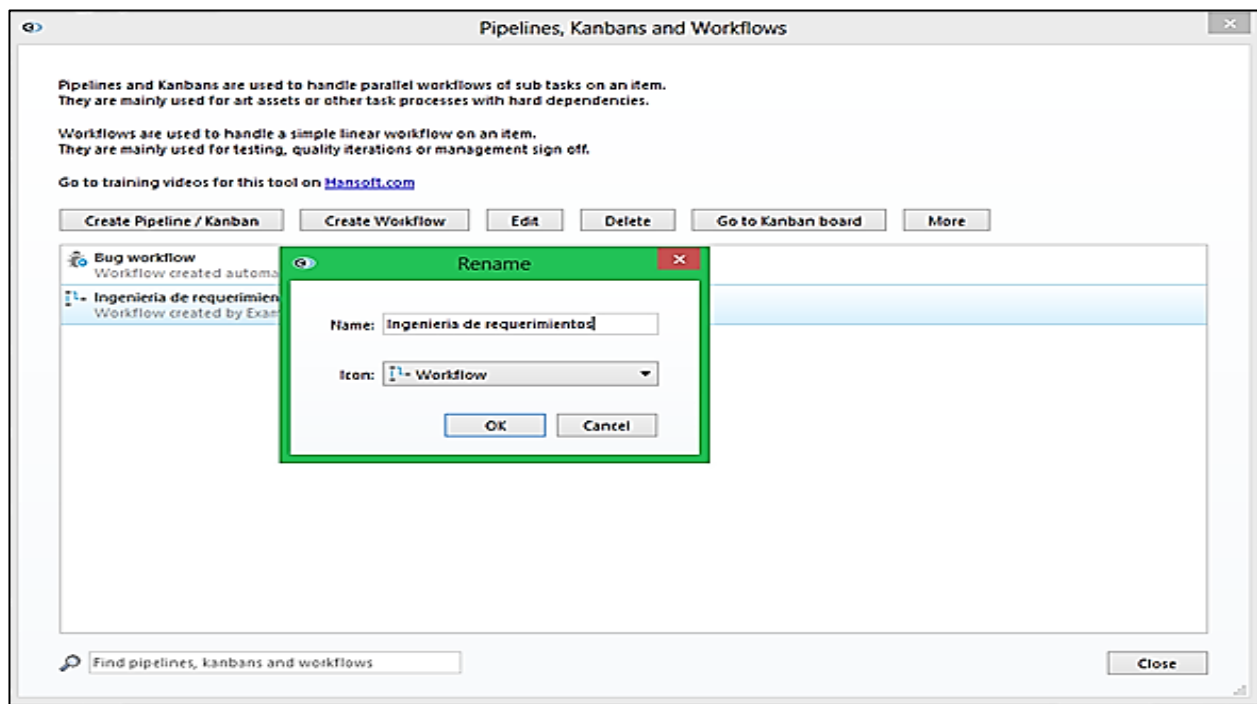


Figura 42: Selección del tipo de gráfica a realizar.

Para crear un Workflow como el que muestra la *Figura 43* se tienen algunas opciones como ingresar el nombre y status, mismo que puede ser editado, asignando miembros del proyecto encargados, opciones u otros status o transiciones en caso de realizarse o no con éxito. También pueden incluirse transiciones mismas que indican la toma de una decisión y relacionarlas con el resto de status o transiciones adicionales mediante el uso de flechas. En sí la creación y asociaciones que se realicen en un Workflow depende del tipo trabajo que se quiera representar, ya que es muy similar a la representación de un diagrama de flujo.

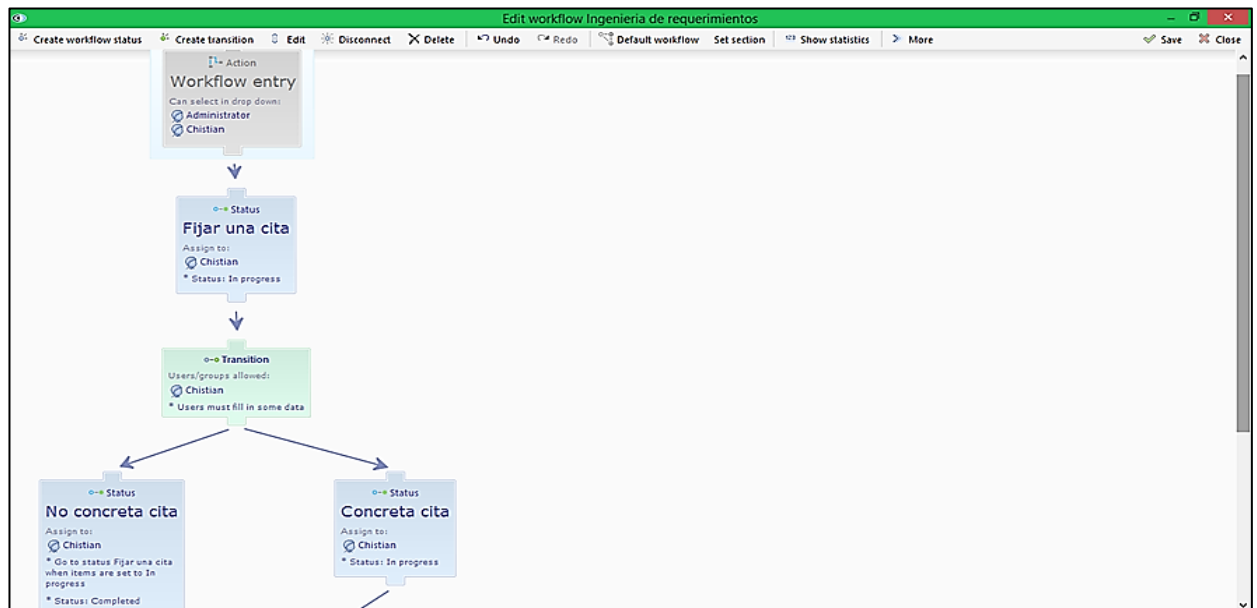


Figura 43: Visualización del workflow terminado.

3.1.7 INFORMES

La sección de informe presenta algunas opciones de gran utilidad. Como en este caso visualizar e imprimir en formato PDF el conjunto de etapas y releases con sus respectivos status. Esto se puede realizar en la sección **More – Print** de la *Figura 44*, que automáticamente llevara a la ventana que se observa a continuación:

Item Name	Assigned to	Status	Work remaining	Start	End	Items
APP Medicamentos	Christian	Completed				
Exploración	Christian	Completed				
Inicialización	Christian	Completed				
Producción	Christian	Completed				
Estabilización	Christian	Completed				
Pruebas y Corrección del Sistema	Christian	Completed				

Figura 44: Visualización del informe.

Otra forma de realizar informes es exportando o importando documentos en diferentes formatos con extensión XML, .xlsx o .text, mediante la sección **More – Export and import** en la *Figura 45* de Hansoft.

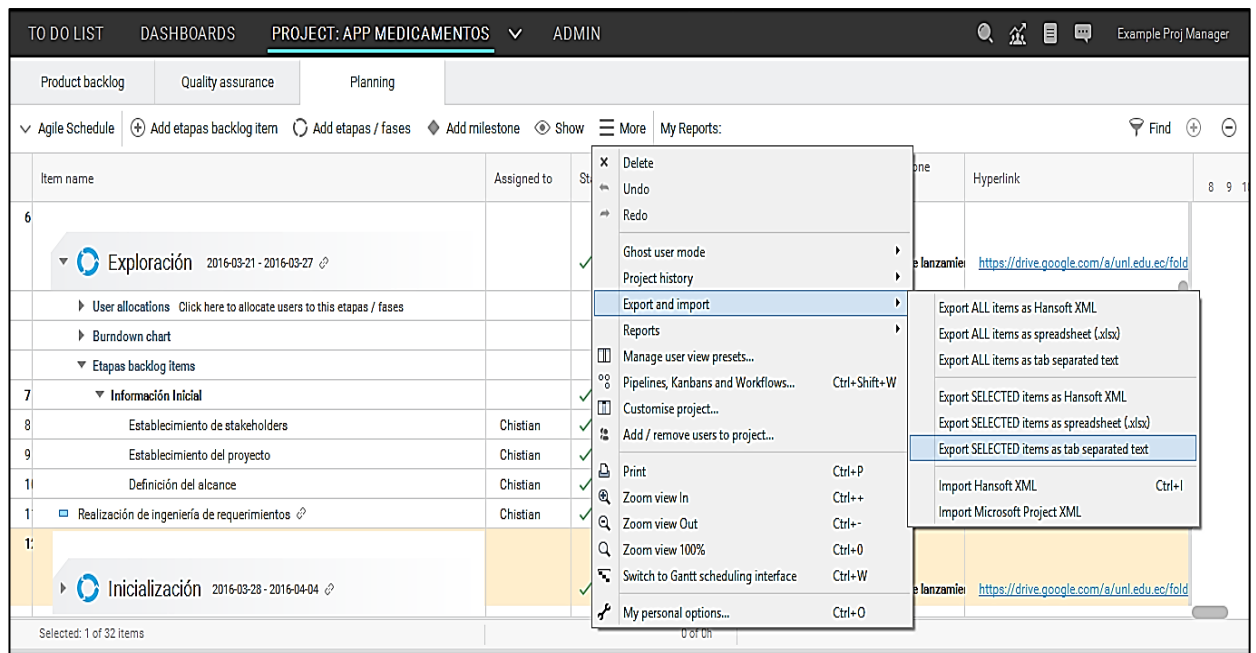


Figura 45: Diferentes extensiones para realizar un informe.

3.2 QUALITY ASSURANCE / ASEGURAMIENTO DE LA CALIDAD

En esta sección se ingresan los bugs del proyecto mediante la sección **Report new bug** de la *Figura 46*, la cual agregará una celda donde se ingresará el nombre del bug, miembro del equipo de trabajo a quien será asignado, el status, el flujo de trabajo, la fase a la que pertenece y la severidad del mismo.

ID	Description	Assigned to	Status	Committed to iteration	Severity ↓
5	Presentación de resultados	Christian	Completed	Fase 4. Realizar	Severity C
1	Entrenamiento de la red Neuronal	Christian	Completed	Fase 3. Implem	Severity A
2	Captura de Audio	Christian	Completed	Fase 2: Desarroll	Severity A
4	Procesamiento de Sonido	Christian	Completed	Fase 2: Desarroll	Severity A

Figura 46: Bugs del proyecto.

3.3 PRODUCT BACKLOG

En esta sección se ingresan los requerimientos funcionales, mediante la opción **Add backlog ítem**, y se ingresará el nombre, el status, la prioridad, la etapa / fase, y releases a la que pertenece y finalmente la estimación en días para su culminación como en la *Figura 47*. Adicionalmente se puede añadir un comentario acerca del mismo.

The screenshot displays the 'Product backlog for APP Medicamentos' interface. At the top, there is a navigation bar with options: 'Hierarchy', 'Add backlog item', 'Show', 'More', and 'My Reports:'. A search icon and 'Find' button are also present. Below this is a table with the following columns: 'Item name', 'Status', 'Product backlog priority', 'Committed to et / fases', 'Milestone tag', and 'Estimated days'. The table contains five rows of items, all with a status of 'Completed'. The first row is highlighted in orange. Below the table, a summary bar indicates 'Selected: 1 of 5 items' and '0.0 of 0.0d'. The bottom section is divided into two panels: 'Backlog item' and 'Comments'. The 'Backlog item' panel shows the details for the selected item: Name 'Realizar la búsqueda del medicamer', Status 'Completed', and Priority 'Very high prio'. The 'Comments' panel has a text area for 'Add a comment...' and buttons for 'Attach files' and 'Organise'.

Item name	Status	Product backlog priority	Committed to et / fases	Milestone tag	Estimated days
Realizar la búsqueda del medicamento por nombre, marca o categoría. ↗	✓ Completed	Very high priority	Producción 2016	Día de lanzamien	0.0d
Mostrar información del medicamento ↗	✓ Completed	Very high priority	Producción 2016	Día de lanzamien	0.0d
Mostrar información de la farmacia que tiene en stock el medicamento ↗	✓ Completed	Very high priority	Producción 2016	Día de lanzamien	0.0d
Mostrar el medicamento genérico ↗	✓ Completed	Very high priority	Producción 2016	Día de lanzamien	0.0d
Crear proformas de medicamentos ↗	✓ Completed	High priority	Producción 2016	Día de lanzamien	0.0d

Figura 47: Ingreso de requisitos funcionales.

4. DASHBOARDS / TABLERO DE INSTRUMENTOS

Esta sección permite crear diferentes gráficos estadísticos en base a los datos que sean seleccionados del proyecto.

4.1 CREAR UN NUEVO GRÁFICO

Para crear un nuevo gráfico Hansoft ofrece algunas opciones, primeramente hay que seleccionar el símbolo + que se encuentra en la parte superior derecha, misma que permitirá crear un nuevo conjunto de gráficos en una “hoja en blanco” tal como se aprecia en la *Figura 48*.

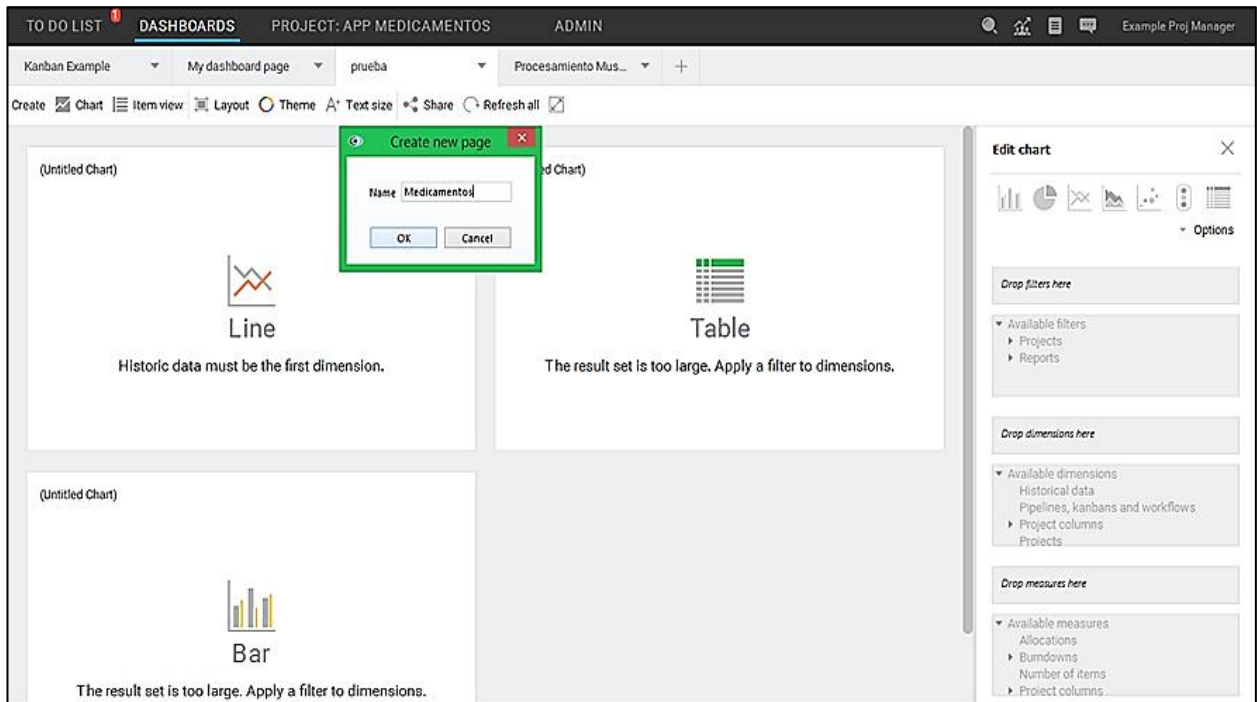


Figura 48: Creación de un gráfico.

Seguidamente en la sección **Chart** se ingresará el nuevo gráfico, luego en la sección **Edit chart** ubicada en la parte derecha se selecciona el tipo de gráfico, proyecto y datos que se desea que aparezcan en el mismo, así como los colores y la visualización de los gráficos creados tal como lo muestra la *Figura 49*.

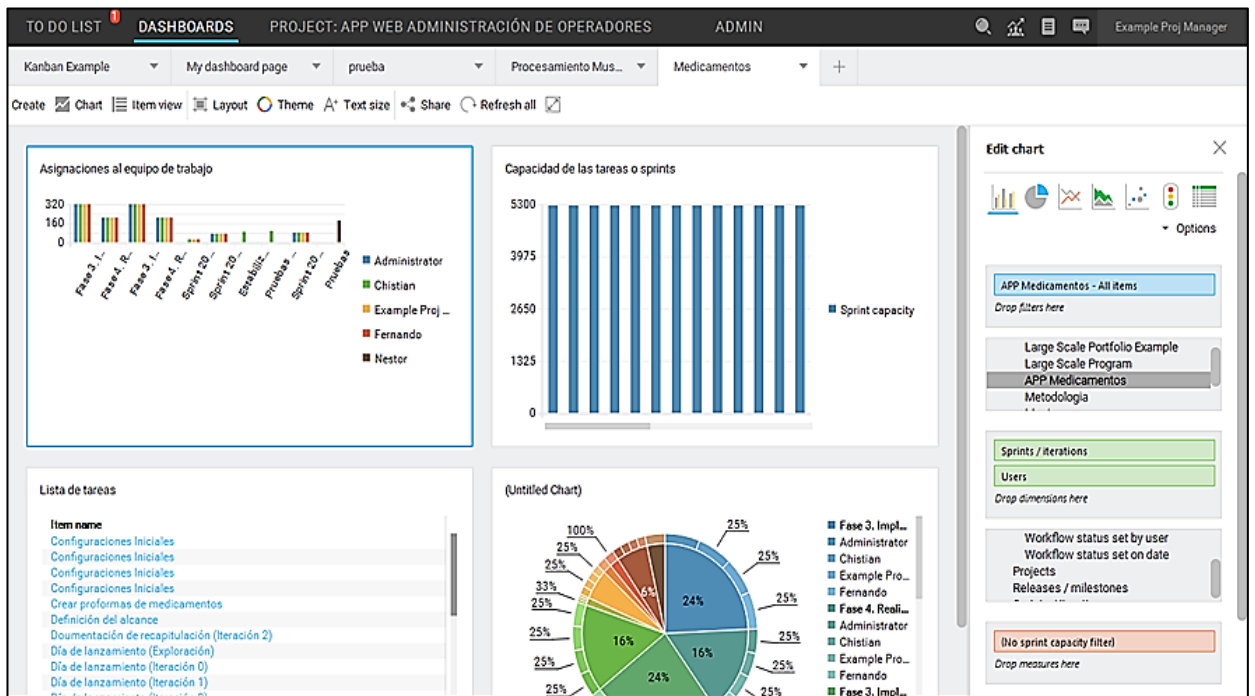


Figura 49: Visualización de los gráficos generados.

5. TO DO LIST / LISTA DE QUEHACERES

Permite visualizar los bugs pendientes a resolver a nivel de todos los proyectos como en la Figura 50.

Description	Status	Sort on	Work remaining / Timeline
<p>Prueba</p> <p>▼ Bugs Status ▼</p> <p> Error al ingresar Assigned Assign...</p> <p>Tutorial Project</p> <p>▼ Sprint Sprint: Change my name by double click on this text Sprint priority • 35h / 2015-07-31</p> <p> Assign each item to a user and estimate the work remaining from now to finish ⊖ Not done 10h</p> <p> The work assigned to you, is updated from your todo list in the top left corner ⊖ Not done 25h</p>			

Item settings ✕

Name

Status

Assigned to

Work remaining

Comments

Figura 50: Visualización de bugs pendientes.

6. PORTFOLIO FIND / BÚSQUEDA DE CARTERA

Al seleccionar esta opción se puede visualizar todos los proyectos creados mediante una selección previa, ya sea esta por sprints, releases, status, etc, tal como lo muestra la *Figura 51*.

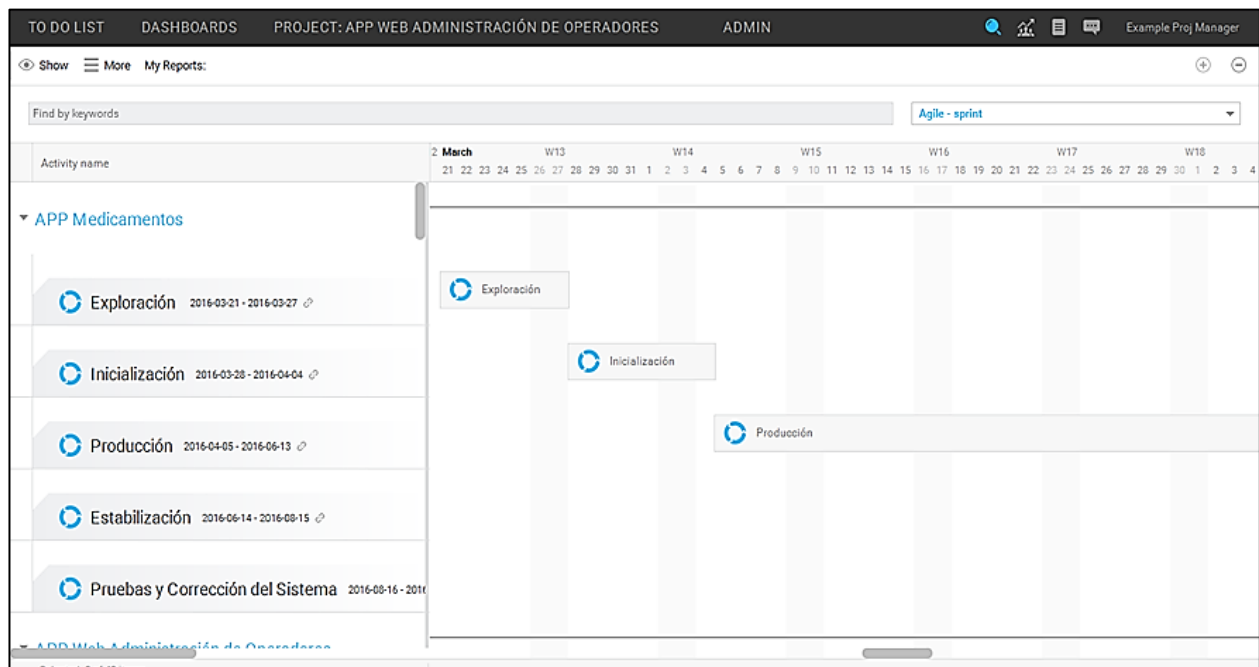


Figura 51: Visualización de todos los proyectos creados.

7. PORTFOLIO ALLOCATIONS / ASIGNACIONES DE CARTERA

Esta opción en la *Figura 52* permite visualizar el trabajo realizado por cada uno de los miembros de los diferentes proyectos con respecto a los tiempos establecidos en cada uno de ellos.

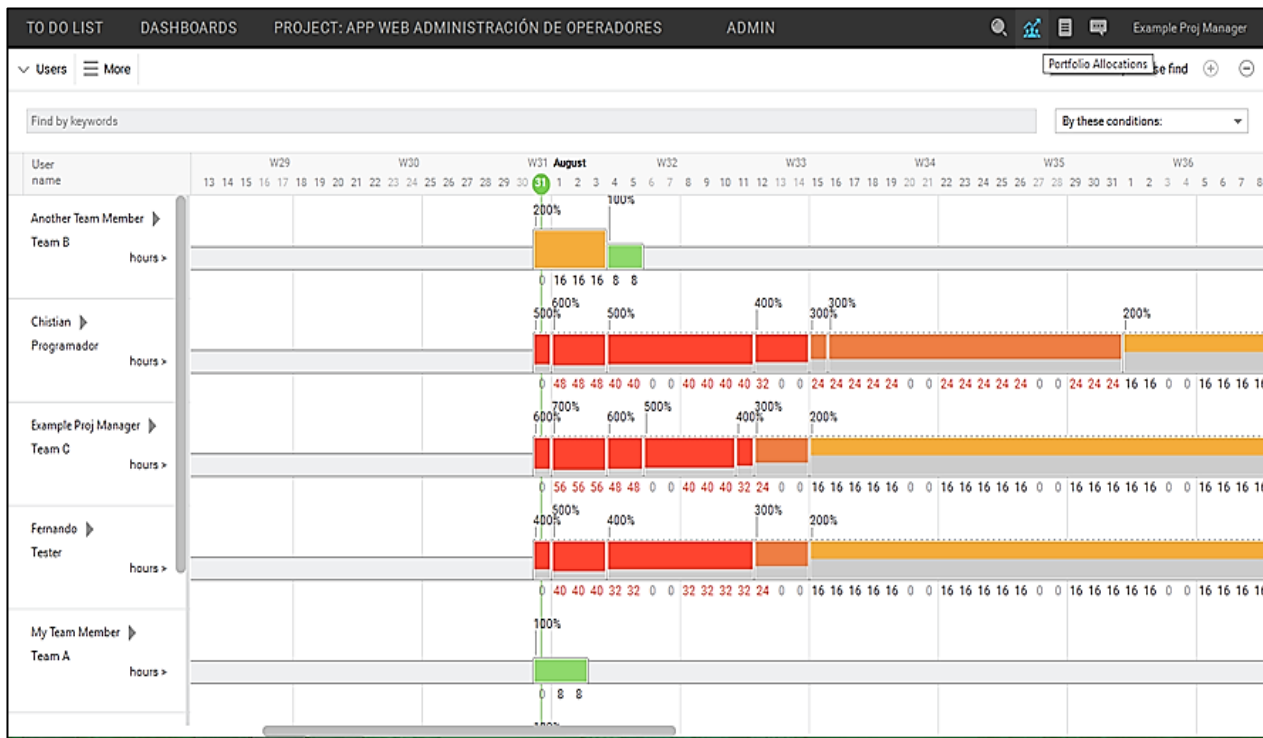


Figura 52: Visualización de los trabajos realizados por los miembros de los proyectos.

8. DOCUMENTS / DOCUMENTOS

Se pueden ingresar documentos según el proyecto seleccionado, haciendo click derecho aparecerán las opciones para crear carpetas e ingresar los documentos como en la *Figura 53*.

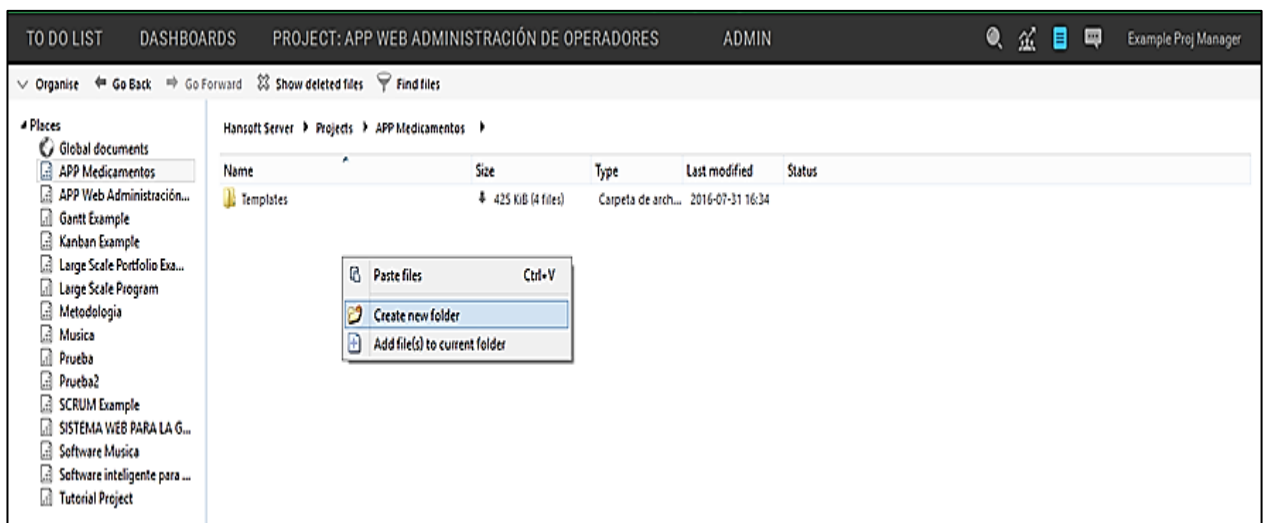


Figura 53: Creación de una carpeta.

Una vez creada la carpeta, se accede a ella e ingresan los documentos a la misma, como en este caso son 4 los documentos a ingresarse en la carpeta Templates y se digita un comentario como muestra la *Figura 54*.

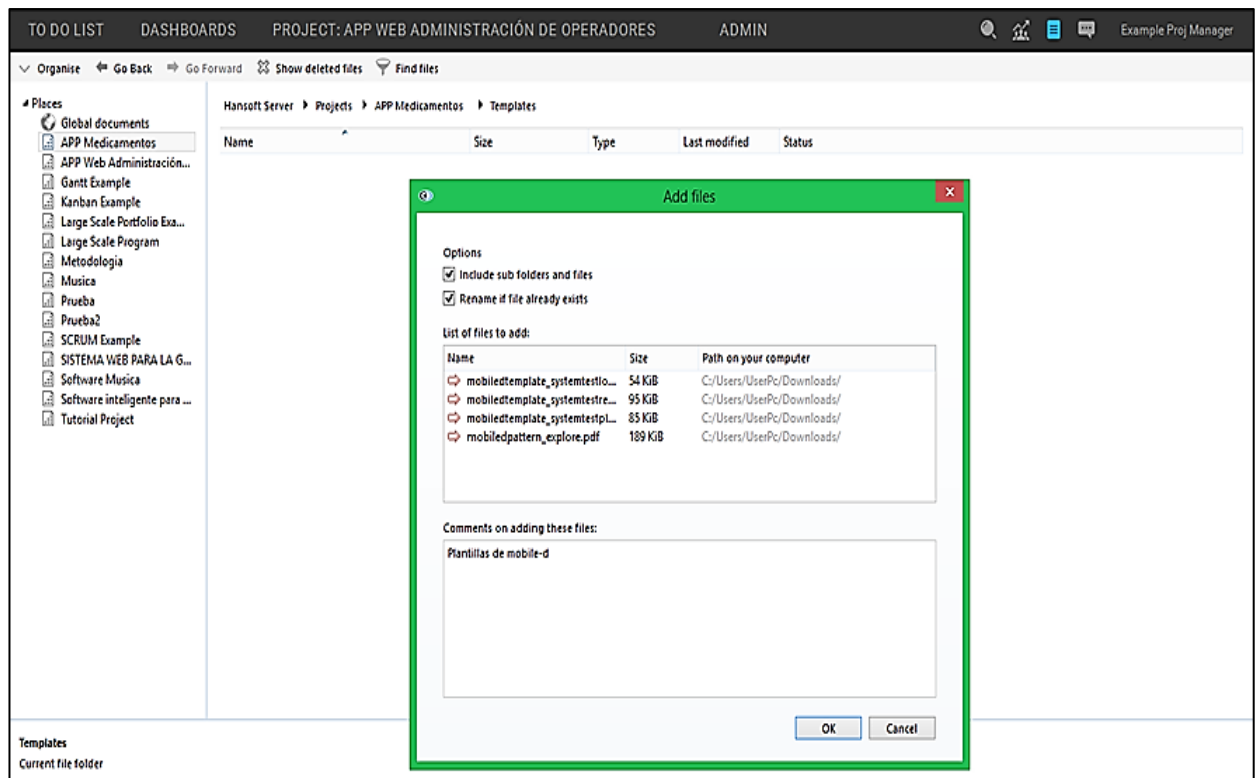


Figura 54: Ingreso de documentos.

9. CHAT

Hansoft también cuenta con el servicio de chat entre los miembros de los proyectos, en este caso basta con seleccionar el icono de mensaje y se desplegará una ventana para seleccionar el usuario con quien se desea iniciar el chat tal como lo muestra la *Figura 55*, cabe destacar que pueden abrirse varias ventanas a la vez como en la *Figura 56*.

TO DO LIST DASHBOARDS PROJECT: APP MEDICAMENTOS ADMIN Example Proj Manager

Product backlog Quality assurance Planning

Agile Schedule Add etapas backlog item Add etapas / fases Add milestone Show More My Reports: Status de tareas

Item name	Assigned to	Status	Work remaining	Etapas / fases priority	Milestone tag
0 Releases		Not done			
1 Día de lanzamiento (Exploración)		Completed			
2 Día de lanzamiento (Iteración 0)		Completed			
3 Día de lanzamiento (Iteración 1)		Completed			
4 Día de lanzamiento (Iteración 2)		Completed			
5 Día de lanzamiento (Iteración 3)		Not done			
6 Exploración 2016-03-21 - 2016-03-27		Completed			Día de lanzamiento
<ul style="list-style-type: none"> User allocations Click here to allocate users to this etapas / fases Burndown chart Etapas backlog items 					
7 Información Inicial		Completed			Día de lanzamiento
8 Establecimiento de stakeholders	Christian	Completed	0h	Medium priority	Día de lanzamiento

Selected: 1 of 32 items 0 of 0h

Do not disturb
 Groups
 Team C
 Christian
 Fernando
 My Team Member
 Nestor

Figura 55: Visualización de miembros del equipo.

TO DO LIST DASHBOARDS PROJECT: APP MEDICAMENTOS ADMIN Example Proj Manager

Product backlog Quality assurance Planning

Agile Schedule Add etapas backlog item Add etapas / fases Add milestone Show More My Reports: Status de tareas Find

Item name	Assigned to	Status	Work remaining	Etapas / fases priority	Milestone tag
0 Releases		Not done			
1 Día de lanzamiento (Exploración)		Completed			
2 Día de lanzamiento (Iteración 0)		Completed			
3 Día de lanzamiento (Iteración 1)		Completed			
4 Día de lanzamiento (Iteración 2)		Completed			
5 Día de lanzamiento (Iteración 3)		Not done			
6 Exploración 2016-03-21 - 2016-03-27		Completed			Día de lanzamiento
<ul style="list-style-type: none"> User allocations Click here to allocate users to this etapas / fases Burndown chart Etapas backlog items 					
7 Información Inicial		Completed			Día de lanzamiento
8 Establecimiento de stakeholders	Christian	Completed	0h	Medium priority	Día de lanzamiento

Selected: 1 of 32 items

Etapas / fases
 Name: Inicialización
 Status: Completed
 Assigned to: More...

Comments
 Add a comment...

Nestor
 2016-07-27
 hola
 Example Proj Manager dale

Fernando
 2016-07-27
 hola
 Example Proj Manager

Christian
 2016-07-31
 hola
 Example Proj Manager

Figura 56: Visualización de chats con los diferente miembros del equipo.

4.8. Difusión

Para llevar a cabo la difusión de la herramienta Hansoft ALM se creó una página web, cuya url es: <https://hansoftunl.000webhostapp.com> y contiene la siguiente información:

- Categorías
- Características
- Funciones
- Video explicativo
- Guía de la herramienta en PDF

La página presenta información básica de Hansoft ALM. En la *Figura 57* consta el nombre del Proyecto de Titulación, del cual se originó la propuesta.

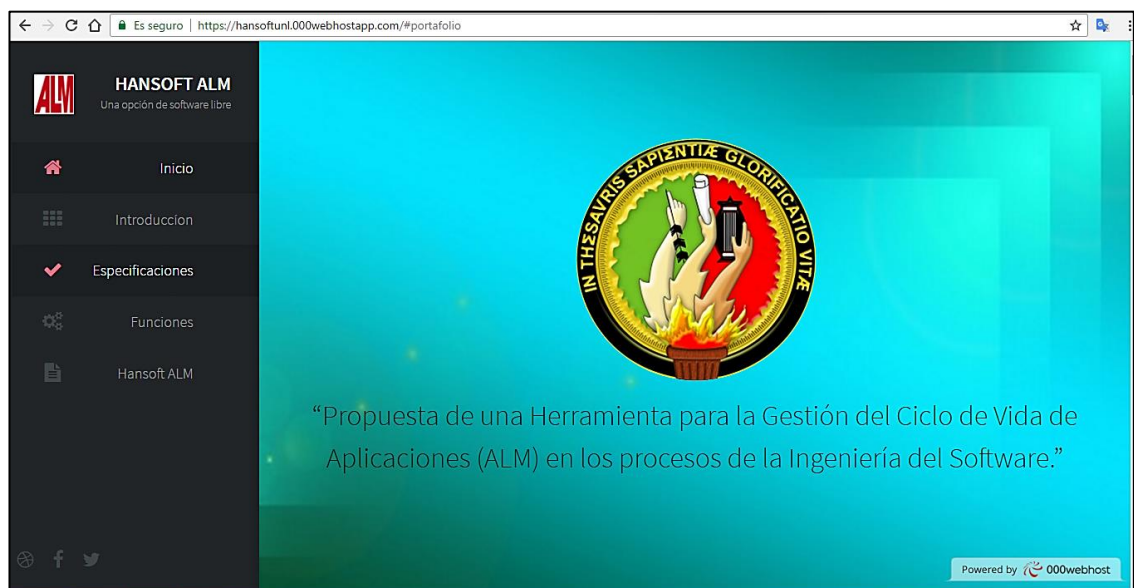


Figura 57: Página de difusión de Hansoft ALM.

Se puede apreciar una breve introducción de lo que son las herramientas ALM y como están conformadas en la *Figura 58*.

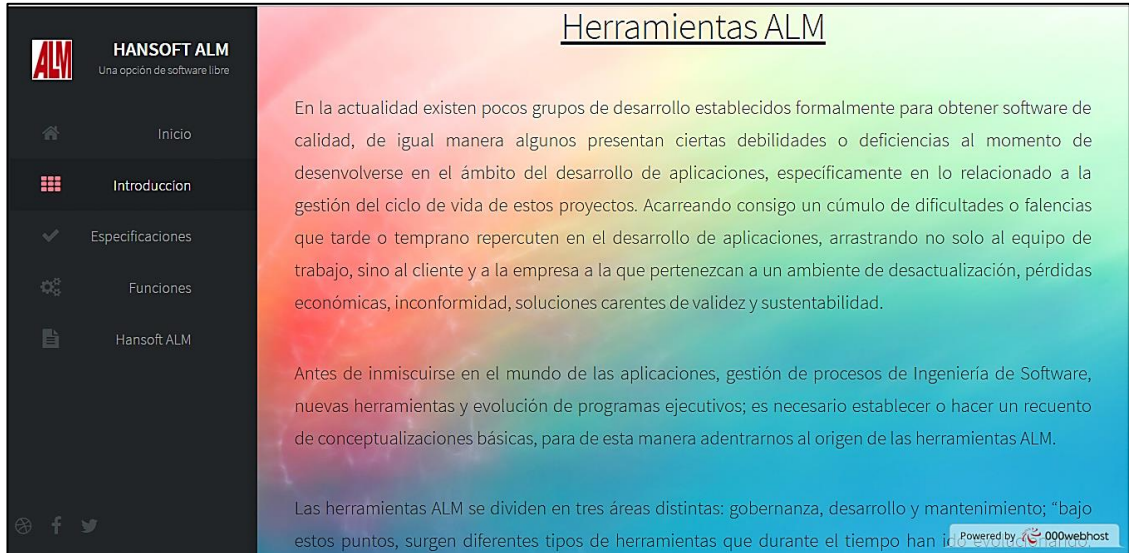


Figura 58: Introducción referente a herramientas ALM.

La página consta de una sección de especificaciones de la herramienta, como puede apreciarse en la *Figura 59*, donde se detallan parte de sus características, categorías y plataformas.



Figura 59: Especificaciones de Hansoft ALM.

La sección de funciones de la *Figura 60*, contiene información relacionada con la instalación, el proyecto y demás funcionalidades ofrecidas por Hansoft ALM.



Figura 60: Funciones de Hansoft ALM.

La página también consta con un video informativo de Hansoft ALM, como puede apreciarse en la *Figura 61*.



Figura 61: Video explicativo de Hansoft ALM.

Adicionalmente se puede descargar la Guía de la herramienta, tal como lo muestra la *Figura 62*. Esta guía se encuentra en PDF para su visualización y posterior descarga.



Figura 62: Guía de Hansoft ALM.

g. DISCUSIÓN

El presente trabajo de titulación denominado “Propuesta de una Herramienta para la Gestión del Ciclo de Vida de Aplicaciones (ALM) en los procesos de la Ingeniería del Software” tuvo como propósito final la presentación de una herramienta ALM (open source) como propuesta, para su utilización en la Escuela de Ingeniería en Sistemas de la Universidad Nacional de Loja. Esta se realizó mediante el cumplimiento de cada uno de los siguientes objetivos específicos:

- **Objetivo específico 1:** Analizar las herramientas ALM difundidas en la actualidad y establecer la especificación de evaluación.

Para efectuar el análisis se realizó la búsqueda de aquellas herramientas libres y privadas difundidas en la web utilizando el Método de Búsqueda de Información Secuencial. Se consideró este método por encima de los demás ya que facilita la identificación de información deseada y no limita a un estilo determinado para la misma.

De esta información obtenida se clasificó a todas las herramientas encontradas en: Libres, Privadas y Otras (véase Tabla I y Tabla II). Se efectuó este tipo de clasificación debido a que la institución de educación superior está en la obligación de usar software libre (véase la sección 3.12 de los Resultados), lo cual permitió su fácil identificación.

En lo concerniente a la especificación de evaluación se realizó una entrevista a las empresas de desarrollo de la ciudad (véase Anexo 2, 3, 4 y 5) debido a la vasta experiencia en desarrollo de las mismas, frente a las diferentes universidades de la localidad ya que la mayoría de docentes y alumnos no poseen la experiencia suficiente como para aportar al correcto desarrollo del Proyecto de Titulación. Obteniendo de esta manera la información que permitió identificar el grado de cumplimiento de las diferentes herramientas ante la especificación de evaluación establecida.

- **Objetivo específico 2:** Evaluar la especificación de herramientas en base a un caso de estudio determinado.

En este objetivo se dio especial énfasis en efectuar la comparativa de herramientas en un caso de estudio, mismo que se obtuvo de los proyectos de titulación ya realizados en la escuela (véase Tabla IV). En esta primera evaluación de herramientas no se trabajó con proyectos en tiempo real debido a que aquellos que estaban en ejecución tardaban demasiado en culminar y de esta evaluación lo que se buscaba era establecer la mejor herramienta del pequeño grupo que ya se tenía. Por lo que se optó por trabajar con aquellos proyectos ya terminados, cuya información estaba lista y permitía determinar con facilidad aquella herramienta que más se ajusta a las necesidades de control y desarrollo de aplicaciones.

Para la realización de la segunda evaluación ya se contaba con la herramienta más adecuada (véase Tabla VIII), lo cual facilitó su aplicación en un nuevo caso de estudio, pero esta vez en tiempo real. Permitiendo conocer y evaluarla más detenidamente, e interactuar con el equipo de desarrollo.

- **Objetivo específico 3:** Establecer la herramienta de Gestión del Ciclo de Vida de Aplicaciones (ALM) para la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja.

Finalmente se procedió a establecer una metodología a parametrizar en la herramienta, mediante la recopilación de información de los proyectos de titulación desarrollados desde los inicios de la carrera, formando así una base de datos que se empleó en una técnica de Inteligencia Empresarial. Dicha técnica permitió realizar la predicción de la metodología; en este caso Pentaho, frente a otras posibles técnicas aportó al desarrollo del trabajo con su combinación de herramientas, componentes y fácil interpretación de resultados. De los cuales se obtuvo la metodología UWE (metodología resultante, véase Tabla XI) en la herramienta Hansoft ALM.

Demostrando así que Hansoft ALM es la herramienta open source más idónea para su uso en la Escuela de Ingeniería en Sistemas, debido a su adaptabilidad, características de control y comunicación entre el equipo de desarrollo. Permitiendo inmiscuir y dar a conocer a los alumnos un mundo de soluciones tecnológicas y de apoyo al desarrollo de aplicaciones, donde la organización,

supervisión y el apoyo en el equipo de trabajo se unen para obtener un producto software de calidad.

h. CONCLUSIONES

- Un adecuado análisis de las Herramientas ALM va íntimamente ligado a la selección de la técnica de búsqueda que se ajuste a las necesidades de obtención de información, para de esta manera dar validez y sustentabilidad al mismo.
- Establecer de forma correcta la especificación de evaluación de un conjunto de herramientas (véase Tabla VIII), permitió cubrir la mayor parte de necesidades de los desarrolladores al momento de optar por una solución tecnológica ofreciendo resultados valederos.
- La evaluación en base al caso de estudio permitió establecer un puente entre la teoría y la práctica, fomentando el desarrollo de un juicio crítico que facilita distinguir la mejor herramienta, como en este caso lo es Hansoft ALM.
- Hansoft ALM ha demostrado ser la herramienta apropiada para los estudiantes de la Escuela de Ingeniería en Sistemas debido a su gestión en el desarrollo de software, facilidad de parametrización, uso y comunicación; así como como una adecuada organización, corrección de errores y control que aportará a la obtención de productos software de calidad, fomentando el trabajo en equipo; mismo que se ve afectado o en la mayoría de los casos representa un “reto” ante alumnos y docentes.

i. RECOMENDACIONES

- Antes de seleccionar una Herramienta ALM es necesario acudir a sus páginas oficiales, grupos en línea o fuentes fidedignas de información, con la finalidad de conocer el grado de gestión, usabilidad y opiniones de usuarios. Para una vez establecidos sus beneficios y funcionalidades poder contrastarlas con las necesidades de los alumnos o equipo de desarrollo. Evitando de esta manera posibles retrasos en el desarrollo de proyectos al optar por una herramienta cuyo cumplimiento de características sean escasas y surjan dificultades en instancias que pongan en riesgo la culminación exitosa de una aplicación.
- Debido a que las entrevistas realizadas en este Proyecto de Titulación arrojaron como resultado un alto grado de desconocimiento en relación a las herramientas ALM; es recomendable darlas a conocer tanto a docentes como alumnos, e incentivar su uso. Donde la difusión de estas puede apoyarse en la página web creada como parte del desarrollo de este proyecto.
- Ya que la herramienta no permite ingresar el nombre de la metodología; antes de realizar la parametrización de esta en Hansoft ALM es importante tener en claro sus principales fases y requerimientos, con la finalidad de asignar correctamente las características, mismas que facilitarán su posterior identificación, evitando confusiones o errores en el control del proyecto.
- Para trabajos futuros se recomienda aportar a la gestión y control de los procesos de ingeniería del software de una empresa de desarrollo de la ciudad de Loja mediante la implementación de una Herramienta ALM; brindando capacitación a los usuarios y seguimiento de resultados a la gestión del proceso de desarrollo.

j. BIBLIOGRAFÍA

Referencias Bibliográficas

- [1] Ian Sommerville, *Ingeniería de Software*. Madrid, España: PERARSON EDUCACIÓN, 2005.
- [2] Universidad a Distancia de Madrid, «Procesos en ingeniería del software», *Universidad a Distancia de Madrid*, 2016. [En línea]. Disponible en: <http://www.udima.es/es/procesos-ingenieria-software.html>. [Accedido: 06-sep-2016].
- [3] A. G. Julián Pérez Porto, «Definición de ingeniería de software — Definicion.de», *Definición.de*, 2009. [En línea]. Disponible en: <http://definicion.de/ingenieria-de-software/>. [Accedido: 12-mar-2017].
- [4] Comunidad IEBS, «Las Metodologías Ágiles: qué son y cuáles son sus ventajas», *Blog de IEBSchool*, 07-feb-2014. [En línea]. Disponible en: <http://comunidad.iebschool.com/iebs/agile-scrum/que-es-agile/>. [Accedido: 06-sep-2016].
- [5] «CyTA», 2010. [En línea]. Disponible en: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm. [Accedido: 09-may-2016].
- [6] Edison Xavier Benítez Chica, «Implementación Del Servicio Web “Calificación De Jugadores En Línea” Para La Asociación De Fútbol Amateur De Pichincha (AFAP) Utilizando La Metodología Iconix». Escuela Politécnica Nacional, Quito, mar-2014.
- [7] J. D. F. M. A. N. Cadavid y Jonathan Morales Vélez, «Revisión de metodologías ágiles para el desarrollo de software». Universidad Icesi, sep-2013.
- [8] Alexandra Patricia Chicaiza Ayala, «Desarrollo De Software De Nomina De Empleados Utilizando La Metodología Crystal». Escuela Politécnica Del Ejército, Sangolquí, jul-2007.
- [9] L. del C. Ramírez-M y A. S. F. Fuentes, «BUENAS PRÁCTICAS, UNA SOLUCIÓN PARA UN MEJOR DESARROLLO DE SOFTWARE», *Rev. MundoFesc*, vol. 2, n.º 8, pp. 37-45, dic. 2014.
- [10] M. Boeras Velázquez y M. B. Velázquez, «Aplicando el Método de Boehm y Turner», *Ser. Científica-Univ. Las Cienc. Informáticas*, vol. 5, n.º 6, may 2012.
- [11] «Conceptos Y Fundamentos Básicos De Las Tecnologías A Utilizar En El Desarrollo Del Sistema». Repositorio Digital Universidad Técnica de Cotopaxi, 14-dic-2015.
- [12] Teresita de Jesus Cepeda Hernández, «Prácticas del CMM y de las Metodologías ágiles para la Mejora del Proceso de Desarrollo de Software en Organizaciones Pequeñas: Un Estudio Exploratorio-Edición Única». Instituto Tecnológico Y De Estudios Superiores De Monterrey, 12.
- [13] C. J. S. Roberth G. Figueroa, «Metodologías Tradicionales Vs. Metodologías Ágiles». Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.
- [14] Wilson Bissi, «SCRUM - Metodología De Desenvolvimento Ágil». Centro Universitário de Maringá (CESUMAR), Tecnologia em Análise e Desenvolvimento de Sistemas, 6.
- [15] «Application lifecycle management Informatika & Komputer Manual Indonesia theology.asin.web.id». [En línea]. Disponible en: http://theology.asin.web.id/_lain.php?_lain=497. [Accedido: 06-sep-2016].
- [16] «Métodos de Búsqueda de Información – Internet», *Como Hacer un Ensayo*, 24-mar-2015. .
- [17] Atlassian, «JIRA Software - Issue & Project Tracking for Software Teams», *Atlassian*. [En línea]. Disponible en:

- https://es.atlassian.com/software/jira?_mid=2dc4a0239c3f7d97446fb3a0e0a66cbc&aceid=&adposition=1t1&adgroup=20440471102&campaign=325695382&creative=81925778422&device=c&keyword=atlassian%20jira&matchtype=e&network=g&placement=&gclid=Cj0KEQjwj7q6BRDcxfg4pNTQ2NoBEiQAzUpuW3xO6sQSC1p-PZJfQBqmlZqD2snwi-SFn2Zzg5gtq20aAiTe8P8HAQ. [Accedido: 02-jun-2016].
- [18] IBM, «IBM - Rational Collaborative Lifecycle Management». [En línea]. Disponible en: <http://www-03.ibm.com/software/products/en/ratliclm>. [Accedido: 02-jun-2016].
- [19] IBM, «IBM - Rational Team Concert», 2016. [En línea]. Disponible en: <http://www-03.ibm.com/software/products/es/rtc>. [Accedido: 03-jun-2016].
- [20] HPE, «Soluciones y software Application Lifecycle Management (ALM) | Hewlett Packard Enterprise España», 2016. [En línea]. Disponible en: <http://www8.hp.com/es/es/software-solutions/application-lifecycle-management.html>. [Accedido: 03-jun-2016].
- [21] DevCon, «StarTeam», *DEVCON*, 2016. [En línea]. Disponible en: <http://www.devcon.biz/starteam/>. [Accedido: 03-jun-2016].
- [22] SAP, «SAP Solution Manager | SAP Support Portal», 2016. [En línea]. Disponible en: <https://support.sap.com/solution-manager.html>. [Accedido: 03-jun-2016].
- [23] «Enjoy integrated management with TraceCloud's SaaS ALM - Five hot ALM SaaS tools and services», 29-jul-2014. [En línea]. Disponible en: <http://searchsoftwarequality.techtarget.com/photostory/2240225723/Five-hot-ALM-SaaS-tools-and-services/6/Enjoy-integrated-management-with-TraceClouds-SaaS-ALM>. [Accedido: 04-jun-2016].
- [24] TraceCloud, «TraceCloud - SAAS Agile Scrum Requirements Management - Collaborate, Define, Manage and Deliver your Customer Requirements», 2016. [En línea]. Disponible en: <https://www.tracecloud.com/GloreeJava2/jsp/WebSite/TCHome.jsp>. [Accedido: 03-jun-2016].
- [25] Project Management.com, «ProjectManagement.com - TOMOS ALM», 2016. [En línea]. Disponible en: <http://www.projectmanagement.com/tools/248531/TOMOS-ALM>. [Accedido: 21-jul-2016].
- [26] «Introduction — Tuleap latest version documentation», 2016. [En línea]. Disponible en: <http://doc-en.tuleap.net/en/latest/user-guide/intro.html>. [Accedido: 04-may-2016].
- [27] «Libre & Open Source tool for Enterprise Application Lifecycle Management, Software Development, Agile tools, Issue tracking | Tuleap», 2014. [En línea]. Disponible en: <https://www.tuleap.org/about-tuleap/overview>. [Accedido: 04-may-2016].
- [28] «Open Source issue & Project Tracking software, Project Management tool | Tuleap», 2014. [En línea]. Disponible en: <https://www.tuleap.org/about-tuleap/features/project-management>. [Accedido: 04-may-2016].
- [29] «Eclipse Mylyn Open Source Project», 2016. [En línea]. Disponible en: <http://www.eclipse.org/mylyn/>. [Accedido: 05-may-2016].
- [30] «Mylyn | projects.eclipse.org», 2016. [En línea]. Disponible en: <https://projects.eclipse.org/projects/mylyn>. [Accedido: 05-may-2016].
- [31] «Endeavour Agile ALM | Agile Software Development & Application Lifecycle Management», 2011. [En línea]. Disponible en: <http://endeavour-mgmt.sourceforge.net/>. [Accedido: 05-may-2016].
- [32] «Endeavour | PM Software - Ratings, Features, and Specs», 2016. [En línea]. Disponible en: <http://project-management.softwareinsider.com/l/58/Endeavour>. [Accedido: 05-may-2016].

- [33] «Endeavour Agile ALM | Agile Software Development & Application Lifecycle Management», 2011. [En línea]. Disponible en: <http://endeavour-mgmt.sourceforge.net/features1.html>. [Accedido: 05-may-2016].
- [34] «About Us | PolarSys - Open Source solutions for Systems Engineering and Embedded Systems», 2016. [En línea]. Disponible en: <https://www.polarsys.org/about-us>. [Accedido: 05-may-2016].
- [35] «ProjectManagement.com - Hansoft», 2016. [En línea]. Disponible en: <http://www.projectmanagement.com/tools/267963/Hansoft>. [Accedido: 06-may-2016].
- [36] Hansoft, «Hansoft 7 - Seriously Social». 2012.
- [37] «Feature List – Hansoft», may-2016. [En línea]. Disponible en: <http://hansoft.com/product/feature-list/#large-scale>. [Accedido: 06-may-2016].
- [38] «Team Foundation Service, tu herramienta ALM en la nube», 2016. [En línea]. Disponible en: <https://msdn.microsoft.com/es-es/library/dn168158.aspx>. [Accedido: 07-may-2016].
- [39] Visual Studio, «Team Foundation Server | Visual Studio», 2016. [En línea]. Disponible en: <https://www.visualstudio.com/es-es/products/tfs-overview-vs.aspx>. [Accedido: 02-jun-2016].
- [40] «Software Configuration Management – StarTeam | Micro Focus», 2016. [En línea]. Disponible en: <http://www.borland.com/en-GB/Products/Change-Management/StarTeam>. [Accedido: 08-may-2016].
- [41] «StarTeam, una alternativa más en software libre de gestión de configuración - Software Libre», 2011. [En línea]. Disponible en: <http://www.somoslibres.org/modules.php?name=News&file=article&sid=4205>. [Accedido: 08-may-2016].
- [42] «Evaluación de herramientas de software». [En línea]. Disponible en: <http://es.slideshare.net/tramullas/evaluacin-de-herramientas-de-software>. [Accedido: 10-jun-2016].
- [43] EcuRed, «Requisitos no funcionales - EcuRed», 2016. [En línea]. Disponible en: http://www.ecured.cu/Requisitos_no_funcionales. [Accedido: 10-jun-2016].
- [44] EcuRed, «Requisitos no funcionales - EcuRed», 2010. [En línea]. Disponible en: http://www.ecured.cu/Requisitos_no_funcionales#Requisito_de_Soporte. [Accedido: 22-jun-2016].
- [45] Hipertextual, «Qué es un sistema de control de versiones», *Hipertextual*, 30-abr-2014. [En línea]. Disponible en: <https://hipertextual.com/archivo/2014/04/sistema-control-versiones/>. [Accedido: 03-jul-2016].
- [46] EcuRed, «Metodología ágil - EcuRed». [En línea]. Disponible en: http://www.ecured.cu/Metodolog%C3%ADa_%C3%A1gil. [Accedido: 21-jul-2016].
- [47] «Conceptos básicos de Software». .
- [48] «17.5. Cubos Olap y navegación dimensional con Mondrian y Jpivot.», *El Rincón del BI*, 20-jul-2010. .
- [49] «Pentaho», *Business Intelligence, Data Warehouse, Monterrey, México: Gravatar*. [En línea]. Disponible en: <http://gravitar.biz/pentaho/>. [Accedido: 28-nov-2016].
- [50] E. P. N. Carlos Andrés Fabara De La Paz, «ELABORACIÓN DE UNA PROPUESTA DE PROCESO DE DESARROLLO PARA SISTEMAS WEB BASADA EN LAS PRINCIPALES METODOLOGÍAS EXISTENTES». Quito, feb-2002.
- [51] Rafael Barzanallana, «Rafael Menéndez-Barzanallana Asensio. Historia del desarrollo de aplicaciones Web. Universidad de Murcia», 10-ago-2012. [En línea]. Disponible en: <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>. [Accedido: 28-nov-2016].

- [52] E. M. Paloma Cáceres, «Procesos Ágiles para el Desarrollo de Aplicaciones Web».
- [53] VIC, «Metodologías de desarrollo del software - La tecla de ESCAPE», 29-jul-2015. [En línea]. Disponible en: <http://latecladeescape.com/h/2015/07/metodologias-de-desarrollo-del-software>. [Accedido: 11-dic-2016].
- [54] F. J. Ramírez, «Metodologías para el desarrollo de software».
- [55] J. O. J. Sibaja, «CUADRO COMPARATIVO ENTRE METODOLOGÍAS DE DESARROLLO DE APLICACIONES WEB».
- [56] Endeavour, «Endeavour Agile ALM | Agile Software Development & Application Lifecycle Management», 2011. [En línea]. Disponible en: <http://endeavour-mgmt.sourceforge.net/buy-support.html>. [Accedido: 22-jun-2016].
- [57] Hansoft, «Hansoft Knowledge Base», 2016. [En línea]. Disponible en: <https://support.hansoft.com/hc/en-us>. [Accedido: 03-jul-2016].
- [58] Tuleap, «Support», *Tuleap*, 2014. [En línea]. Disponible en: <https://www.tuleap.org/support>. [Accedido: 22-jun-2016].
- [59] Free Software Foundation, «gnu.org», 12-abr-2016. [En línea]. Disponible en: <https://www.gnu.org/licenses/license-list.es.html>. [Accedido: 22-jun-2016].
- [60] GENBETA, «GPL v3, versión final se hace pública». [En línea]. Disponible en: <http://www.genbeta.com/actualidad/gpl-v3-version-final-se-hace-publica>. [Accedido: 22-jun-2016].
- [61] P. de las H. Q. Jesús González Barahona, «Licencia Pública GNU». [En línea]. Disponible en: <http://gugs.sindominio.net/licencias/gples.html>. [Accedido: 22-jun-2016].
- [62] Aníbal Pérez Liñán, «El Método Comparativo: Fundamentos y Desarrollos Recientes». nov-2007.
- [63] Routio Pentti, «Estudio comparativo», 03-ago-2007. [En línea]. Disponible en: <http://www2.uiah.fi/projects/metodi/272.htm#norm>. [Accedido: 05-jul-2016].
- [64] Secretaría Nacional de la Administración Pública, «Software Libre | Secretaría Nacional de la Administración Pública». [En línea]. Disponible en: <http://www.administracionpublica.gob.ec/software-libre/>. [Accedido: 06-jun-2016].
- [65] R. C. República del Ecuador, «Decreto 1014». 10-abr-2008.
- [66] G. «Gabo» J. López, «Introducción a la Gestión del Ciclo de Vida de las Aplicaciones (ALM) y TFS 2010», *Centro De Estudiantes Microsoft*, 23-mar-2011. .
- [67] «FORMULACION DE CRITERIOS PARA LA SELECCION DE METODOLOGIAS DE DESARROLLO DE SOFTWARE LEONARDO FLOREZ MARIN FELIPE GRISALES TOBON». [En línea]. Disponible en: <http://docplayer.es/1450172-Formulacion-de-criterios-para-la-seleccion-de-metodologias-de-desarrollo-de-software-leonardo-florez-marin-felipe-grisales-tobon.html>. [Accedido: 03-dic-2016].
- [68] «17.3. Preparando el análisis dimensional. Definición de cubos utilizando Schema Workbench.», *El Rincon del BI*, 04-jul-2010. .
- [69] Meteorite.bi, «Saiku | meteorite.bi», 2015. [En línea]. Disponible en: <http://www.meteorite.bi/products/saiku>. [Accedido: 01-dic-2016].
- [70] «metodologia de aplicacion web.doc», *Google Docs*. [En línea]. Disponible en: https://docs.google.com/document/d/1ngnt_Vb3eha4qFkX2Op88zazN14Aww8OeiF-P7-h2K90/edit?hl=es&usp=embed_facebook. [Accedido: 14-dic-2016].
- [71] E. J. G. M. Daniel Mínguez Sanz, «Metodologías para el Desarrollo de Aplicaciones Web: UWE». [En línea]. Disponible en: <http://docplayer.es/3248294-Metodologias-para-el-desarrollo-de-aplicaciones-web-uwe.html>. [Accedido: 14-dic-2016].

k. ANEXOS

Anexo 1: Entrevista a empresas de Desarrollo de Software de la ciudad de Loja.

UNIVERSIDAD NACIONAL DE LOJA

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES
NO RENOVABLES**

ESCUELA DE INGENIERÍA EN SISTEMAS

ENTREVISTA

Empresa:

Entrevistado:

Dirección:

Fecha: 20 – 05 – 2016

Buenas tardes, la finalidad de la presente entrevistas es conocer como su empresa realiza la Gestión de Ciclo de vida de sus Aplicaciones.

1. ¿Qué tipo de requerimientos debe tener una herramienta para ser utilizada en la gestión del ciclo de vida de una aplicación?
2. ¿Cuáles son los requerimientos mínimos?
3. ¿Qué requerimientos debería tener para ser una herramienta exitosa?
4. Su equipo de desarrollo, ¿usa un solo tipo de herramienta para todos los proyectos? ¿Abarca todos los requerimientos?
5. ¿Qué tipo de requerimientos son más difíciles de gestionar con una herramienta?
6. ¿Considera necesario que las herramientas contengan plantillas para la realización de reportes o documentación?
7. Al usar estas herramientas, ¿su equipo de desarrollo usa las plantillas que vienen predeterminadas de las metodologías, las adapta o simplemente no las usa?
8. ¿Cómo gestiona los riesgos su equipo de desarrollo?
9. ¿Cómo gestiona el tiempo y el cumplimiento de tareas su equipo de desarrollo?
10. ¿Cómo resuelve una estimación errónea de costos?

Anexo 2: Entrevista 1

UNIVERSIDAD NACIONAL DE LOJA

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES
NO RENOVABLES**

ESCUELA DE INGENIERÍA EN SISTEMAS

ENTREVISTA

Empresa: IOET Cía. Ltda.

Entrevistado: Ing. Patricio Valarezo

Dirección: Av. Pío Jaramillo Alvarado y Reinaldo Espinosa, La Argelia

Fecha: 20 – 05 – 2016

Buenas tardes, la finalidad de la presente entrevistas es conocer como su empresa realiza la Gestión de Ciclo de vida de sus Aplicaciones.

1. ¿Qué tipo de requerimientos debe tener una herramienta para ser utilizada en la gestión del ciclo de vida de una aplicación?

Bueno, en primer lugar lo que se apega concretamente a la metodología que estamos usando y hemos evaluado algunas herramientas para la gestión de proyectos, gestión de productos, gestión de requisitos y hay cosas que se pegan cosas que no se pegan. Siempre aprendemos de la experiencia de nuestros clientes, tenemos clientes que ya han estado trabajando con unas herramientas que recién llegamos a conocer y decimos porque les gusta esto o es por esto y por esto. La evaluación va por muchas partes pero lo que concretamente termina siendo usado es lo que es más cómodo para nosotros.

2. ¿Cuáles son los requerimientos mínimos?

Mínimo mínimo tiene que ser multiusuario y multitarea, es en lo primero en lo que se debe pensar, otra cosa es que sea una aplicación a nivel de servicio porque no podemos darnos el lujo de instalar un programa en una computadora y nadie lo pueda usar porque sólo tengo licencia para esta; si no tenemos la aplicación y no tenemos el dinero ni el presupuesto como para costear una aplicación que esté colgada en la nube nos proponemos dos partes o hacemos un Clon de esa aplicación y buscamos

alternativas bastante tecnológicas para resolverlo si el problema se vuelve crítico pues tenemos que pagar eso sí.

En lo relacionado al Control de versionamiento

Nosotros estamos trabajando con Git o mejor dicho únicamente desde que evolucionó ahora tenemos un control de versiones multidistribuido muy bueno. Pero cuál es el asunto, existen proveedores de git en el mundo, primero empezamos con una herramienta propia levantamos la aplicación en uno de nuestros servidores y vimos que tenían problemas y nos cambiamos al GitHub.

3. ¿Qué requerimientos debería tener para ser una herramienta exitosa?

Principalmente los requerimientos que acabo de mencionarle.

4. Su equipo de desarrollo, ¿usa un solo tipo de herramienta para todos los proyectos? ¿Abarca todos los requerimientos?

Tenemos muchas herramientas realmente pero concretamente damos la libertad a cada uno de los programadores; por ejemplo un editor de texto, cada uno es libre y puede usar lo que quiera. Pero tenemos un código de conducta respecto a la liberación de código, por ejemplo no pueden liberar código que esté mal escrito y tratamos de usar algunas herramientas para solucionar ese problema.

5. ¿Qué tipo de requerimientos son más difíciles de gestionar con una herramienta?

Tal vez las características de algunos de los clientes, ya que algunos son muy exigentes me piden que uno haga una especie de ritual cuándo va a subir el código. Pero hay unos que son más abiertos y nos dan la libertad de que nosotros lo hagamos de la forma que nosotros queremos, pero cuando más grande es la institución o el cliente y mientras más dinero tenga de por medio son mucho más estrictos a cómo se opera la situación. Yo creo que instituciones como universidades deberían ser más estrictos respecto a esto y no invertir mucho en Hardware porque el hardware se caduca, por eso prefiero invertir yo en desarrolladores y que los desarrolladores tengan un buen IDE.

6. ¿Considera necesario que las herramientas contengan plantillas para la realización de reportes o documentación?

Sí, aunque claro siempre es bueno personalizar porque hay clientes que quieren hacerlo impecablemente clientes que se lucen mucho, Héctor responde específicamente a casos como de analíticas dónde tenemos clientes que tienen sus equipos propios de analytics y ellos tratan de averiguar cuántas personas entraron a mi web, cuántos me compraron y cuántos no me compraron y de los que no me compraron porque no lo hicieron, y tratar de analizar este problema para convencer a los que no me compraron de que compren. Y para ese tipo de cosas necesitan reportes específicos, ahora tener un reporte estándar para los que no tienen nada ayuda hartísimo si tenemos plantillas en el hecho de decir yo no sé nada de esto ni este campo me va a servir mucho esa la plantilla de ahí que me dé la facilidad de cambiarla, modificarla y personalizarla que son tres cosas importantes: cambiarla para poner la fecha arriba o abajo, modificarla que me permita hacerlo en una forma muy práctica y personalizarla que me permite agregarle un logo o algún tipo de cosas.

7. Al usar estas herramientas, ¿su equipo de desarrollo usa las plantillas que vienen predeterminadas de las metodologías, las adapta o simplemente no las usa?

Si, hablando concretamente de una de las herramientas que utilizado para la metodología una de ellas se llama Jira, que me permite generar reportes y saber que están haciendo cada uno de los desarrolladores cuáles son sus tickets y cuantos tickets resolvieron durante una semana. Qué tan bueno es, pues tenemos gráficas hoy por hoy utilizamos las que nos provee la herramienta es bueno que la herramienta nos dé algo y si quieres cambiarlo pues lo haces así, o sea puedes hacerlo pero arréglatelas; por ejemplo cuando trabajamos con productos que compramos estos tienen un muy buen reporte con horas y todo eso.

8. ¿Cómo gestiona los riesgos su equipo de desarrollo?

Bueno, específicamente aquí no hacemos gestión de riesgos porque son más casos administrativos generalmente la gestión de riesgos es manejada por el cliente, en donde los clientes se arriesgan a contratar equipos de trabajo extranjeros por ejemplo la empresa y iote que está en Ecuador que va a costar 3 veces más con relación a una empresa local siendo está una empresa estadounidense, pros y contras donde ellos

evalúan el riesgo, nosotros básicamente nos centramos en hacer releases o entregas del código.

9. ¿Cómo gestiona el tiempo y el cumplimiento de tareas su equipo de desarrollo?

Se evalúa, existe una persona que sabe programar porque para eso se necesita un programador y él dice este trabajo te va a tomar exactamente 3 días y lo que yo hago siendo el top desarrollador de aquí digo: a mí ese trabajo me toma un día, a mis mejores desarrolladores les doy una holgura de 2 días y les digo si a mí me toma un día puede que a ellos les tome dos o menos y mejor quiere decir que el “man” está trabajando. Tenemos varios niveles de desarrolladores: los top que tienen maestrías y buen nivel que trabajan en Industrias y saben bastante pero también tengo la cola de la gente y a ellos les doy holguras de una semana con el mismo trabajo pero en una semana, para eso generalmente no usamos una herramienta. Sólo nos informa si ha cumplido o no ha cumplido pero hay herramientas que aún no le permiten trackear el tiempo a nivel de desarrollador, la parte bonita de esto es que nosotros hacemos evaluaciones periódicas y decimos esta persona tiene buenos feedbacks de gente.

En lo relacionado al Incumplimiento fechas: Se estima primero, gracias a Dios tenemos gente que es cuerda en el nivel de proyectos y sabe que no puede pedir una locura, yo he trabajado en la industria local universidades y rectores y ellos no ven nada él sólo dice quiero que me hagan esto y tienen un mes y se vá, en donde los desarrolladores se sienten frustrados al realizar ajuste de tiempos y a la final esos proyectos fracasan. Pero dando gracias a Dios con la gente que trabajo internacionalmente conocen el negocio, mientras la persona tenga alguien de confianza que sepa estimar tiempos y si es un programador y dice esto va a costar tanto y tanto podemos presionarlo sí, pero siempre y cuando el trabajo de él sea respetado de mejor manera y repito nuevamente, el desarrollo de un sistema no es una cosa que empiezo y termino en un mes, sino es una cosa que va a continuar funcionando, si a esa persona le gustó el sistema Yo sé que dentro de dos meses va a necesitar un cambio, por ejemplo el IVA dentro de un mes va a cambiar al 14% en donde muchas empresas van a necesitar cambios en su sistema, y si ellos se portaron mal con los programadores tendrán que pagar el doble o el triple de lo que hubieran pagado y si se hubieran portado bien.

Nosotros usamos Notificaciones: Bastante por supuesto, incluso tenemos todo tipo de notificaciones de hecho trabajamos con una herramienta que nos ayuda a comunicarnos entre nosotros, es una herramienta que es una mezcla de chat con servicios en línea, tenemos boots que nos explican por ejemplo el comit que usted acaba de hacer en el servidor está fallando en tal test por favor revíselo; entonces todo mundo está notificado a tiempo y la idea de esto es filtrar todos los errores que podamos tener y poder terminar con una mejor versión y las notificaciones puede ser a todo nivel desde una notificación al celular y depende básicamente de cómo nos organizamos, la clave de esto es la integración donde puedes conectarte con Jira y Taiga cualquier otro tipo de CRM o aplicaciones en donde se produce una incidencia y se replica a todos los programadores, ahora podemos saber más pronto que el cliente mismo, que el cliente está teniendo problemas e incluso solucionarlo antes de que el cliente nos llame para comentarnos del problema, no estamos esperando que el cliente nos lo diga gracias a Dios.

10. ¿Cómo resuelve una estimación errónea de costos?

La estimación de costos siempre va a ser un problema por eso nosotros tenemos una premisa del código, lo que realmente deba a costar y hablando de estimación nuestros clientes siguen pensando que la casa es un sistema y que él compró un sistema como comprar una casa, es por eso que los clientes internacionales nos contratan mediante un contrato abierto, en donde el cliente se dedica a su empresa y no a pensar en que esté funcionando bien o no el sistema, pero no me va a ver cómo un gasto; en el momento en el que la empresa me empiece a ver como un gasto estamos en problemas. La estimación para mí es “cuánto está dispuesto a pagar usted y como le gusta tratarse”.



.....
Ing. Patricio Valarezo

Anexo 3: Entrevista 2

UNIVERSIDAD NACIONAL DE LOJA

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES
NO RENOVABLES**

ESCUELA DE INGENIERÍA EN SISTEMAS

ENTREVISTA

Empresa: UTI (Unidad de Telecomunicaciones e Información).

Entrevistado: Ing. Edison Coronel.

Dirección: Av. Pío Jaramillo Alvarado y Reinaldo Espinosa, La Argelia. UNL.

Fecha: 20 – 05 – 2016

Buenas tardes, la finalidad de la presente entrevistas es conocer como su empresa realiza la Gestión de Ciclo de vida de sus Aplicaciones.

1. ¿Qué tipo de requerimientos debe tener una herramienta para ser utilizada en la gestión del ciclo de vida de una aplicación?

Gestionar procesos, gestionar tareas, gestión de riesgos, gestión de tiempos y costos del proyecto, especialmente lo que son las alertas o notificaciones y sobre todo las dependencias de algunas tareas. Por ejemplo, si yo quiero hacer una tarea que me involucre ciertas funciones que dependen de otra tarea, otra cosa es la notificación de llegar a plazos, notificación de una tarea cumplida por otra persona, que la herramienta que estamos utilizando actualmente sí la cumple. Pero ahí también nos falta un poquito una cultura organizacional, porque ahí tenemos todas las tareas y por a o b razón nos descuidamos no actualizamos la tarea que nosotros hicimos; entonces no saben las otras personas, entonces también necesitaríamos una alerta que nos diga "actualice".

Por qué cumplirlas

Porque al ser nosotros una entidad pública estamos sujetos a control por otras entidades como contraloría, auditoría etc. Entonces tenemos que evidenciar el trabajo

que se está realizando de tal forma que se evidencie que se ha cumplido los plazos establecidos Y de esa forma cumplir con nuestra labor como funcionarios públicos. Otra es que como desarrolladores nosotros tenemos que llevar nuestro orden de las tareas, entonces tenemos que planificar todo lo que tenemos que hacer y cumplir con todo lo planificado ya que la mente es frágil y a veces nosotros decimos tengo que hacer esto, esto y si no lo registramos en algún lado se olvida y a veces no lo cumplimos y es una forma de organizarnos a nosotros mismos como desarrolladores.

2. ¿Cuáles son los requerimientos mínimos?

Social, colaborativa y notificaciones.

3. ¿Qué requerimientos debería tener para ser una herramienta exitosa?

Que sea adaptable por ejemplo, ahorita trabajamos con una organización por cualquier razón esta cambia, la herramienta debe adaptarse a la nueva estructura eso sería lo fundamental.

4. Su equipo de desarrollo, ¿usa un solo tipo de herramienta para todos los proyectos? ¿Abarca todos los requerimientos?

Una única herramienta, incluso ahora que está aquí el ingeniero Milton nosotros empezamos a tener una cultura de que todos los proyectos tienen que estar evidenciados digitalmente y no estar vinculados a una cuenta; que significa eso, yo como Edison Coronel tengo mi propia cuenta entonces cuando yo deje la subdirección me llevo toda la información. Entonces lo que nosotros estamos haciendo es que estamos ligados a una cuenta de la subdirección. Entonces el subdirector que venga después de mí va a tener toda la información, entonces ya no sería Edison Coronel el propietario de la información de la subdirección.

5. ¿Qué tipo de requerimientos son más difíciles de gestionar con una herramienta?

Lo más difícil es cumplir con los tiempos planificados, porque al ser una entidad que da soporte técnico a toda la universidad por a o b siempre vienen a pedir soporte y nosotros tenemos que dejar de hacer la planificación que tenemos para brindar soporte entonces eso es lo más difícil de gestionar, el tiempo.

Incumplimiento plazos

Cuando ya estamos cerca de los tiempos nos apurarnos e inclusive quedarnos más tiempo trabajando hasta un poquito fuera del horario llevándonos el trabajo a la casa para poder cumplir con los plazos, porque otra cosa es que nosotros debemos cumplir con los plazos quiera o no se quiera.

6. ¿Considera necesario que las herramientas contengan plantillas para la realización de reportes o documentación?

Si, inclusive nosotros dentro de la unidad definimos un modelo de plantilla. Entonces cuando se define una plantilla se la elabora y la mandamos a que sea aprobada por un comité, entonces ese comité aprueba la plantilla y se la empieza a utilizar; todo esto lo hacemos mediante Google apps entonces ahí las tenemos a todas almacenadas.

7. Al usar estas herramientas, ¿su equipo de desarrollo usa las plantillas que vienen predeterminadas de las metodologías, las adapta o simplemente no las usa?

Por ejemplo, si es que hay una plantilla predeterminada en una herramienta la sometemos análisis y luego si es que cumple un análisis determinado la utilizamos y si es que no la modificamos, para que se cumpla con los estándares por ejemplo, vienen plantillas predeterminadas pero nos exigen que éstas sean con los colores institucionales y a los formatos que tienen aquí, más trabajamos especialmente con la UCI (Unidad de Comunicación Institucional).

8. ¿Cómo gestiona los riesgos su equipo de desarrollo?

En base al alcance del proyecto.

9. ¿Cómo gestiona el tiempo y el cumplimiento de tareas su equipo de desarrollo?

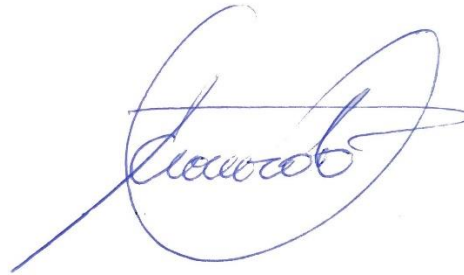
Hay incumplimiento de tareas especialmente cuando dependemos de otro departamento, qué significa; digamos que por ejemplo necesitamos el orgánico estructural y funcional aprobado legalmente y si es que no nos proveen esa información a tiempo ahí hay el incumplimiento.

Para solucionar en este caso lo que hacemos es crear un ambiente, simularlo, pero no es un ambiente real sino simulado entonces creamos ambientes simulados para cumplir. Otro incumplimiento se da por fuerza mayor por ejemplo, cuando un técnico sufre un percance a su salud entonces sus tareas son asignadas a otro técnico y este

le da apoyo ahora. Estas tareas que tenemos aquí las planteamos de tal manera que no las haga una sola persona sino por pares por ejemplo, en el sistema de las unidades de titulación no ponemos un único encargado sino ponemos dos y uno siempre va a estar a la cabeza, si por a o b situaciones necesita apoyo tiene al que está acá y le apoya y así mismo capacita a otro después. Y de esa forma siempre van a haber dos personas y evitaremos que un sistema se quede sin soporte o una actividad sin resolverse.

10. ¿Cómo resuelve una estimación errónea de costos?

Mediante una renegociación.



.....
Ing. Edison Coronel

Anexo 4: Entrevista 3

UNIVERSIDAD NACIONAL DE LOJA

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES
NO RENOVABLES**

ESCUELA DE INGENIERÍA EN SISTEMAS

ENTREVISTA

Empresa: Prendho.

Entrevistado: Ing. Carlos Correa

Dirección: San Cayetano Alto, Calle París. UTPL.

Fecha: 20 – 05 – 2016

Buenas tardes, la finalidad de la presente entrevistas es conocer como su empresa realiza la Gestión de Ciclo de vida de sus Aplicaciones.

1. ¿Qué tipo de requerimientos debe tener una herramienta para ser utilizada en la gestión del ciclo de vida de una aplicación?

Dependería más que todo del campo de aplicación de la herramienta de software, hay condiciones de mercado que son muy cambiantes que requieren actualizaciones muy rápidas, por ejemplo aplicativos móviles, aplicativos desktop y hay otro tipo de requerimientos que en cambio impacta mucho en el ciclo de vida del software por ejemplo un ERP, un LMS, son herramientas que pueden tener un mayor alcance en el tiempo. Entonces, dependiendo de esos dos la misma estructura que tenemos para el desarrollo de software se aplicaría al tema de la maduración y escalamiento por ejemplo, versionamiento eso sería una herramienta que ayudaría muchísimo a la gestión. Antiguamente el versionamiento era descentralizado y más que todo con aplicativos desktop tenía que bajarme el control de versionamiento en mi equipo, ahora la recomendación es que ese versionamiento sea en la nube y colaborativo. Herramientas como GitHub o cualquiera de esas podrían ayudar mucho. Otro es el aseguramiento de calidad que también ayudaría mucho a mejorar el ciclo de vida con diferentes técnicas como son pruebas de unidad, pruebas de volumen, etc. Y también

una correcta aplicación de los criterios de ingeniería de software, asignación de procesos, metodologías ágiles como Scrum, todo lo que la teoría de Ingeniería de Software recomienda.

2. ¿Cuáles son los requerimientos mínimos?

Lo que he argumentado creo que está por ahí, si el aplicativo tiene un ciclo de vida súper rápido habría que tener cuidado con temas de versionamiento aseguramiento de calidad e Ingeniería de Software esos 3 básicamente.

3. ¿Qué requerimientos debería tener para ser una herramienta exitosa?

Desde el ámbito neto y técnico debería tener los requerimientos antes mencionados pero desde el ámbito gerencial del proyecto tiene que tener mucha interacción con el usuario porque al final el usuario impacta mucho en qué tan rápido tiene que madurar el software y qué tan rápido tiene que ser el ciclo de vida, porque como técnico puedo armar una plataforma pero si no escucho al usuario, que es al final quién manda, a quién da la normativa no serviría de nada. Entonces si nos referimos al ámbito netamente técnico yo me quedaría con esas.

4. Su equipo de desarrollo, ¿usa un solo tipo de herramienta para todos los proyectos? ¿Abarca todos los requerimientos?

Depende del tipo de mercado, ahí depende muchísimo, hay herramientas que usamos de terceros cuando usamos software de código abierto dependemos mucho de la comunidad de software que está aportando. Cuando son desarrollos propios ahí sí somos quienes marcamos esa pauta en el tema de software de código abierto o software libre dependemos mucho de la comunidad.

5. ¿Qué tipo de requerimientos son más difíciles de gestionar con una herramienta?

El alcance siempre es una dificultad, especialmente cuando el usuario no tiene claro qué es lo que quiere; desarrollamos una plataforma y el usuario visualiza cuál es su alcance, hasta dónde quiero ir y con qué funcionalidades; qué historias de usuario voy a referir, qué características de la plataforma, y ayudarle al usuario cuesta mucho. Pues todavía no se aclara, eso cambia mucho y no digo que esté mal. Eso le da agilidad; Scrum por ejemplo se adapta mucho a esas cosas pero cuando todavía no lo tiene claro eso cuesta mucho al usuario.

6. ¿Considera necesario que las herramientas contengan plantillas para la realización de reportes o documentación?

Si, en BPM ahí es ideal dependiendo de plantillas del mercado al que el software va a ayudar a funcionar, es ideal tener plantillas.

7. Al usar estas herramientas, ¿su equipo de desarrollo usa las plantillas que vienen predeterminadas de las metodologías, las adapta o simplemente no las usa?

Adaptamos, a mí personalmente me gusta la metodología pero nunca termina siendo una camisa de fuerza, la solución nuestra debe estar por encima de la metodología, la metodología no tiene que terminar marcándome la pauta. Entonces tomo muchas cosas de la metodología que me parecen muy buenas y además de eso de la experiencia de proyectos previos lo tomo y junto ambas cosas.

8. ¿Cómo gestiona los riesgos su equipo de desarrollo?

Mapa de riesgos y dependiendo del alcance del proyecto, con proyectos mucho más grandes se asignaba un gerente de riesgos y el mitigaba.

9. ¿Cómo gestiona el tiempo y el cumplimiento de tareas su equipo de desarrollo?

En gestión de proyectos Project Management Office (La Oficina de Gestión de Proyectos) a través de una programación con diagrama de Gantt, un tablero que ayude a que se cumplan los tiempos.

En lo relacionado al Incumplimiento de tareas

Hay varios motivos, y depende del que cambia requerimientos o agrega cosas nuevas entonces renegociamos el sprint, entonces si el sprint tenía que entregarte en 15 días el cliente trae nuevos requerimientos junto con él se renegocia el sprint o ese sprint se subdivide en 2 entregables distintos o sprints distintos, con distintas fechas si depende de algo nuestro como equipo de desarrollo ahí generalmente lo que se hace es más horas extras de desarrollo. Si se estimaba que iban a ser 20 horas en una semana pues dicen nos hemos demorado, dupliquemos el número de horas para poder cumplir ese sprint. Hay que entender que siempre en Scrum los sprints son negociables con todo el equipo.

10. ¿Cómo resuelve una estimación errónea de costos?

Depende, hay casos en donde la herramienta proporcionan insumos para tomar una decisión que impacta en el costo, si mi diagrama de Gantt me está diciendo de que el problema es que el desarrollador se despistó y gastó unas horas en algo que no debería hacerlo, ahí es la responsabilidad del desarrollador, de la empresa desarrolladora y entonces tienen que asumir ese costo en negociación con el desarrollador. Pero si la demora parte del cliente ahí se renegocia el sprint se le dice al cliente: tienes que tener conciencia que es un nuevo requerimiento y no está contemplado en las historias de usuario inicial.



.....
Ing. Carlos Correa

Anexo 5: Entrevista 4

UNIVERSIDAD NACIONAL DE LOJA

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES
NO RENOVABLES**

ESCUELA DE INGENIERÍA EN SISTEMAS

ENTREVISTA

Empresa: GOOTLEK Cía. Ltda.

Entrevistado: Ing. Fernando Dávila

Dirección: Bolívar y Miguel Riofrío

Fecha: 20 – 05 – 2016

Buenas tardes, la finalidad de la presente entrevistas es conocer como su empresa realiza la Gestión de Ciclo de vida de sus Aplicaciones.

1. ¿Qué tipo de requerimientos debe tener una herramienta para ser utilizada en la gestión del ciclo de vida de una aplicación?

Primeramente, debe ser de tipo distribuida, es decir cliente-servidor. Nosotros utilizamos GitLab, que es la versión gratuita de GitHub. Además debe tener servicios en la nube y utilizar nodos, aprobación de versiones y control de versiones, ya que cada miembro del equipo desarrolla una parte del proyecto y realiza sus propias modificaciones, además debe permitir regresar a una versión estable en caso de que algo falle, debe tener el servicio de notificaciones, para que el desarrollador pueda informar si está a gusto con su trabajo designado o a su vez conocer si ya culminó su tarea, la herramienta debe permitir unificar todo el trabajo una vez esté terminado. Debe permitir gestionar los proyectos y archivar el código. En lo personal usamos Mantis para la gestión de tareas, en donde se puede también asignar expedientes y realizar peticiones en caso de que no se haya podido culminar el trabajo en el tiempo establecido.

Considero que debe cumplir estos requerimientos, una porque va a ser la guía total de la aplicación y otra... Bueno al menos aquí contamos con un equipo estable y de forma legal en la empresa pero en otros lados pues llega un punto en el que el programador pierde la cabeza o le llega una nueva oportunidad de trabajo y se va, entonces la persona que lo reemplaza simplemente coge la cuenta de usuario y esa herramienta lo va a orientar y le va a decir donde esta cada cosa. Otra cosa también en lo referente a servicios en la nube y tecnologías como GitHub o GitLab nos permiten que el programador no tenga que estar aquí físicamente, actualmente se está haciendo así, se están contratando programadores “free lanser”, donde por ejemplo yo desarrollo un API, pero escucho que existe un argentino fenomenal o un peruano que es una “bestia” programando la interfaz gráfica, entonces yo lo contrato y le doy acceso por medio de estas herramientas a mi API, entonces simplemente el cobra su dinero y me da toda la documentación necesaria y yo le doy el acceso a la herramienta para que le guie y me de todas las facilidades.

2. ¿Cuáles son los requerimientos mínimos?

Primero, nos debe llevar un control total del ciclo de vida de una aplicación, desde los requerimientos, los diagramas, las versiones del código y si fuera posible el control de las pruebas unitarias.

3. ¿Qué requerimientos debería tener para ser una herramienta exitosa?

Una combinación entre Mantis y GitHub sería lo ideal, en donde lleve el ciclo de vida de la aplicación pero también lleve el control de versiones del código. Uso las dos herramientas porque se complementan, porque mientras Mantis me controla “quien está haciendo qué”, GitHub me controla “quien hizo qué” y sobretodo también hay herramientas para ver la eficiencia de los programadores, miden las líneas de código, miden todo eso, entonces se pueden sacar estadísticas del proyecto, se puede ver o determinar si una tecnología no fue exitosa, y si crearon un dolor de cabeza ya no volver a utilizarlas de nuevo.

4. Su equipo de desarrollo, ¿usa un solo tipo de herramienta para todos los proyectos? ¿Abarca todos los requerimientos?

Solo de software, hay proyectos en los que se requiere el uso de servidores y esas cosas, ya se requiere otro tipo de herramienta como Packet Tracer. Y a veces hay

proyectos que no tienen que ver mucho con programación y no se utilizan herramientas.

5. ¿Qué tipo de requerimientos son más difíciles de gestionar con una herramienta?

Los requerimientos funcionales, son los más difíciles porque normalmente en la primera entrevista con el usuario, si este ha tenido una experiencia previa que chévere, él sabe lo que quiere o si el usuario tiene un departamento de sistemas hay una persona que puede hablar con usted, pero cuando el usuario los que tiene son expertos ya en gestión de procesos o alguna cosa esas personas la primera vez no saben que pedir y describen los procesos y funcionalidades de una manera y al presentar los requerimientos en la primera reunión están bien pero va a la siguiente reunión y le dicen que no están bien. Entonces eso la herramienta debería tratar de llevar un control.

6. ¿Considera necesario que las herramientas contengan plantillas para la realización de reportes o documentación?

Es muy necesaria principalmente por motivo manejar de manera estandarizada la documentación de proyectos y reporte de avances.

7. Al usar estas herramientas, ¿su equipo de desarrollo usa las plantillas que vienen predeterminadas de las metodologías, las adapta o simplemente no las usa?

Cuando la herramienta las posee claro que las usamos, como es caso del Enterprise Architect, en algunos casos hasta se ha comprado dichas plantillas a la empresa desarrolladora, pero cuando no hay se han generado plantillas propias a nivel interno del grupo.

8. ¿Cómo gestiona los riesgos su equipo de desarrollo?

Calificamos en un formato los requerimientos solicitados por el cliente por complejidad de desarrollo, el uso de nuevas tecnologías de hardware, de software, aquellos requerimientos que retrasarían al proyecto al requerir un mayor esfuerzo e inversión de tiempo y dinero por parte del equipo.

9. ¿Cómo gestiona el tiempo y el cumplimiento de tareas su equipo de desarrollo?

Dividimos las tareas y establecemos fechas de entrega primero a nivel interno y luego para con el cliente.

Tiene costos reflejados en la remuneración mensual de cada uno, obviamente si todo se cumple existen premios y bonos.

10. ¿Cómo resuelve una estimación errónea de costos?

Se trata de pactar nuevamente con el cliente, pero la mayoría de veces estos errores los asume la empresa.



.....
Ing. Fernando Dávila

Anexo 6: Análisis de Entrevistas

Entrevista realizada a empresas de Desarrollo de Software de la ciudad de Loja

El objetivo de las entrevistas realizadas es conocer cuáles son los requerimientos previos que debe cumplir una herramienta para la Gestión del Ciclo de Vida antes de ser utilizada, con la finalidad de establecer si sus funcionalidades serán capaces de suplir o cubrir las necesidades del proyecto a realizarse. Y recolectar información que posteriormente permitirá establecer los criterios de evaluación.

1. ¿Qué tipo de requerimientos debe tener una herramienta para ser utilizada en la gestión del ciclo de vida de una aplicación?

Respuestas	Fr	%
1. tipo distribuida	1	4,34%
2. servicios en la nube	1	4,34%
3. control de versiones	2	8,69%
4. unificar el trabajo	1	4,34%
5. notificaciones	1	4,34%
6. gestión de tareas	2	8,69%
7. gestión de proyectos	2	8,69%
8. gestión de procesos	2	8,69%
9. gestión de productos	1	4,34%
10. gestión de requisitos	1	4,34%
11. gestión de riesgos	1	4,34%
12. gestión de tiempos	1	4,34%
13. gestión de costos	1	4,34%
14. gestión de código	1	4,34%
15. colaborativa	1	4,34%
16. aseguramiento de calidad	1	4,34%
17. control de pruebas	1	4,34%
18. uso de metodologías ágiles	2	8,69%
Totales	23	100%



Análisis e Interpretación de los Resultados

El 8,69% de las personas entrevistadas manifestaron que los requerimientos que debe tener una herramienta para ser utilizada en la gestión del ciclo de vida de una aplicación son el control de versiones, gestión de tareas, proyectos y procesos, y permitir el uso de metodologías ágiles. Ante un 4,34% que consideran importantes la gestión de requisitos, riesgos y tiempos, entre otros requerimientos.

2. ¿Cuáles son los requerimientos mínimos?

Respuestas	Fr	%
1. control de requerimientos	1	7,69%
2. control de diagramas	1	7,69%
3. control de versiones	2	15,38%
4. control de pruebas unitarias	1	7,69%
5. control de calidad	1	7,69%
6. ingeniería de software	1	7,69%
7. multiusuario	1	7,69%
8. multitarea	1	7,69%
9. a nivel de servicio	1	7,69%
10. social	1	7,69%
11. colaborativa	1	7,69%
12. notificaciones	1	7,69%
Totales	13	100%

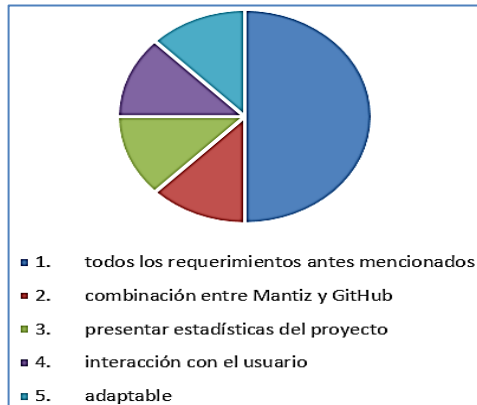


Análisis e Interpretación de los Resultados

El 15,38% de los entrevistados aseguran que el control de versiones es uno de los requerimientos mínimos que debe poseer una herramienta para gestionar el ciclo de vida de una aplicación. Mientras que el 7,69% manifiesta que el control de requerimientos, diagramas, pruebas unitarias y calidad, ser a nivel de servicio, multiusuario, multitarea, social, colaborativa y tener notificaciones como parte de estos requerimientos.

3. ¿Qué requerimientos debería tener para ser una herramienta exitosa?

Respuestas	Fr	%
1. todos los requerimientos antes mencionados	4	50%
2. combinación entre Mantis y GitHub	1	12,5%
3. presentar estadísticas del proyecto	1	12,5%
4. interacción con el usuario	1	12,5%
5. adaptable	1	12,5%
Totales	8	100%

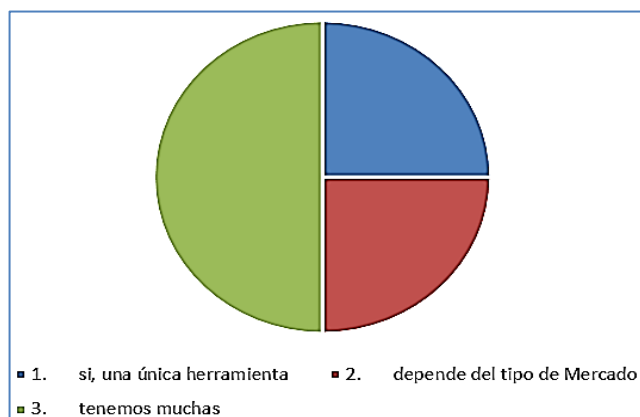


Análisis e Interpretación de los Resultados

En este caso el 50% de las personas entrevistadas manifiestan que una herramienta sería exitosa si cumpliera con todos los requerimientos que ellos mencionaron con anterioridad, mientras que el 12,5% asegura que el éxito está en una combinación entre Mantis y GitHub, la presentación de estadísticas del proyecto, interacción con el usuario y la adaptabilidad.

4. Su equipo de desarrollo, ¿usa un solo tipo de herramienta para todos los proyectos? ¿Abarca todos los requerimientos?

Respuestas	Fr	%
1. si, una única herramienta	1	25%
2. depende del tipo de Mercado	1	25%
3. tenemos muchas	2	50%
Totales		100%

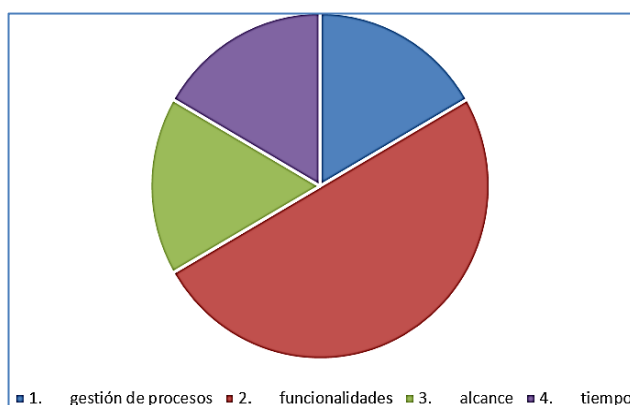


Análisis e Interpretación de los Resultados

El 50% de las personas entrevistadas aseguran que tienen diferentes herramientas para la gestión del ciclo de vida de sus aplicaciones, ante un 25% que manifiesta que utilizan una única herramienta o una o varias depende del tipo de mercado.

5. ¿Qué tipo de requerimientos son más difíciles de gestionar con una herramienta?

Respuestas	Fr	%
1. gestión de procesos	1	16,66%
2. funcionalidades	3	50%
3. alcance	1	16,66%
4. tiempo	1	16,66%
Totales	6	100%



Análisis E Interpretación De Los Resultados

El 50% de los entrevistados manifiestan que las funcionalidades de una aplicación son los requerimientos más difíciles de gestionar, mientras que el 16,66% consideran que la gestión de procesos, el alcance y el tiempo son los más difíciles.

6. ¿Considera necesario que las herramientas contengan plantillas para la realización de reportes o documentación?

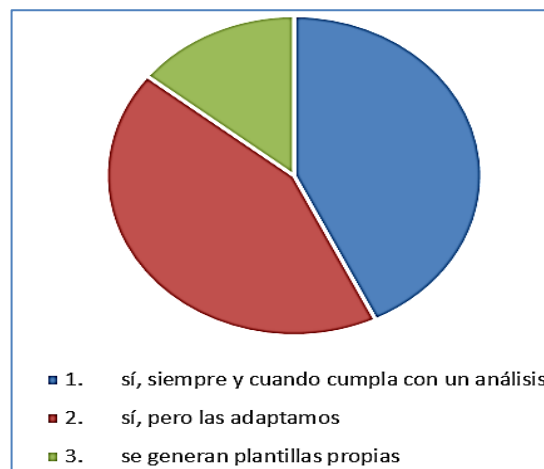
Respuestas	Fr	%
1. Si	4	100%
Totales	4	100%

Análisis e Interpretación de los Resultados

En este caso el 100% de los encuestados aseguran que es necesario que las herramientas contengan plantillas para la realización de reportes o documentación.

7. Al usar estas herramientas, ¿su equipo de desarrollo usa las plantillas que vienen predeterminadas de las metodologías, las adapta o simplemente no las usa?

Respuestas	Fr	%
1. sí, siempre y cuando cumpla con un análisis	3	42,85%
2. sí, pero las adaptamos	3	42,85%
3. se generan plantillas propias	1	14,28%
Totales	7	100%

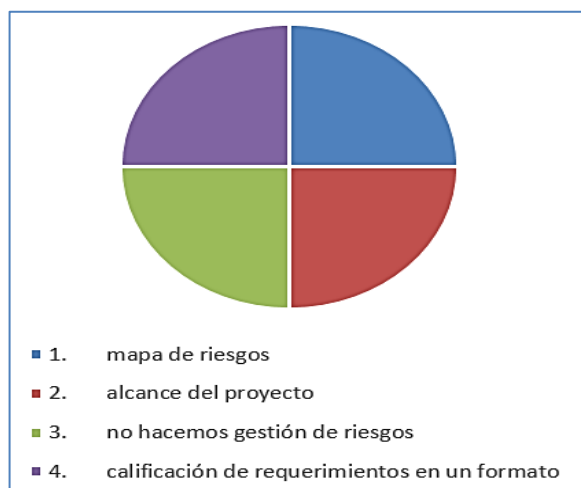


Análisis E Interpretación De Los Resultados

El 42,85% de los entrevistados manifestó que su equipo de desarrollo usa las plantillas que vienen predeterminadas de las metodologías siempre y cuando estas cumplen con un análisis, sino lo que hacen es adaptarlas a sus necesidades. Mientras que el 14,28% generan plantillas propias.

8. ¿Cómo gestiona los riesgos su equipo de desarrollo?

Respuestas		Fr	%
1.	mapa de riesgos	1	25%
2.	alcance del proyecto	1	25%
3.	no hacemos gestión de riesgos	1	25%
4.	calificación de requerimientos en un formato	1	25%
Totales		4	100%



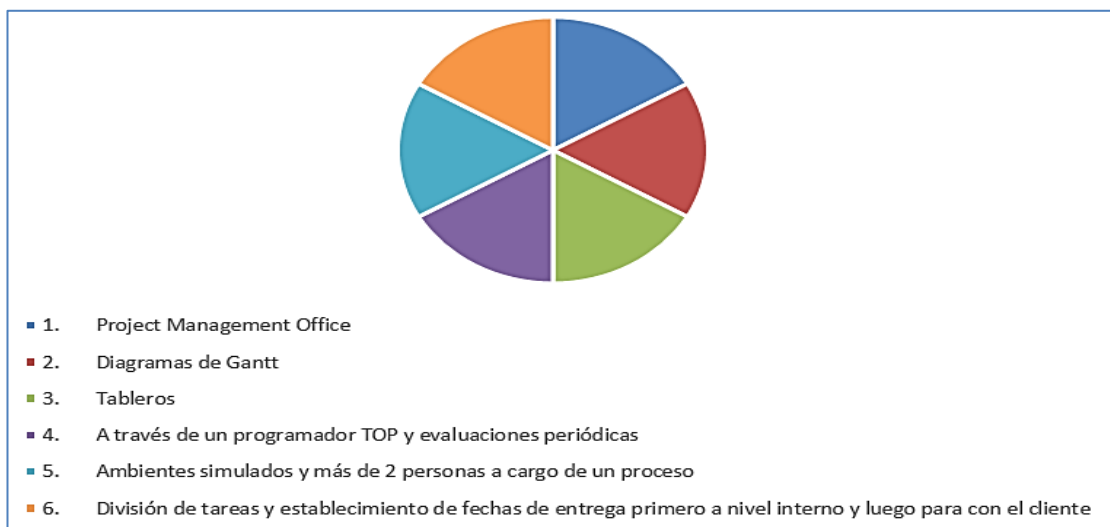
Análisis E Interpretación De Los Resultados

El 25% de los entrevistados afirman que su equipo de desarrollo gestiona los riesgos mediante el uso de mapa de riesgos, alcance del proyecto o la calificación de requerimientos en un formato. Mientras que el 25% restante no realiza gestión de riesgos.

9. ¿Cómo gestiona el tiempo y el cumplimiento de tareas su equipo de desarrollo?

Respuestas		Fr	%
1.	Project Management Office	1	16,66%
2.	Diagramas de Gantt	1	16,66%
3.	Tableros	1	16,66%
4.	A través de un programador TOP y	1	16,66%

evaluaciones periódicas		
5. Ambientes simulados y más de 2 personas a cargo de un proceso	1	16,66%
6. División de tareas y establecimiento de fechas de entrega primero a nivel interno y luego para con el cliente	1	16,66%
Totales	6	100%

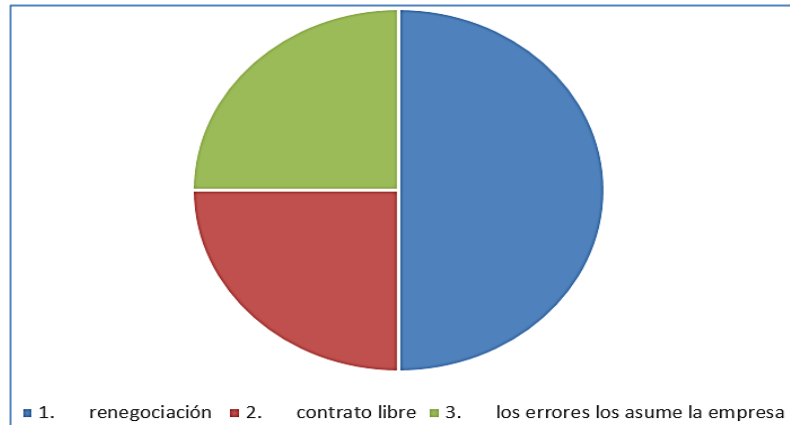


Análisis E Interpretación De Los Resultados

Los entrevistados aseguran que en su equipo de desarrollo el tiempo y cumplimiento de tareas se gestiona mediante Project Management Office, Diagramas de Gantt, Tableros, a través de un programador TOP y evaluaciones periódicas, ambientes simulados y más de 2 personas a cargo de un proceso y división de tareas y establecimiento de fechas de entrega primero a nivel interno y luego para con el cliente. Cada una de estas con un porcentaje de 16,66% respectivamente.

10. ¿Cómo resuelve una estimación errónea de costos?

Respuestas	Fr	%
1. renegociación	2	50%
2. contrato libre	1	25%
3. los errores los asume la empresa	1	25%
Totales	4	100%



Análisis E Interpretación De Los Resultados

En este caso el 50% de entrevistados aseguran que la estimación errónea de costos la resuelven mediante renegociación, mientras que el 25% manifiesta que lo realizan a través de un contrato libre o q los errores son asumidos por la empresa.

Anexo 7: Evaluación de Herramientas

Una vez establecido el caso de estudio, las herramientas Endeavour, Hnasoft y Tuleap serán sometidas a una evaluación. En la cual se gestionará el ciclo de vida del proyecto “Software Inteligente para el Procesamiento Musical de Guitarra Melódica Aplicando una Red Neuronal Supervisada”. Del cual se tomarán las características del escenario, para ubicarlas o asignarlas a los diferentes campos solicitados por las herramientas. Esta evaluación permitirá conocer las funcionalidades y características de cada herramienta, permitiendo establecer cuál de ellas ofrece una mejor gestión frente a las demás. Estableciendo el alcance o los límites de gestión de las mismas.

Endeavour

Como se puede observar la interfaz de Endeavour es muy intuitiva y se pueden crear proyectos con facilidad, asignar responsables, iteraciones, tareas, dependencias, asignar fechas, actores, casos de uso, reportes y bugs. Pero no permite establecer la metodología a usar, subtareas, release, ni historias de usuario.

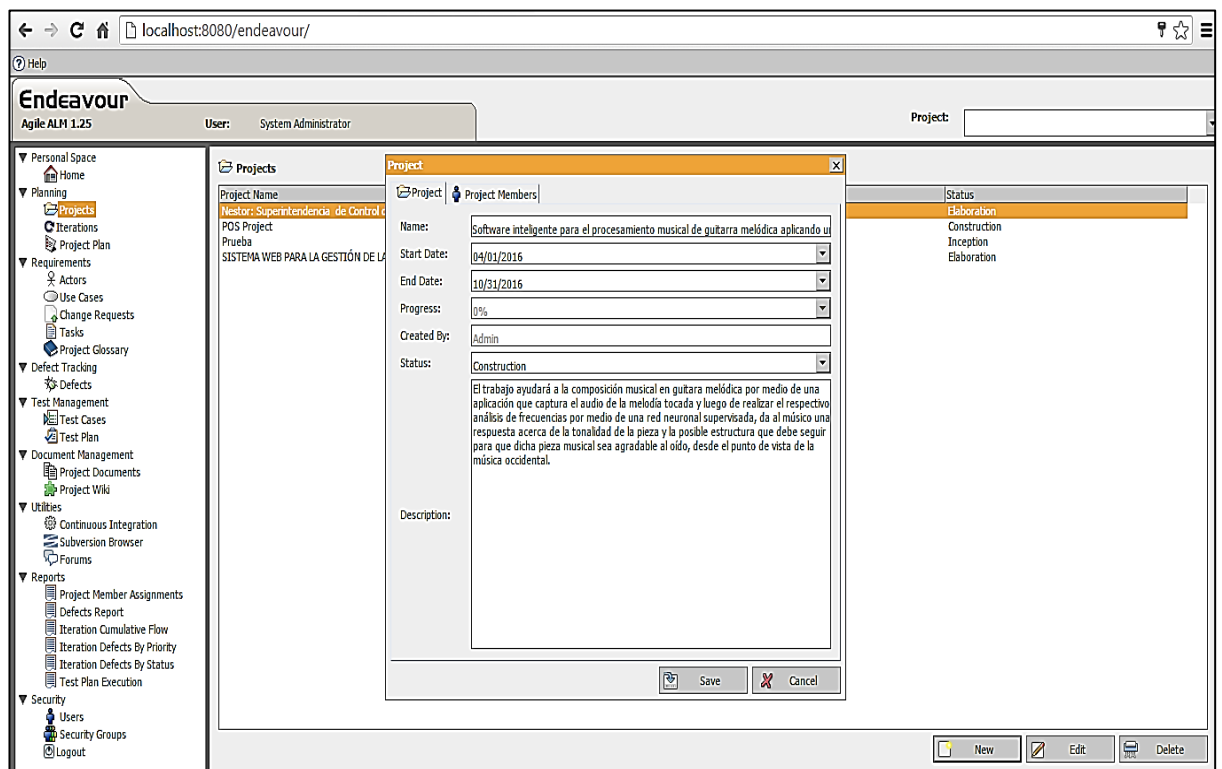


Figura 63: Creación de un nuevo proyecto.

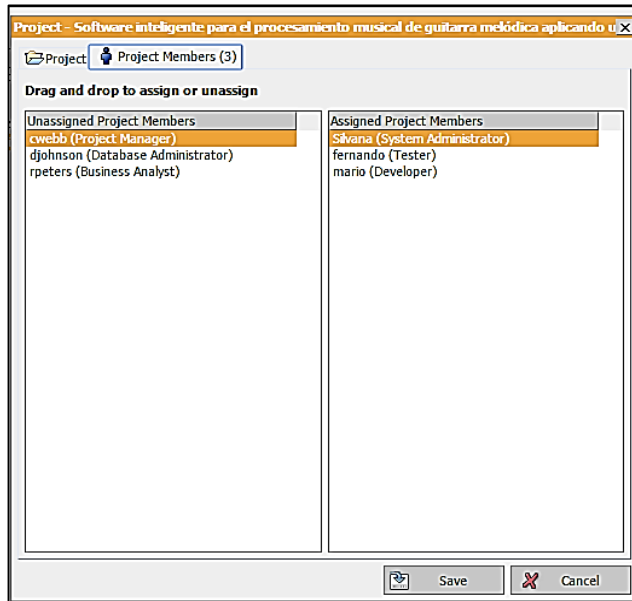


Figura 64: Asignación de miembros al proyecto.

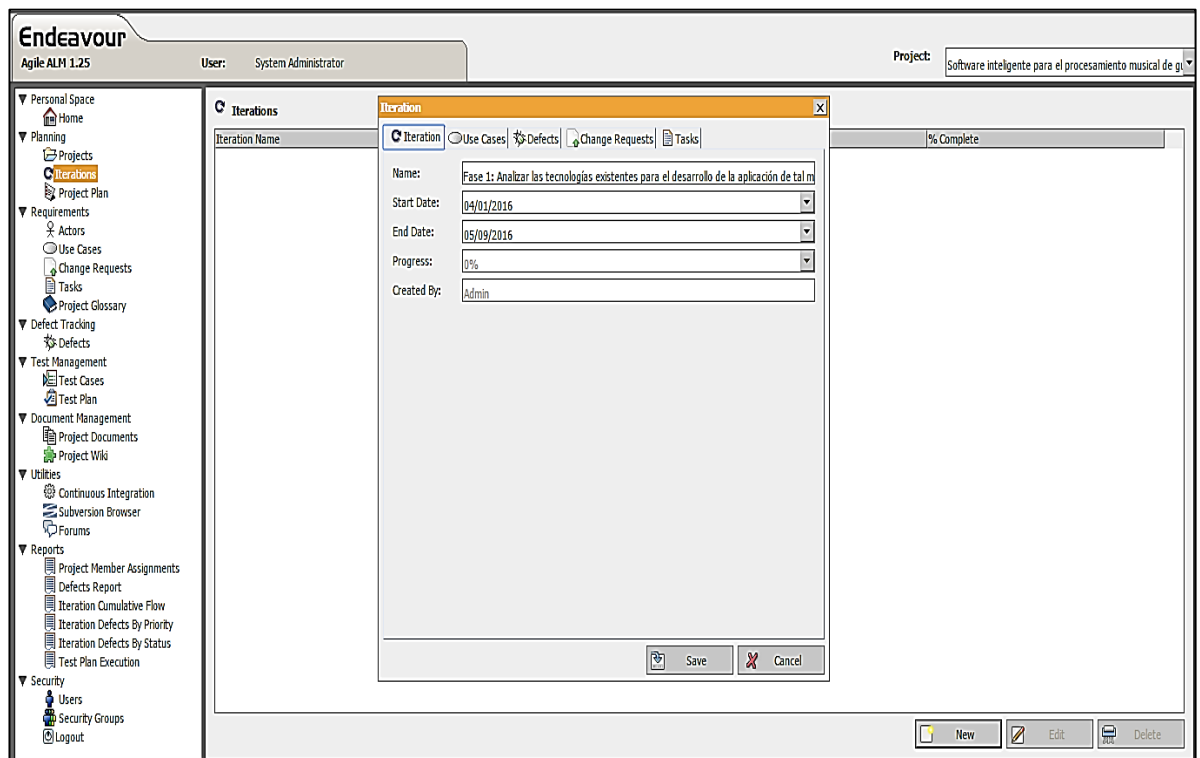


Figura 65: Creación de iteraciones del proyecto.

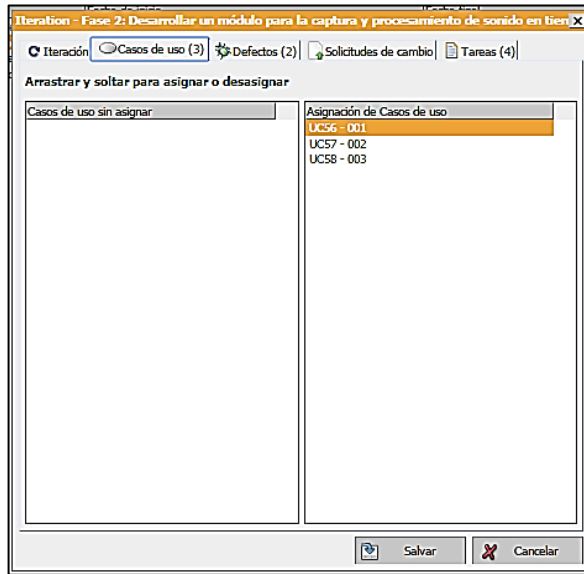


Figura 66: Asignación de casos de uso y defectos a las iteraciones del proyecto.

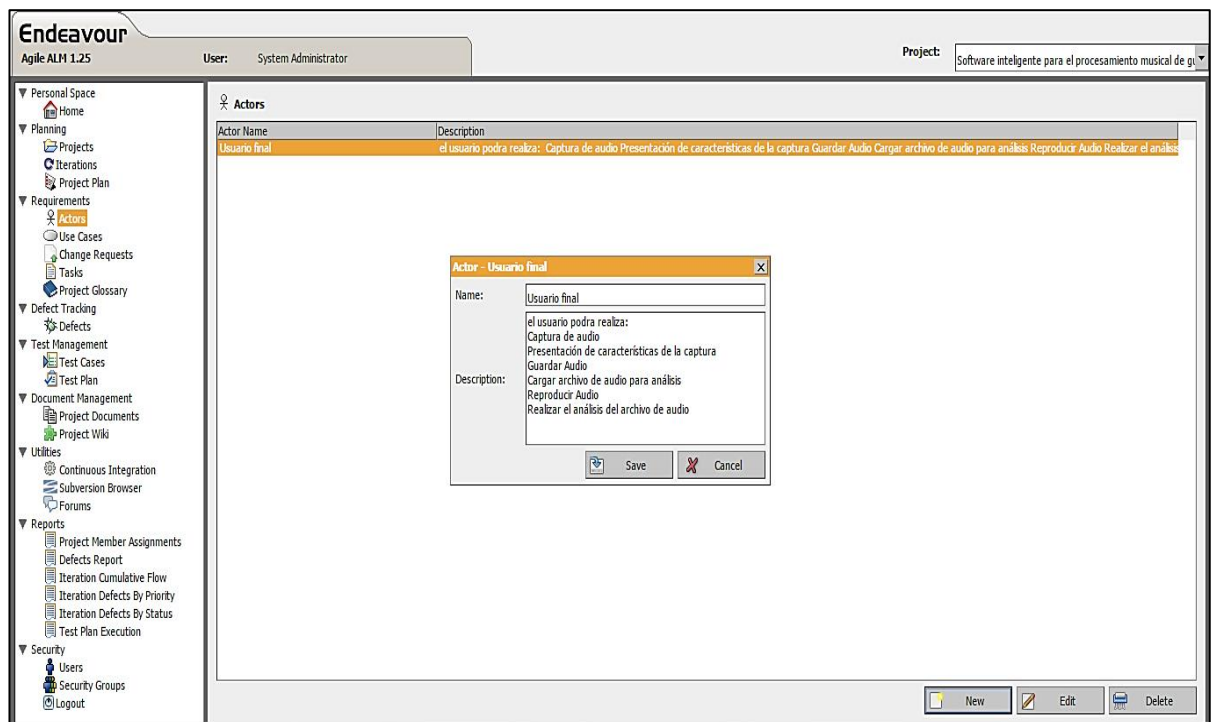


Figura 67: Creación de actores del proyecto.

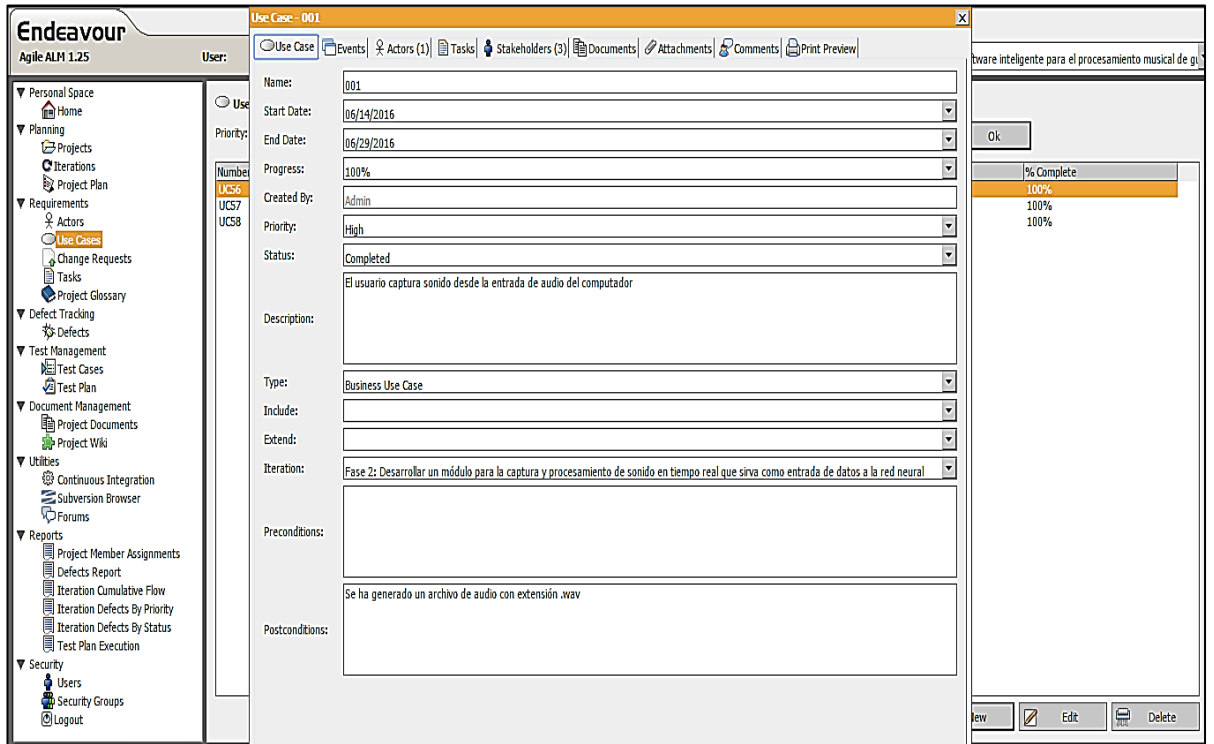


Figura 68: Creación de casos de uso del proyecto.

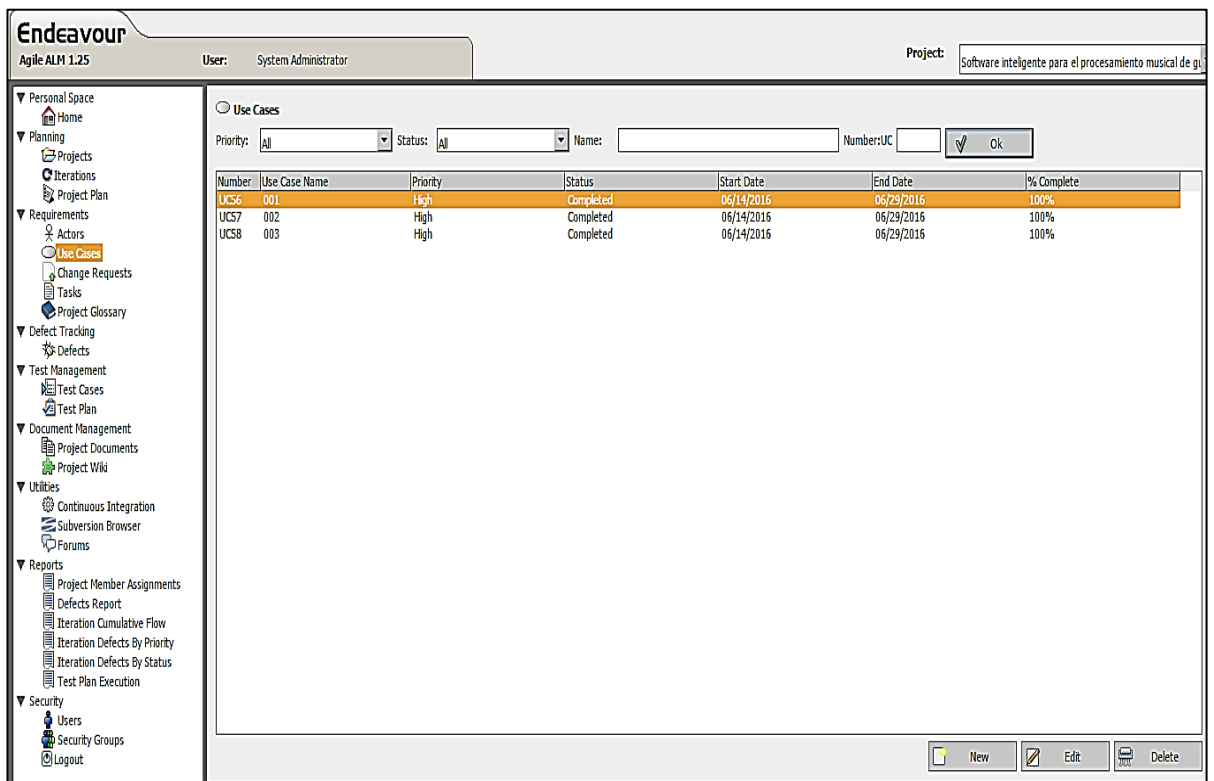


Figura 69: Visualización de todos los casos de uso del proyecto.

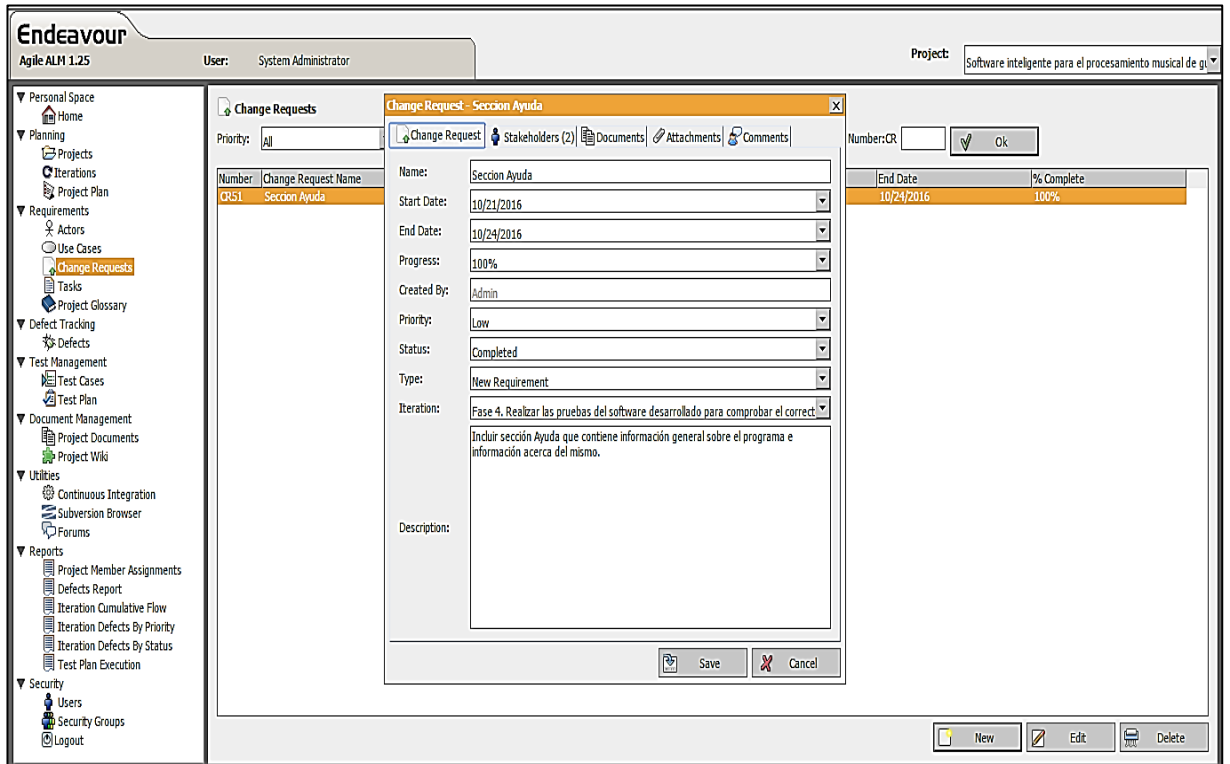


Figura 70: Cambio de requerimientos del proyecto.

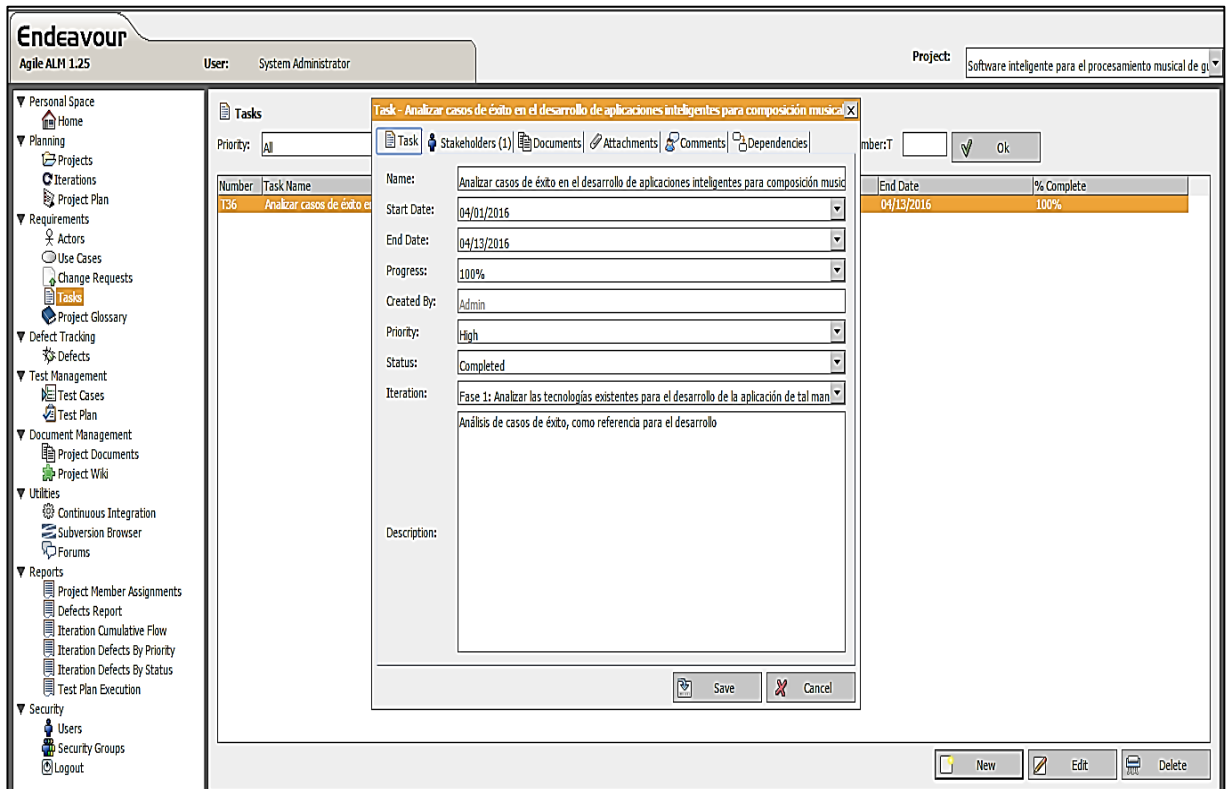


Figura 71: Creación de tareas del proyecto.

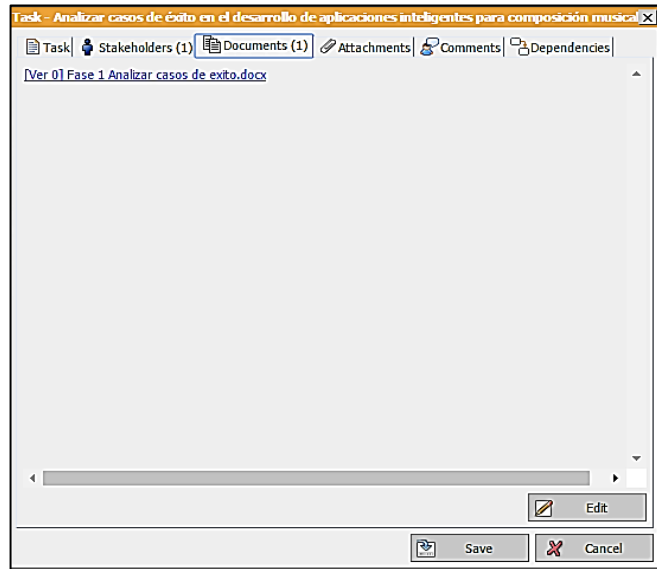


Figura 72: Asignación de documentos y stakeholders a las tareas del proyecto.

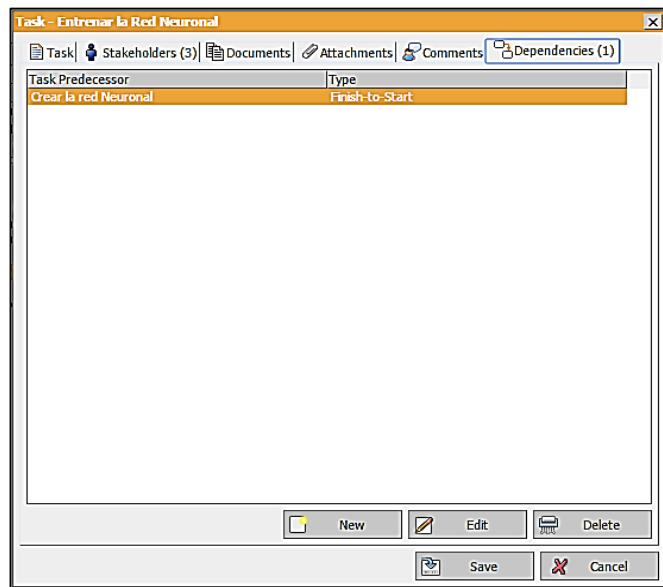


Figura 73: Asignación de dependencias a las tareas del proyecto.

Tasks

Priority: All Status: All Name: Number: T

Number	Task Name	Priority	Status	Start Date	End Date	% Complete
T36	Analizar casos de éxito en el desarri	High	Completed	04/01/2016	04/13/2016	100%
T37	Comparar herramientas para la cre	High	Completed	04/14/2016	04/27/2016	100%
T38	Seleccionar las alternativas tecnoló	High	Completed	04/28/2016	05/09/2016	100%
T39	Definir una metodología de desarro	Medium	Completed	05/10/2016	05/17/2016	100%
T40	Historias de usuario	Medium	Completed	05/18/2016	05/25/2016	100%
T41	Prototipado del sistema	High	Completed	05/26/2016	06/10/2016	100%
T42	Implementación del módulo en un I	High	Completed	06/13/2016	06/30/2016	100%
T43	Recolectar datos - Definir el tipo de	Medium	Completed	07/01/2016	07/07/2016	100%
T44	Obtención del Dataset - Obtener la	Medium	Completed	07/08/2016	07/15/2016	100%
T45	Crear la red Neuronal	High	Completed	07/18/2016	08/26/2016	100%
T46	Entrenar la Red Neuronal	High	Completed	08/29/2016	09/09/2016	100%
T47	Validar la Red Neuronal	High	Completed	09/10/2016	09/16/2016	100%
T48	Usar la Red Neuronal - Evaluar los	High	Completed	09/19/2016	09/23/2016	100%
T49	Pruebas de interfaz	Medium	Completed	09/26/2016	10/12/2016	100%
T50	Pruebas Unitarias	High	Completed	10/13/2016	10/31/2016	100%

Figura 74: Visualización de todas las tareas del proyecto.

Endeavour Agile ALM 1.2.5 User: System Administrator Project: Software inteligente para el procesamiento musical de g

Project Glossary

Term	Description
Red Neuronal	Son inspiradas en el comportamientos de las neuronas y conexiones del cerebro humano tratando de crear un programa, sistema o maquina que sea capaz de solucionar pro

Glossary Term - Red Neuronal

Name: Red Neuronal

Created By: Silvana

Description: Son inspiradas en el comportamientos de las neuronas y conexiones del cerebro humano tratando de crear un programa sistema o maquina que sea capaz de solucionar problemas difíciles, actuar de forma humana , y realizar trabajos pesados mediante técnicas algorítmicas convencionales

Save Cancel

New Edit Delete

Figura 75: Ingreso de las palabras del glosario del proyecto.

Defect - Entrenamiento de la red Neuronal

Defect | Stakeholders (1) | Documents | Attachments | Comments

Name: Entrenamiento de la red Neuronal

Start Date: 09/01/2016

End Date: 09/08/2016

Progress: 100%

Created By: Admin

Priority: High

Status: Completed

Iteration: Fase 3. Implementar una red neuronal de acuerdo a un modelo de aprendizaje

Description: Datos obtenidos tras entrenar la red neuronal no satisfacen la solución del problema

Save Cancel

Figura 76: Creación de bugs del proyecto.

Endeavour
Agile ALM 1.25

User: System Administrator

Project: Software inteligente para el procesamiento musical de gu

Defects High Medium Low

Priority: All Status: All Name: Number: D Ok

Number	Priority	Status	Defect Name	Description
D52	High	Completed	Entrenamiento de la red Neuronal	Datos obtenidos tras entrenar la red neuronal no satisfacen la solución del problema
D53	High	Completed	Captura de Audio	El formato mp3 para grabar un archivo de audio genera pérdida de frecuencias de sonido.
D54	High	Completed	Procesamiento de Sonido	El procesamiento de las frecuencias de sonido consume bastante procesador.
D55	Medium	Completed	Presentación de resultados	La presentación de los resultados del entrenamiento de la red neuronal en la interfaz gráfica del programa puede ser inexacta debido a la

New Edit Delete

Figura 77: Visualización de los bugs del proyecto.

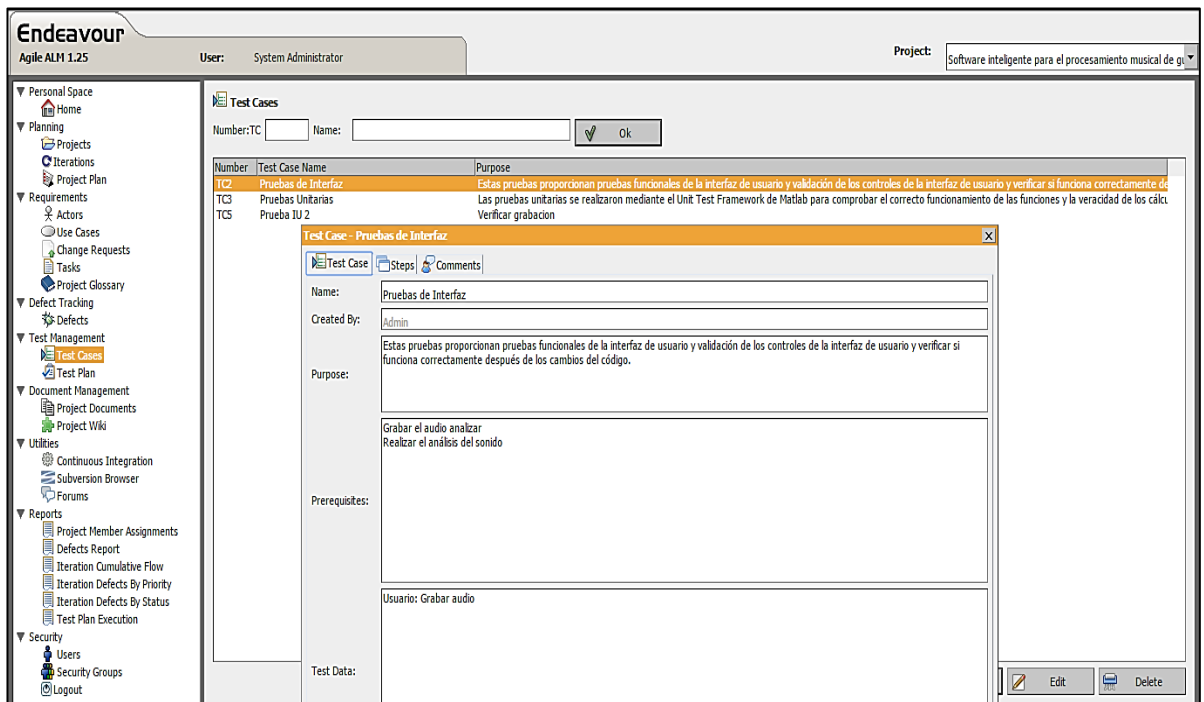


Figura 78: Creación de Pruebas del proyecto.

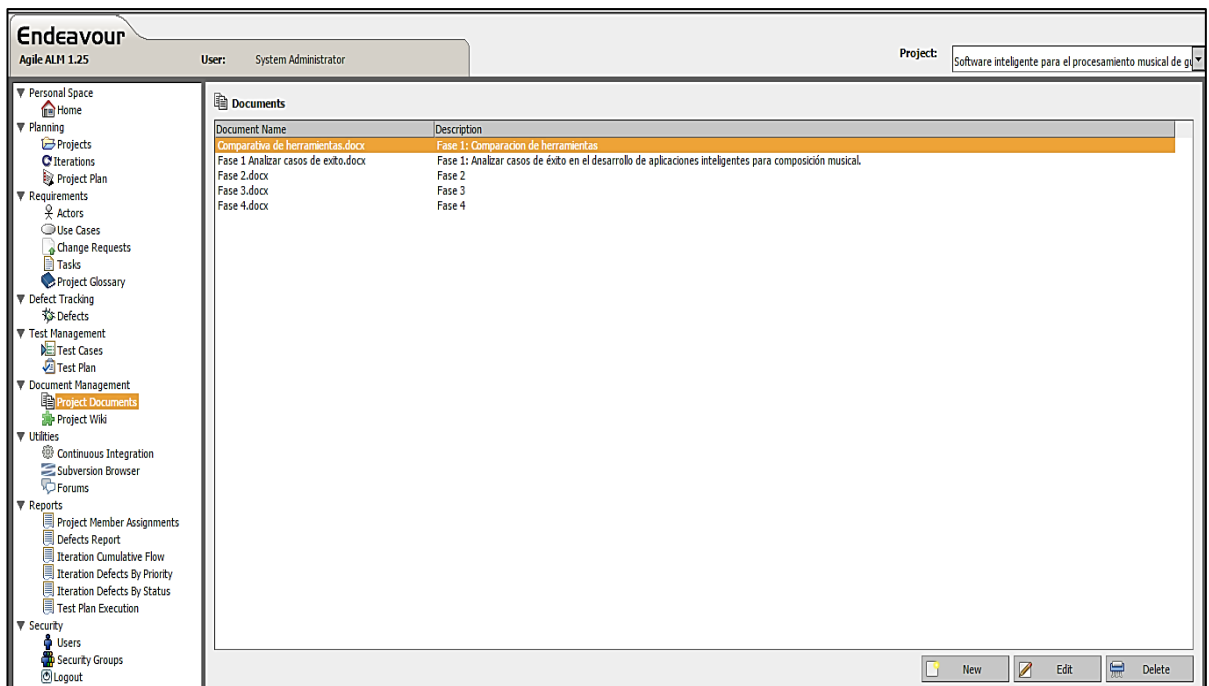


Figura 79: Documentación del proyecto.

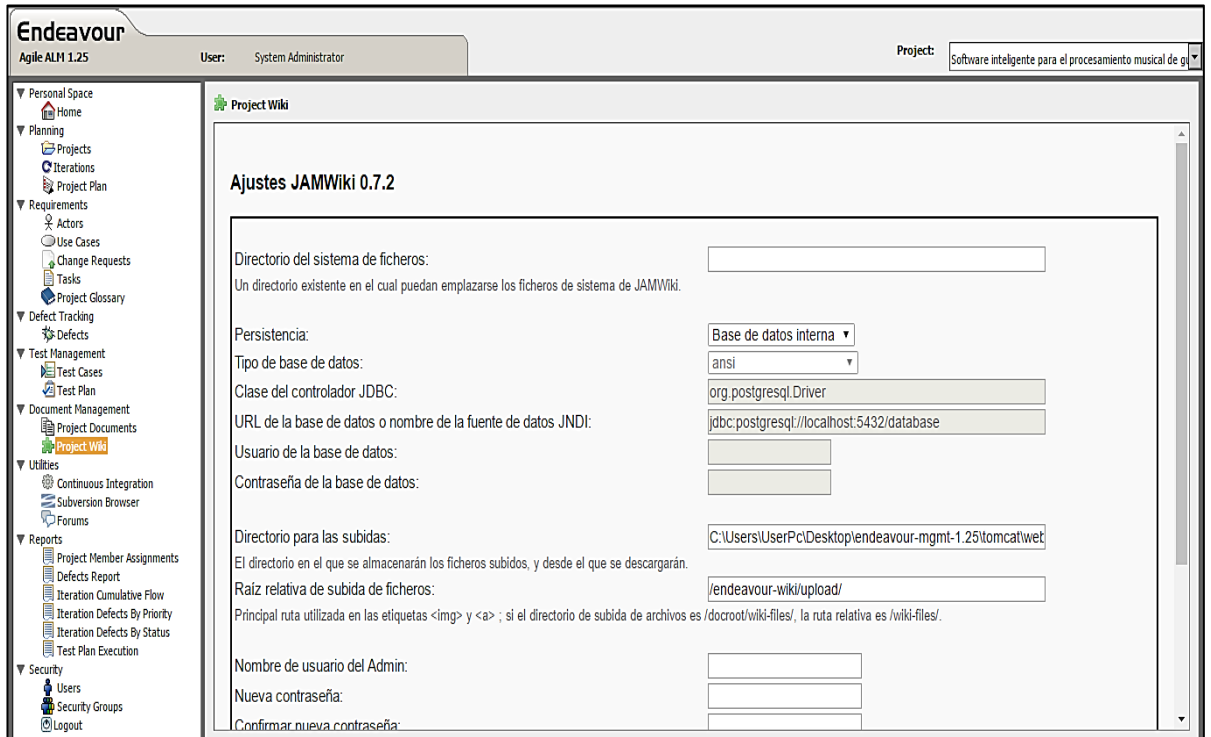


Figura 80: Creación de Wikis del proyecto.

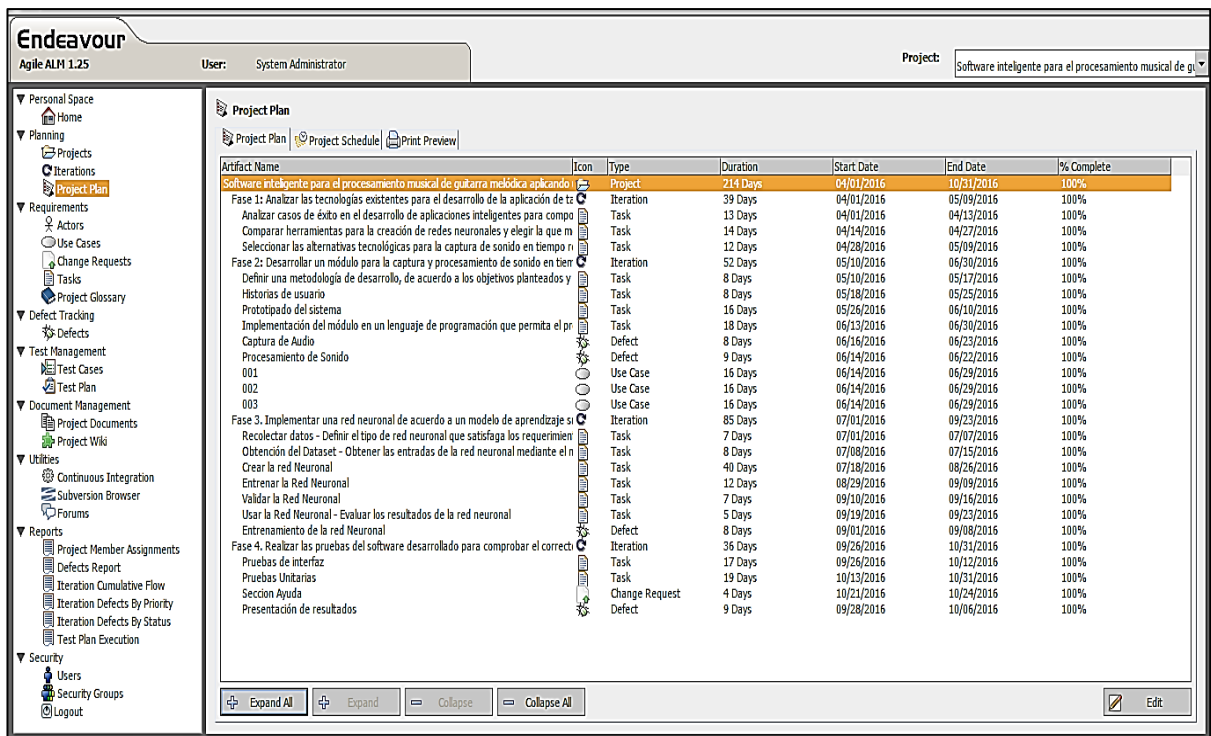


Figura 81: Visualización total del proyecto.

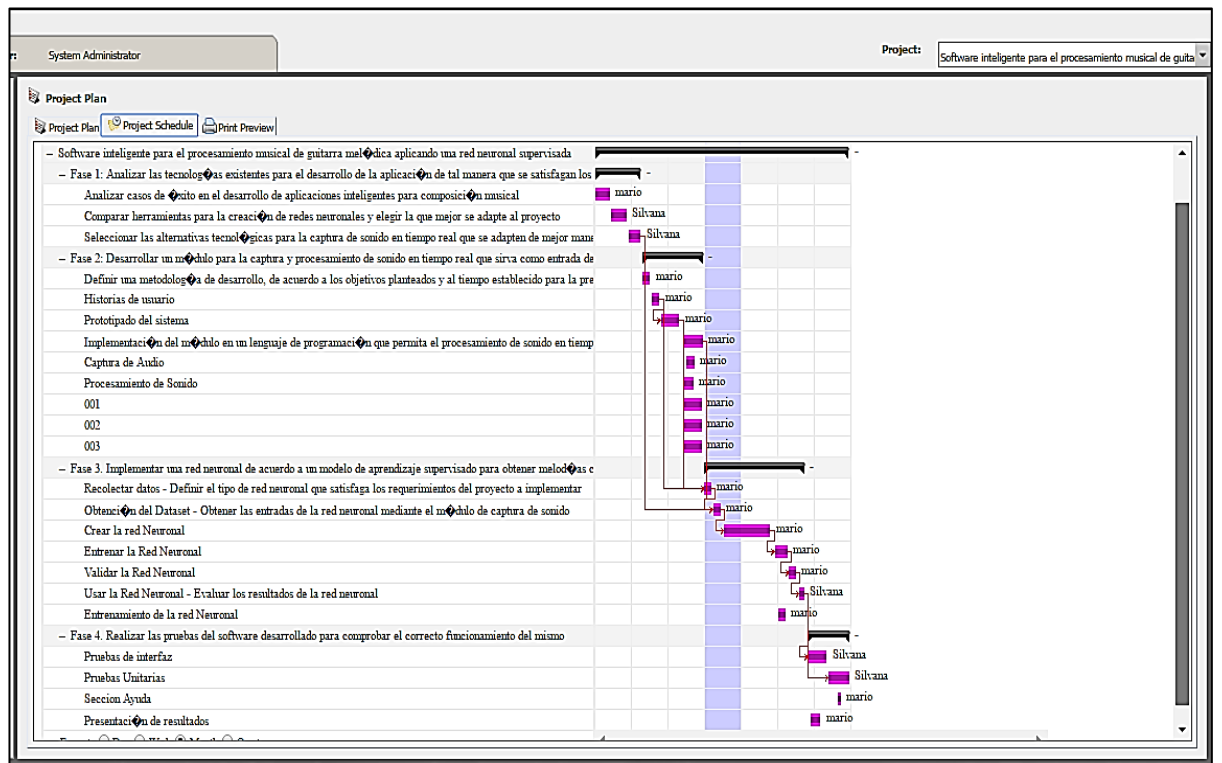


Figura 82: Visualización del cronograma del proyecto.

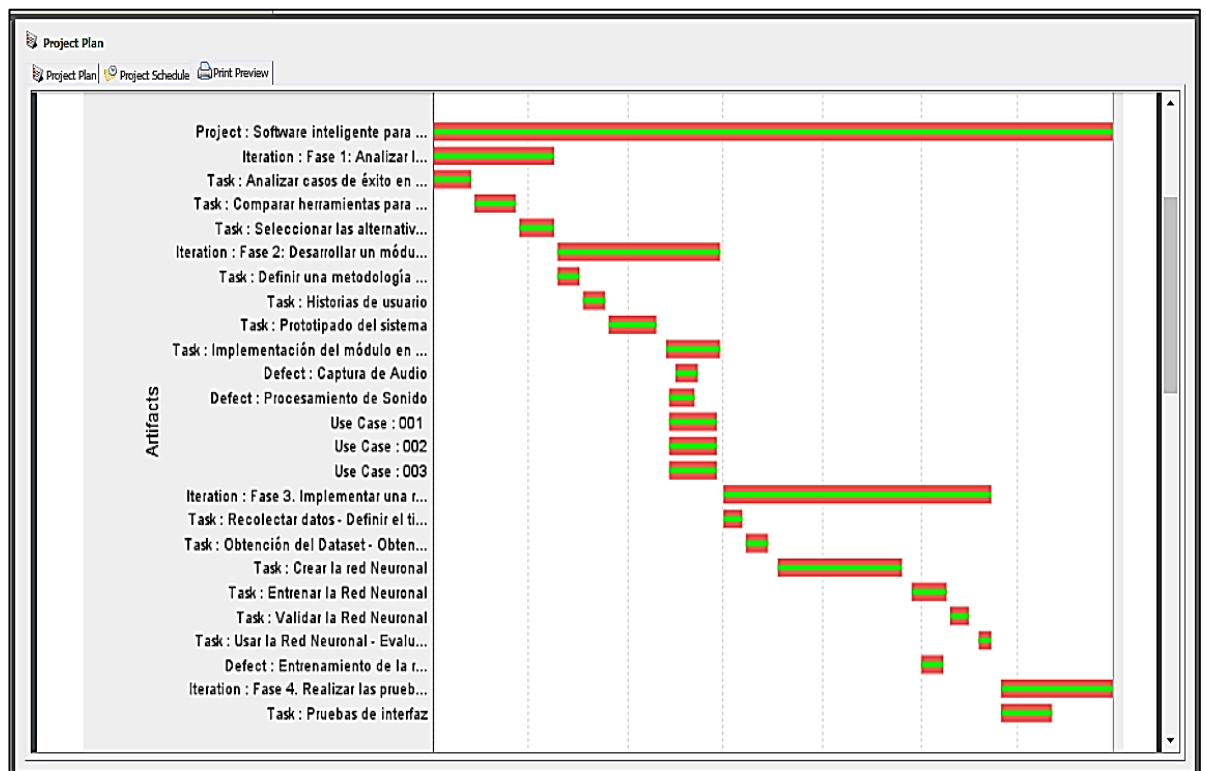


Figura 83: Vista previa de impresión del proyecto.

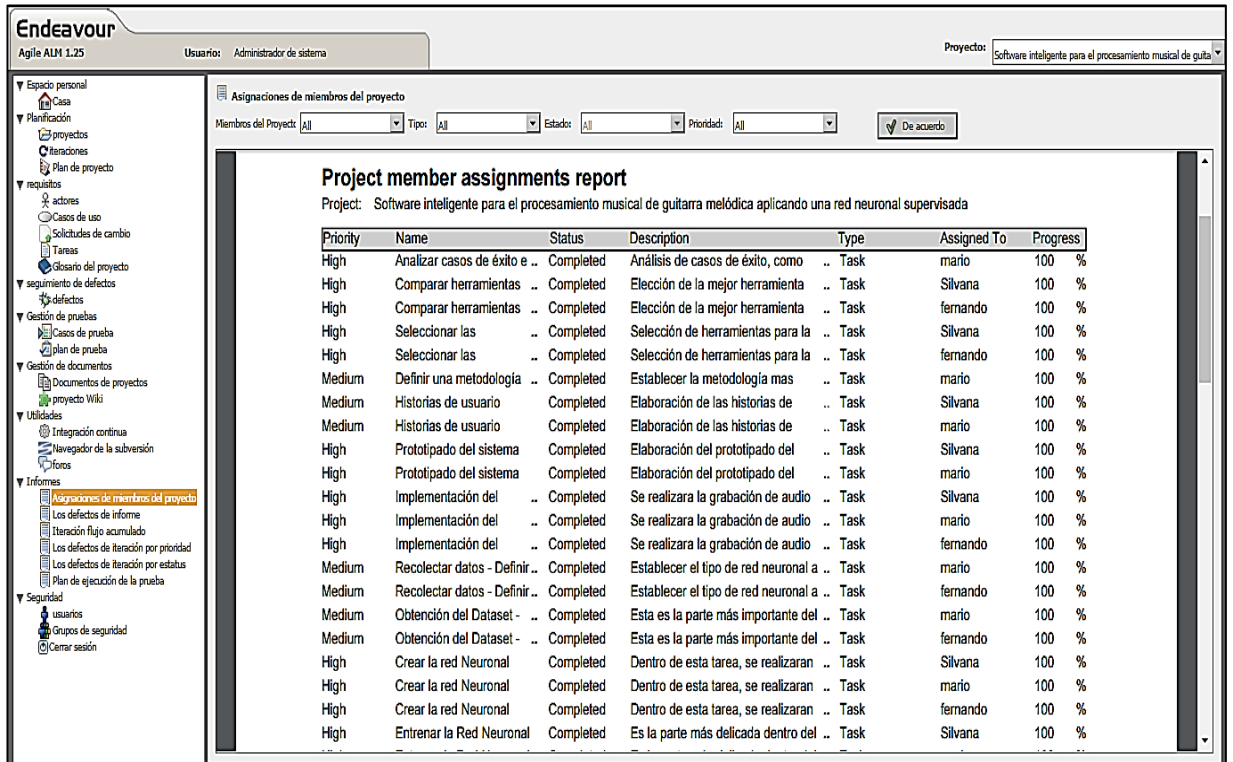


Figura 84: Reporte de asignación de miembros al proyecto.

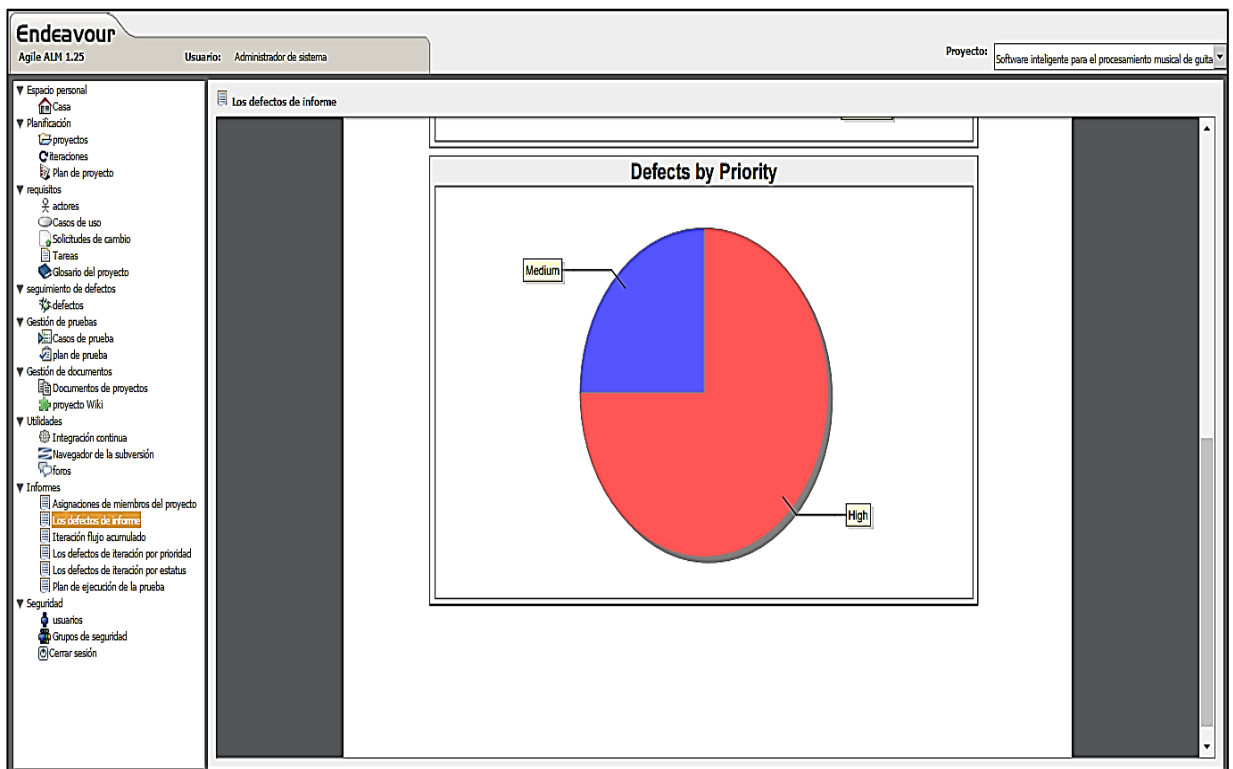


Figura 85: Reporte de defectos del proyecto.

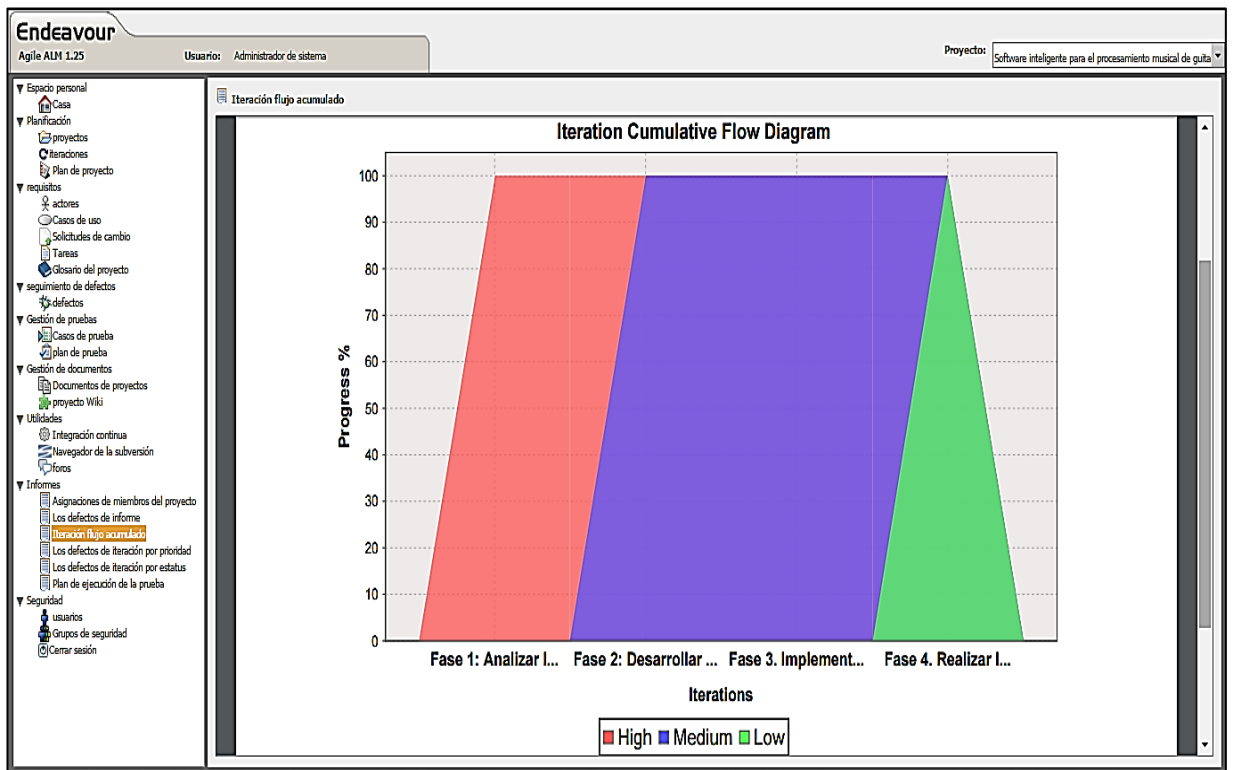


Figura 86: Reporte de iteración de flujo acumulado del proyecto.

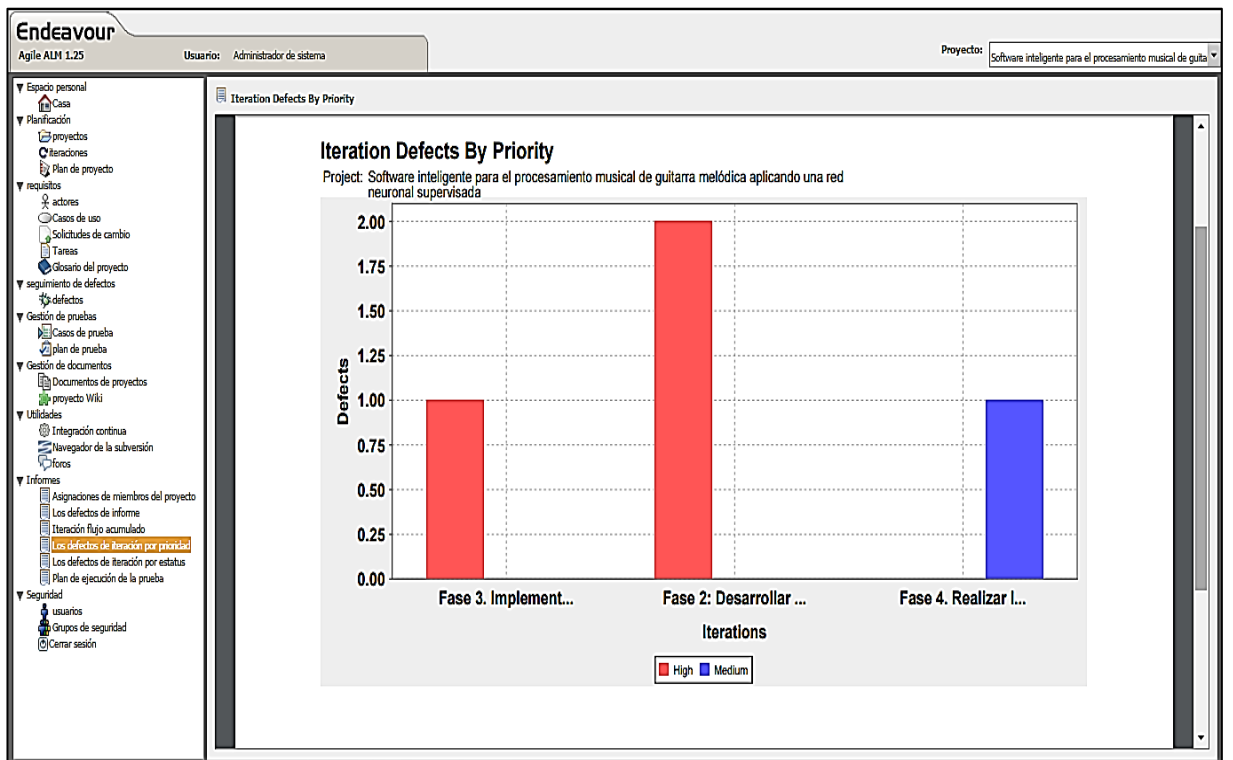


Figura 87: Reporte de defectos de iteración por prioridad del proyecto.

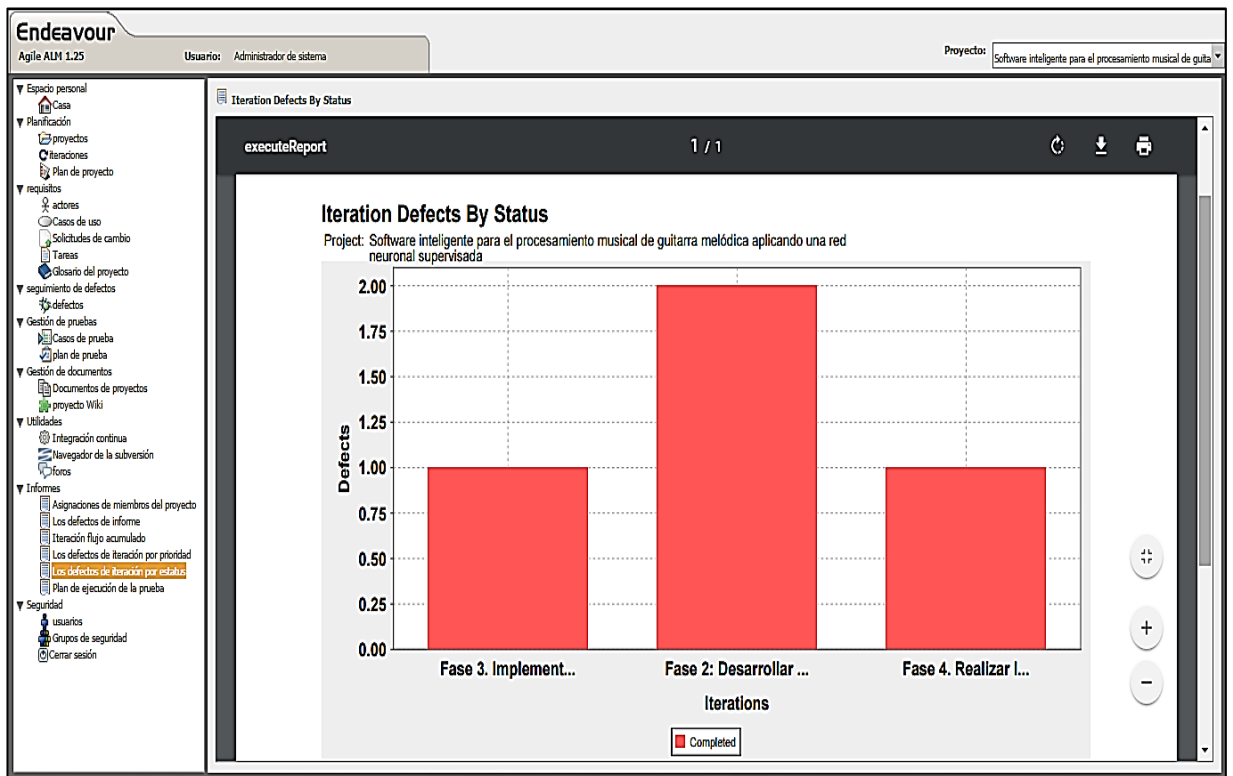


Figura 88: Reporte de defectos de iteración por status del proyecto.

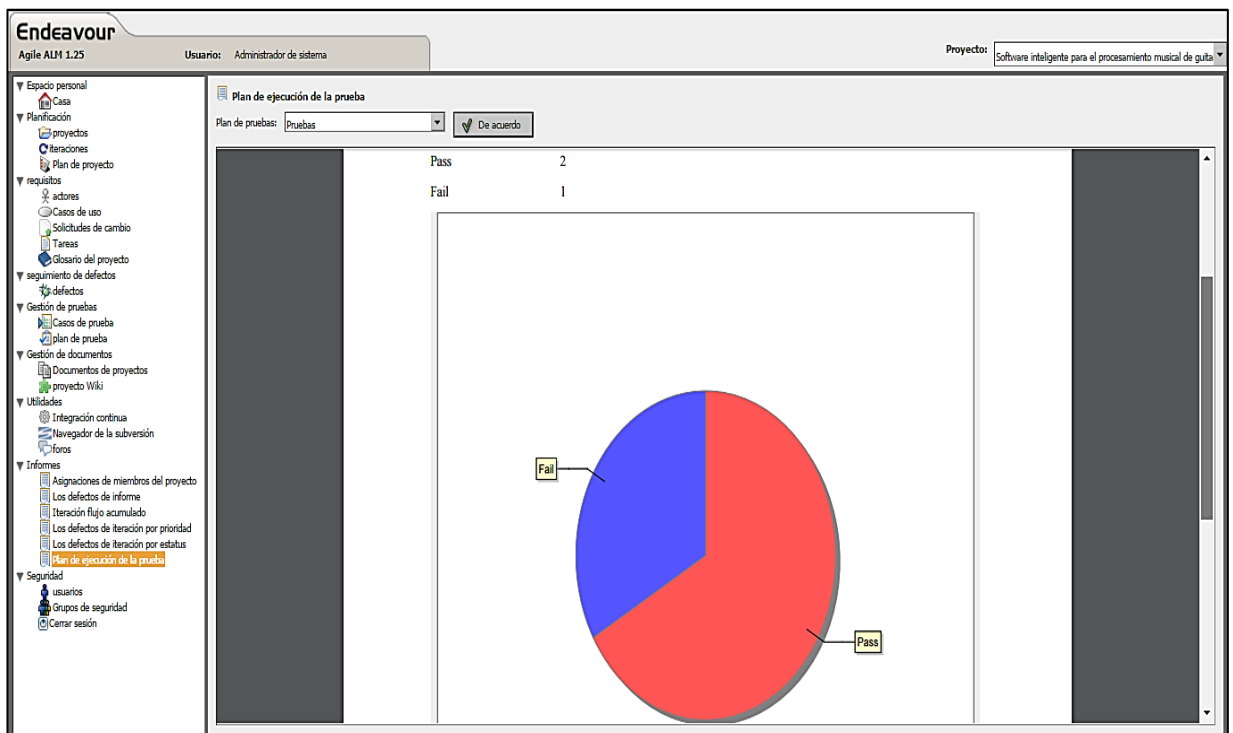


Figura 89: Reporte del plan de ejecución del proyecto.

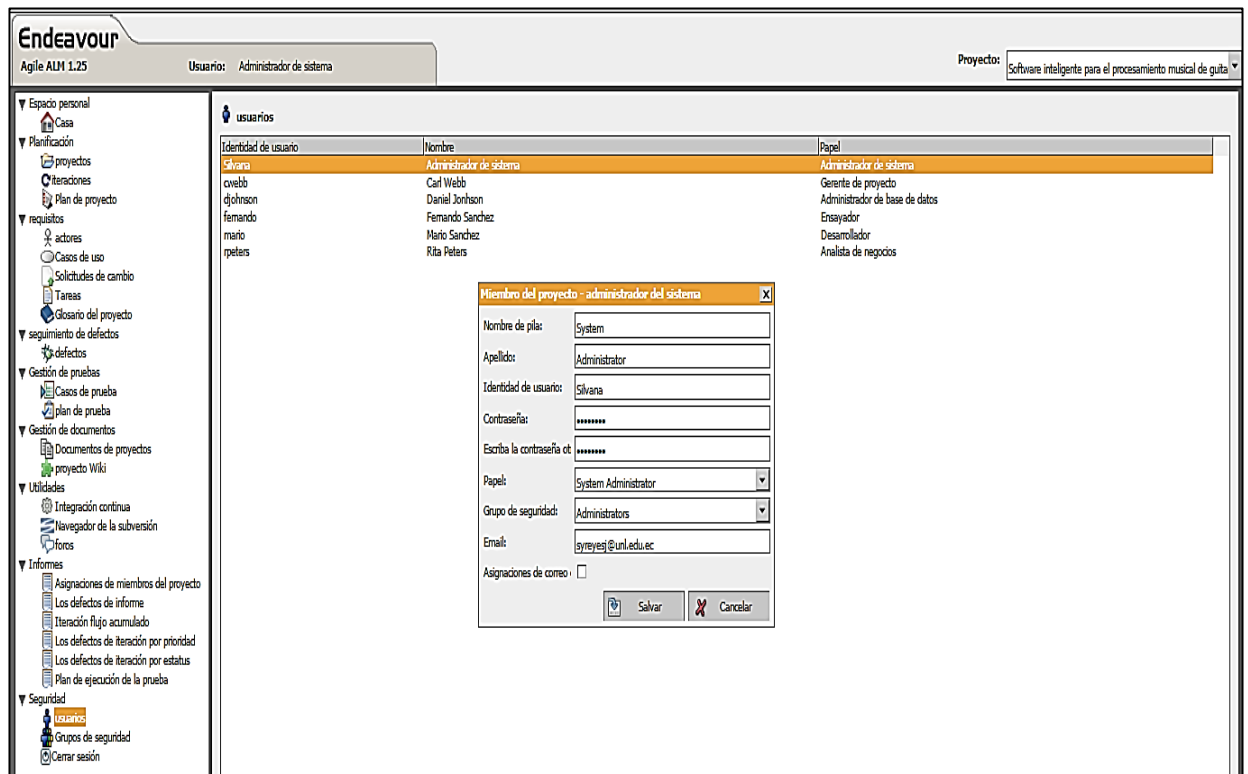


Figura 90: Creación de usuarios del proyecto.



Figura 91: Soporte de Endeavour en su página oficial.

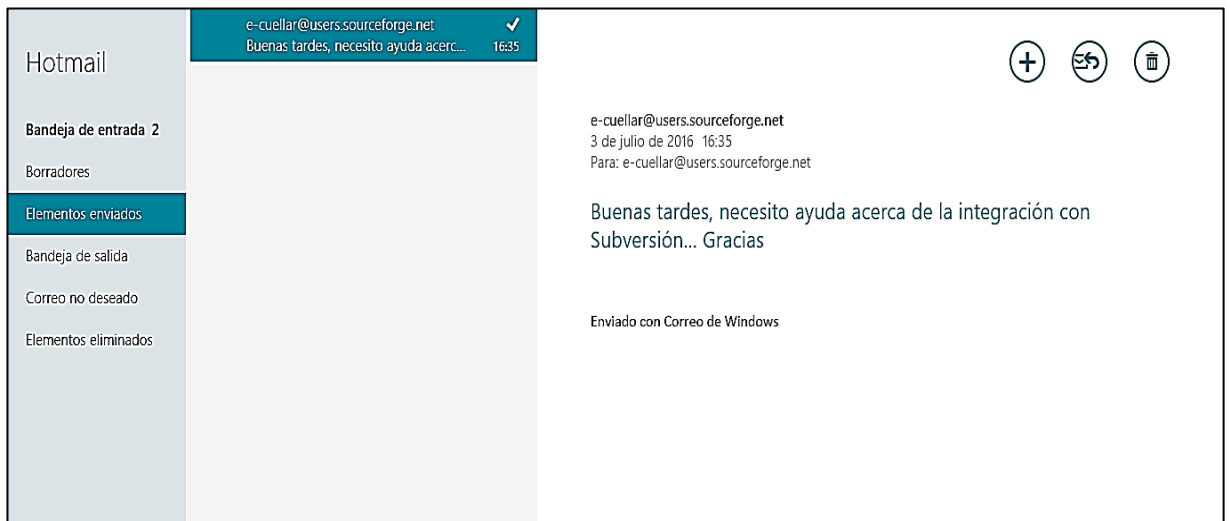


Figura 92: Soporte a través del envío de email.

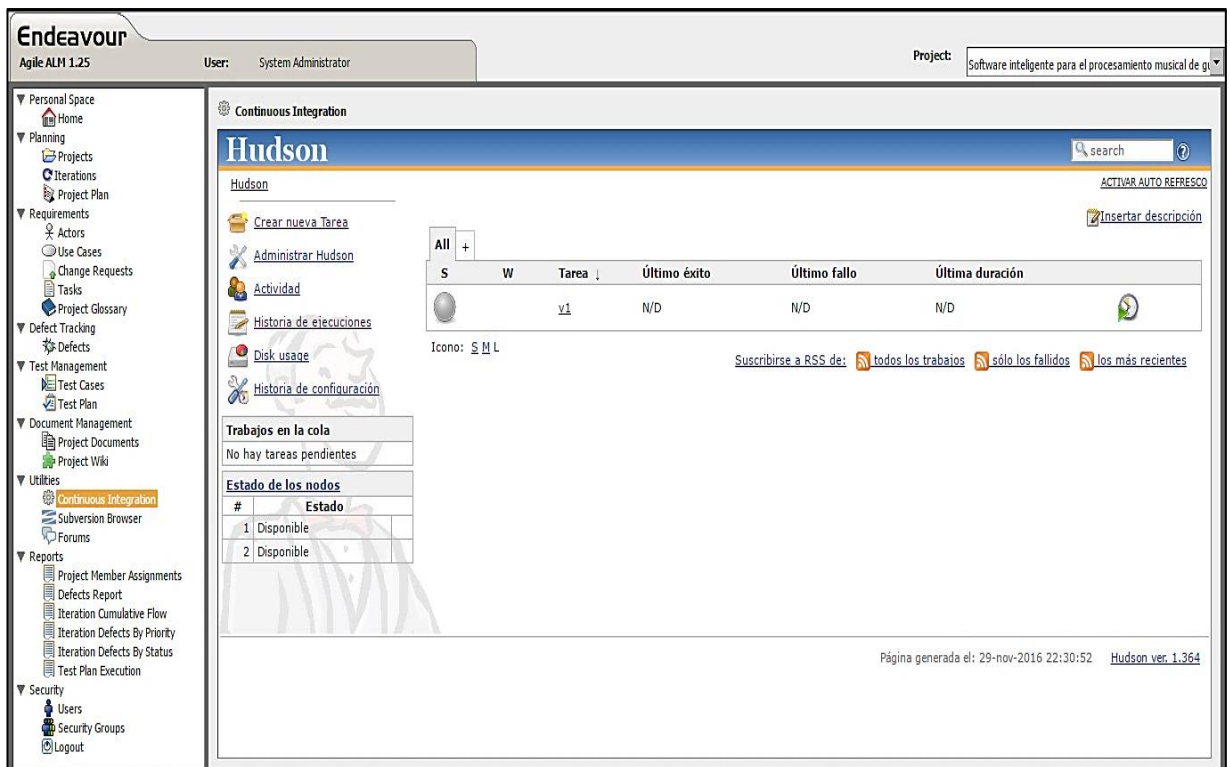


Figura 93: Versionamiento a través Hudson.

Hansoft

Por otra parte, Hansoft permite crear proyectos seleccionando la metodología a usar, o ingresando uno mismo. En este caso en particular, el proyecto se está desarrollando con la metodología XP, permite asignar responsables, iteraciones, release, tareas, asignar fechas, reportes estadísticos, chat de grupo, bugs y crear gráficas.

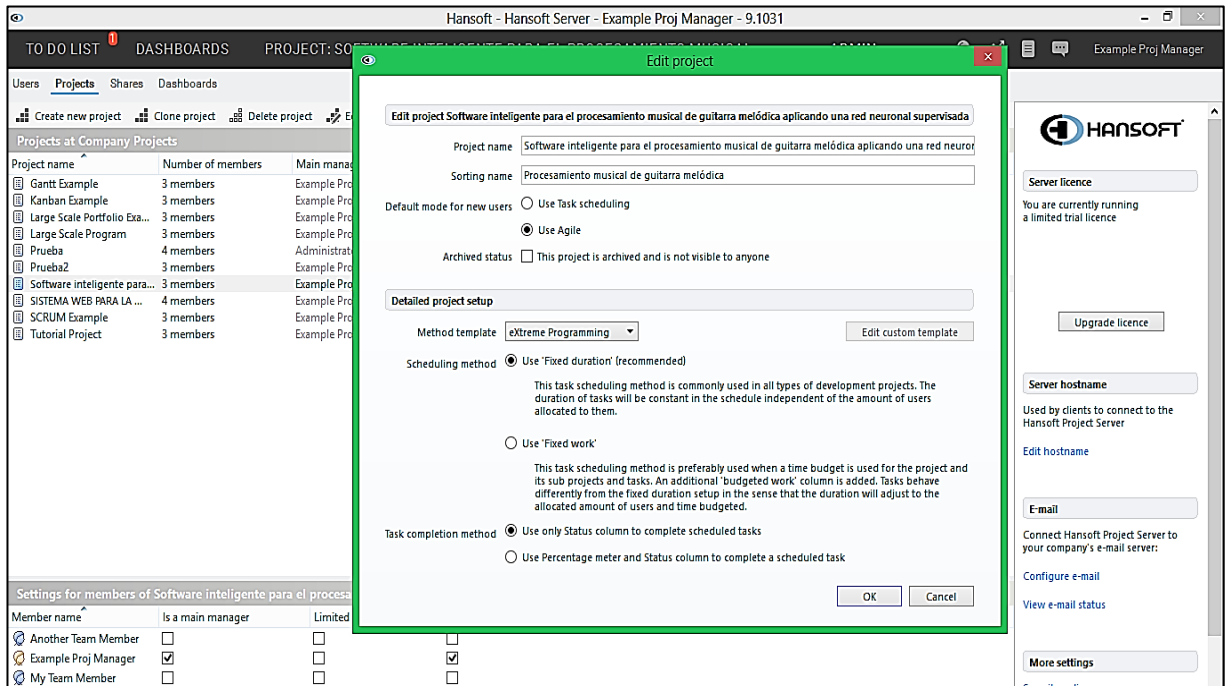


Figura 94: Creación de un nuevo proyecto.

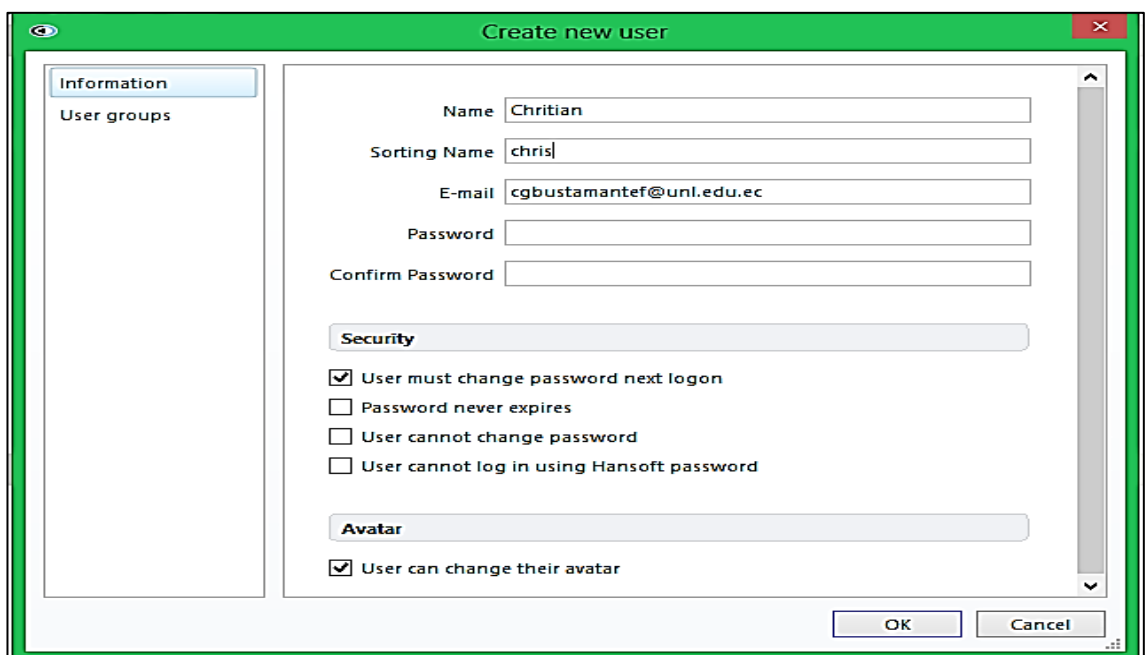


Figura 95: Creación de Usuarios.

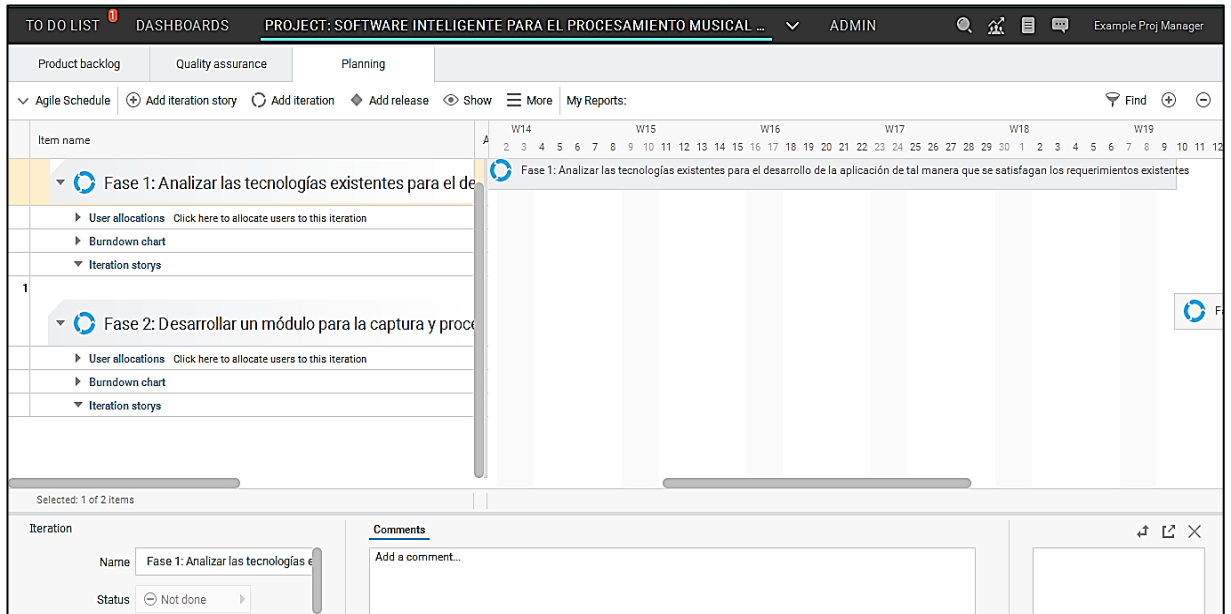


Figura 96: Creación fases o iteraciones del proyecto.

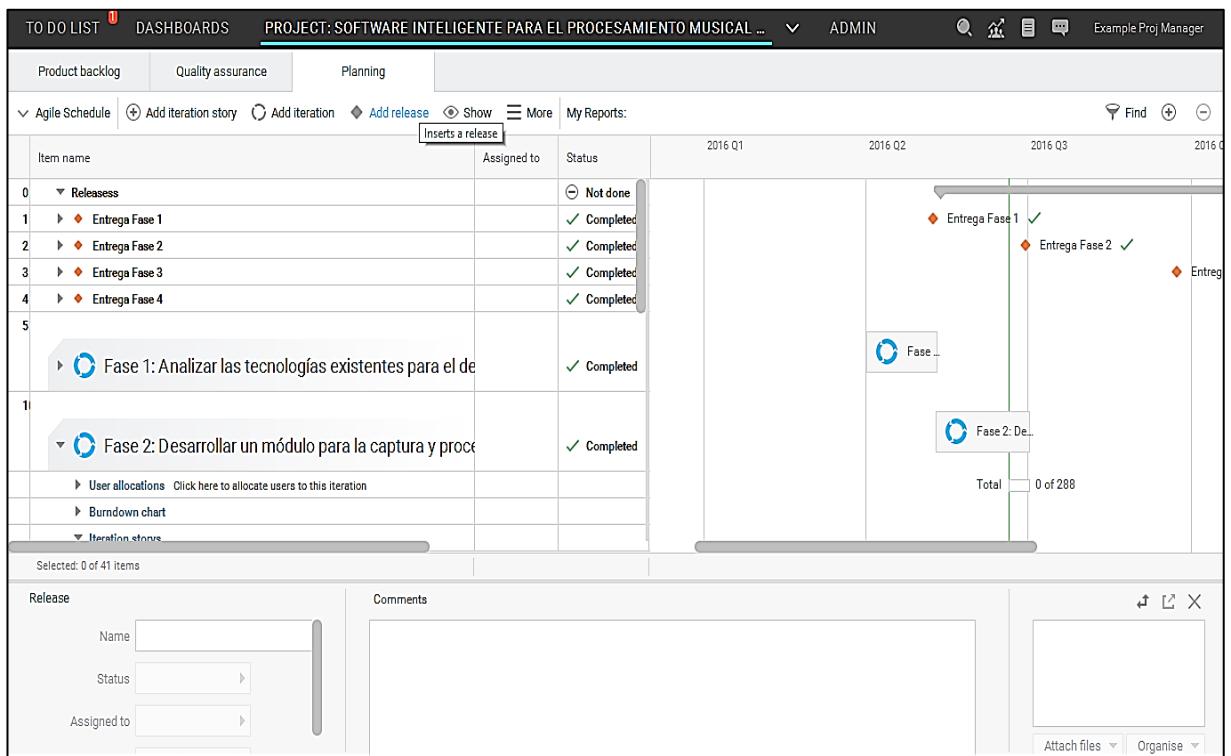


Figura 97: Creación de releases del proyecto.

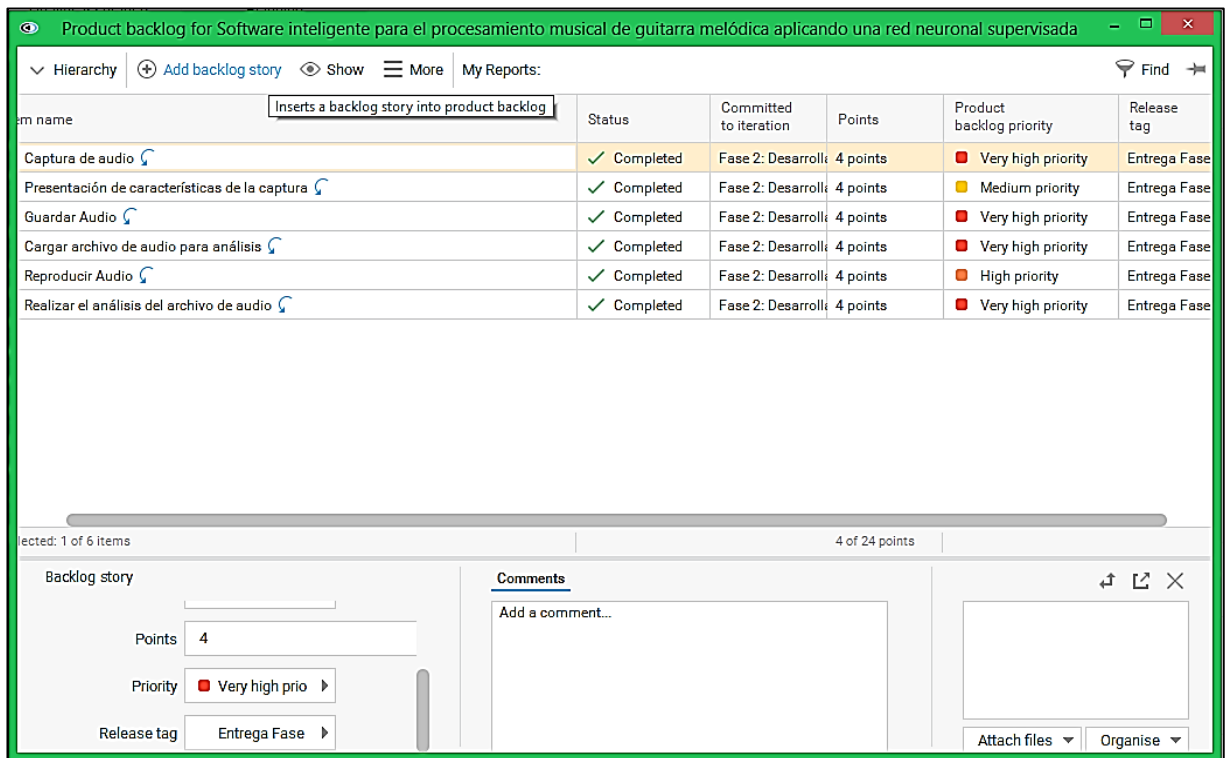


Figura 98: Backlog del proyecto.

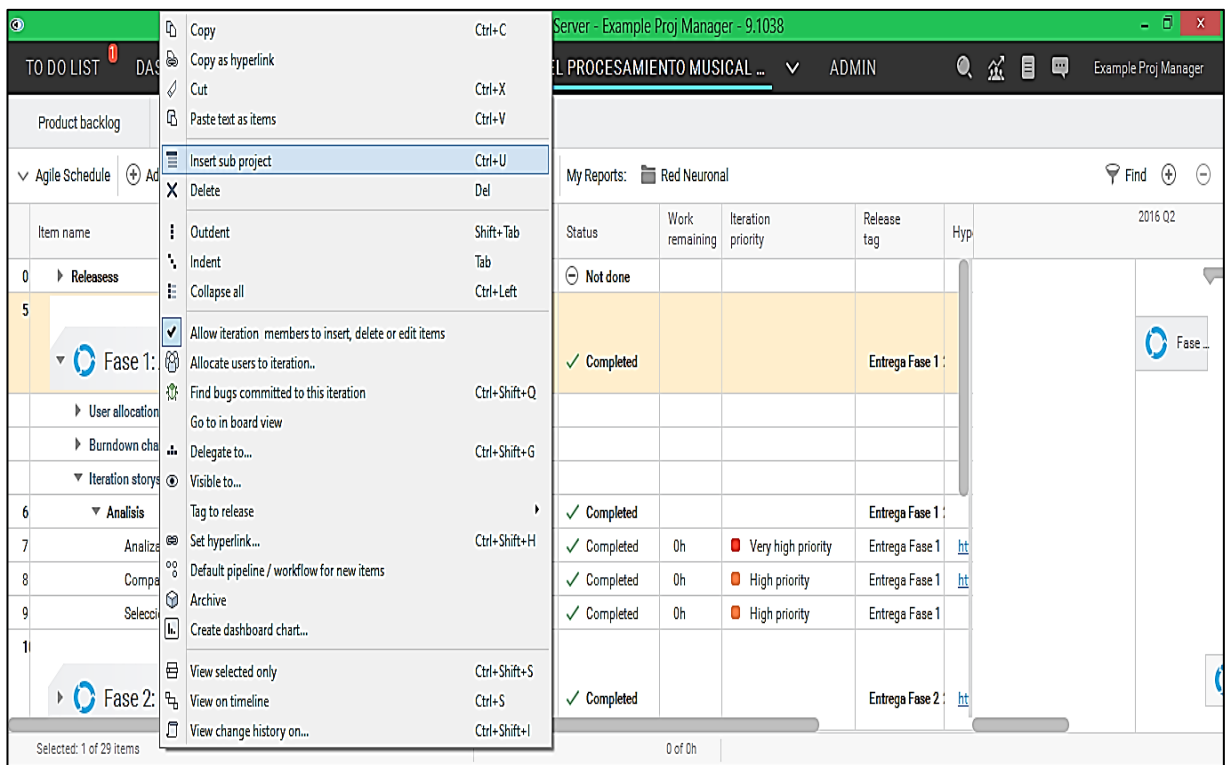


Figura 99: Creación de subproyectos o tareas del proyecto.

TO DO LIST DASHBOARDS PROJECT: SOFTWARE INTELIGENTE PARA EL PROCESAMIENTO MUSICAL ... ADMIN Example Proj Mana

Product backlog Quality assurance Planning

Agile Schedule Add iteration story Add iteration Add release Show More My Reports: Find

Item name	Assigned to	Status	Work remaining	Iteration priority	Release tag	2016 Q1
Iteration stories						
1 Realizar el análisis del archivo de audio	Fernando	Completed	0h	Very high priority	Entrega Fase 2 2016-06-30	
1 Reproducir Audio	Mario	Completed	0h	Medium priority	Entrega Fase 2 2016-06-30	
1 Cargar archivo de audio para análisis	Mario	Completed	0h	Very high priority	Entrega Fase 2 2016-06-30	
1 Guardar Audio	Mario	Completed	0h	Very high priority	Entrega Fase 2 2016-06-30	
1 Presentación de características de la captura	Mario	Completed	0h	High priority	Entrega Fase 2 2016-06-30	
1 Captura de audio	Mario	Completed	0h	Very high priority	Entrega Fase 2 2016-06-30	
Captura y procesamiento de sonido						
1 Definir una metodología de desarrollo, de acuerdo a los objetivos plante...	Mario	Completed	0h	Medium priority	Entrega Fase 2 2016-06-30	
1 Historias de usuario	Mario	Completed	0h	High priority	Entrega Fase 2 2016-06-30	
2 Prototipado del sistema	Mario	Completed	0h	Very high priority	Entrega Fase 2 2016-06-30	
Implementación del módulo en un lenguaje de programación que permit...						
2 Grabación de audio con Matlab	Mario	Completed	0h	High priority	Entrega Fase 2 2016-06-30	
2 Procesamiento Digital de la Señal (DSP)	Mario	Completed	0h	Very high priority	Entrega Fase 2 2016-06-30	

Selected: 0 of 41 items 0 of 0h

Release Name Status Assigned to

Comments Attach files Organise

Figura 100: Visualización de backlog y tareas de la Fase 1 del proyecto.

TO DO LIST DASHBOARDS PROJECT: SOFTWARE INTELIGENTE PARA EL PROCESAMIENTO MUSICAL ... ADMIN Example Proj Manager

Product backlog Quality assurance Planning

Agile Schedule Add iteration story Add iteration Add release Show More My Reports: Find

Item name	Assigned to	Status	Work remaining	Iteration priority	Release tag	2016 Q1
0 Releassess		Not done				
5 Fase 1: Analizar las tecnologías existentes para el de		Completed			Entrega Fase 1 2016-05-09	
1 Fase 2: Desarrollar un módulo para la captura y proc		Completed			Entrega Fase 2 2016-06-30	
2 Fase 3. Implementar una red neuronal de acuerdo a		Completed			Entrega Fase 3 2016-09-23	
3 Fase 4. Realizar las pruebas del software desarrollad		Completed			Entrega Fase 4 2016-10-31	

Selected: 0 of 41 items 0 of 0h

Release Name Status Assigned to

Comments Attach files Organise

Figura 101: Visualización de las fases y su estado en el proyecto.

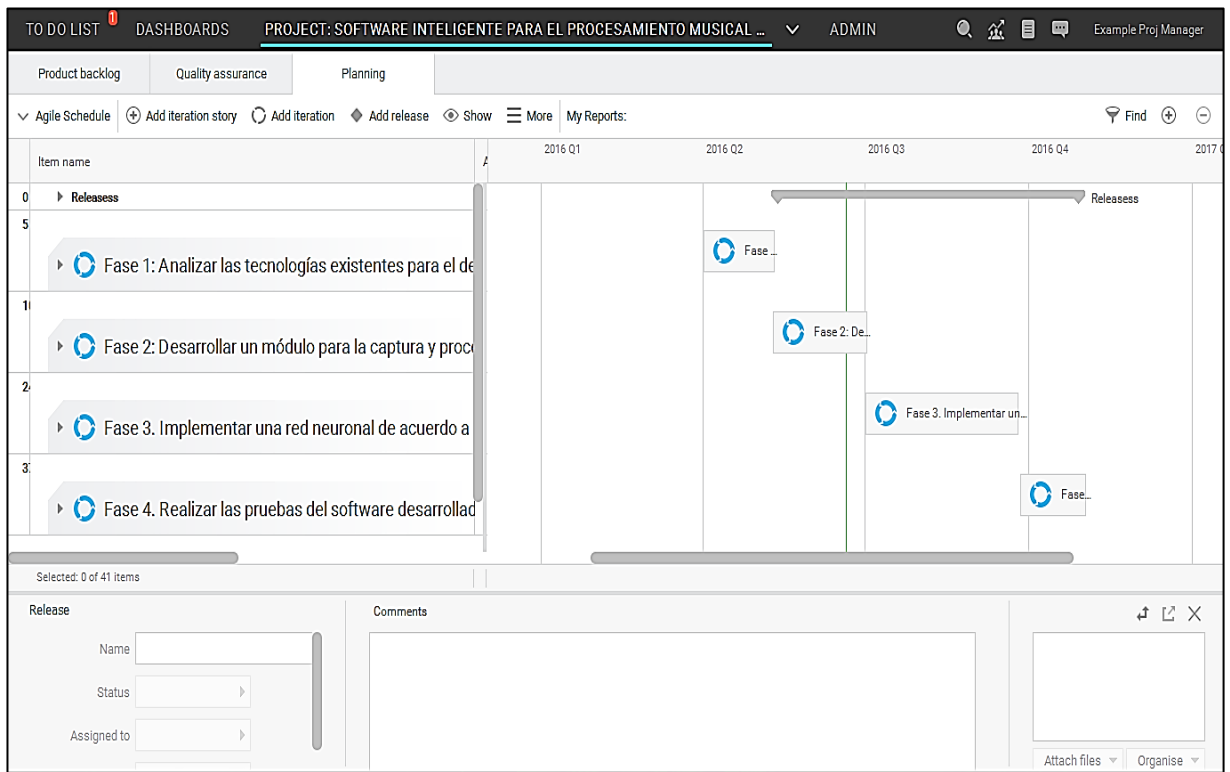


Figura 102: Visualización de los tiempos de las fases del proyecto.

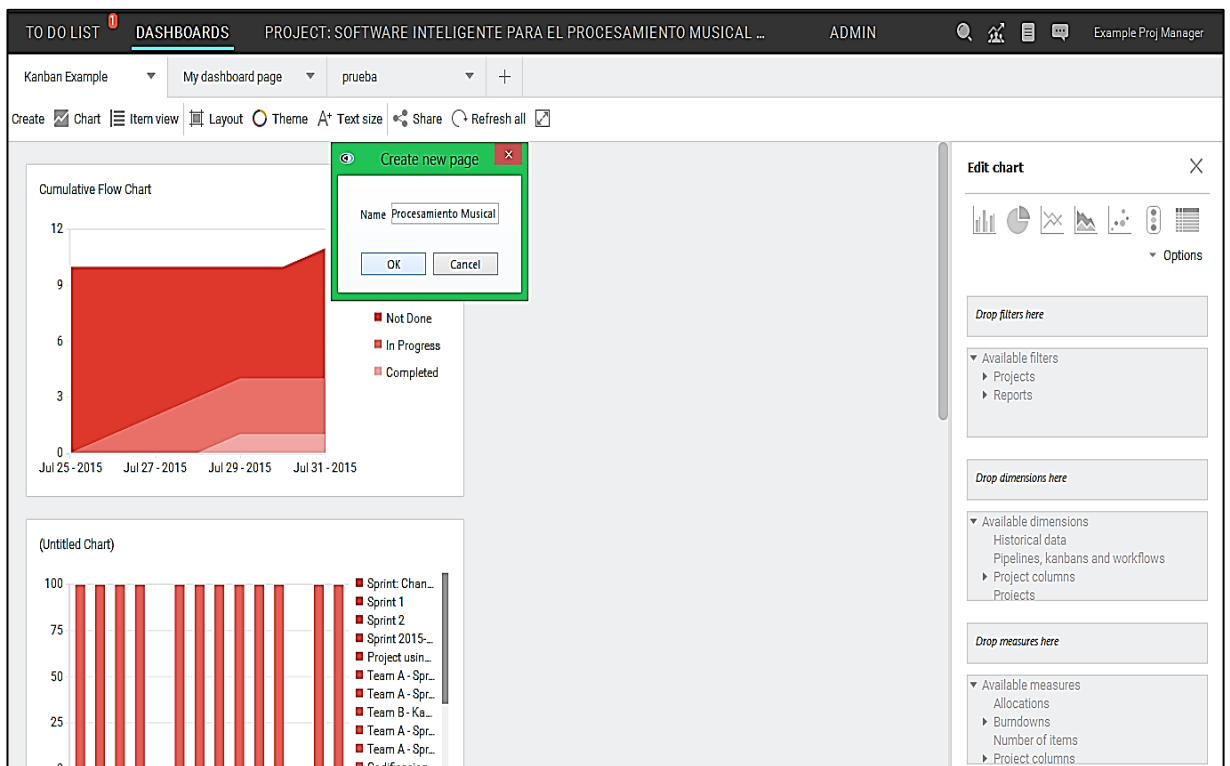


Figura 103: Creación de la página de dashboards del proyecto.

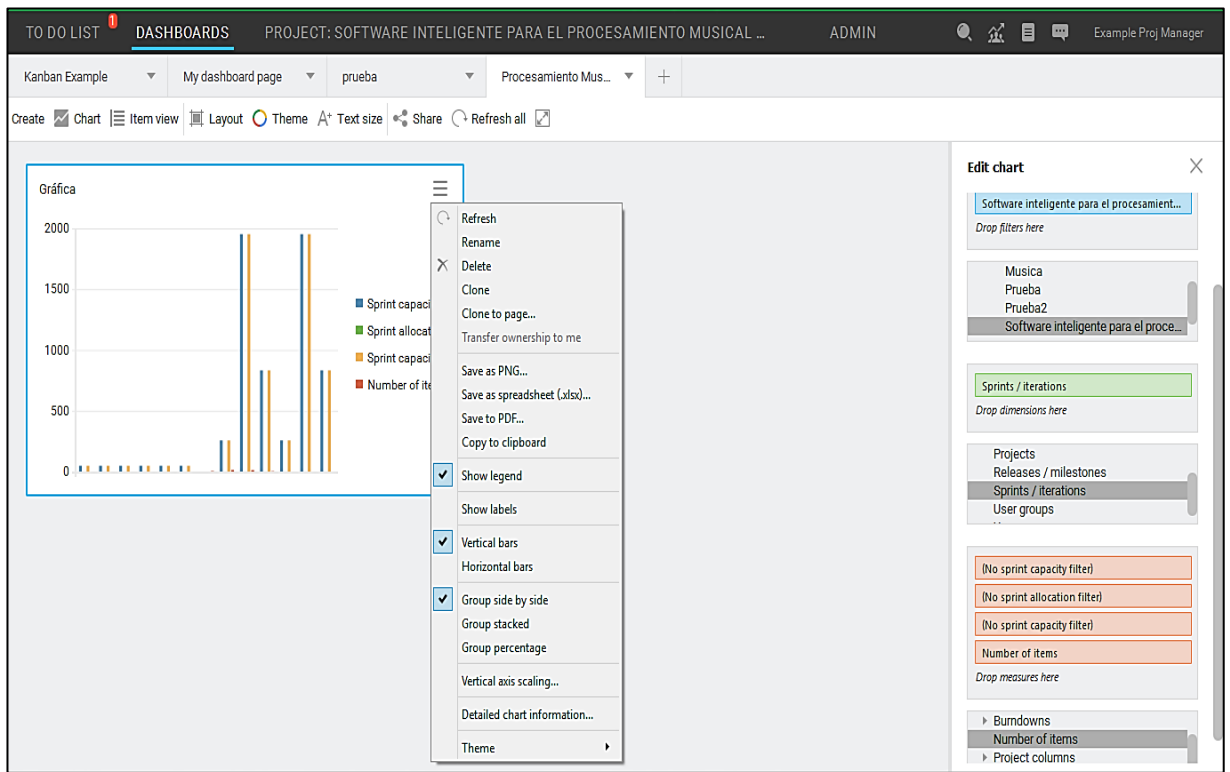


Figura 104: Creación de diagramas del proyecto.

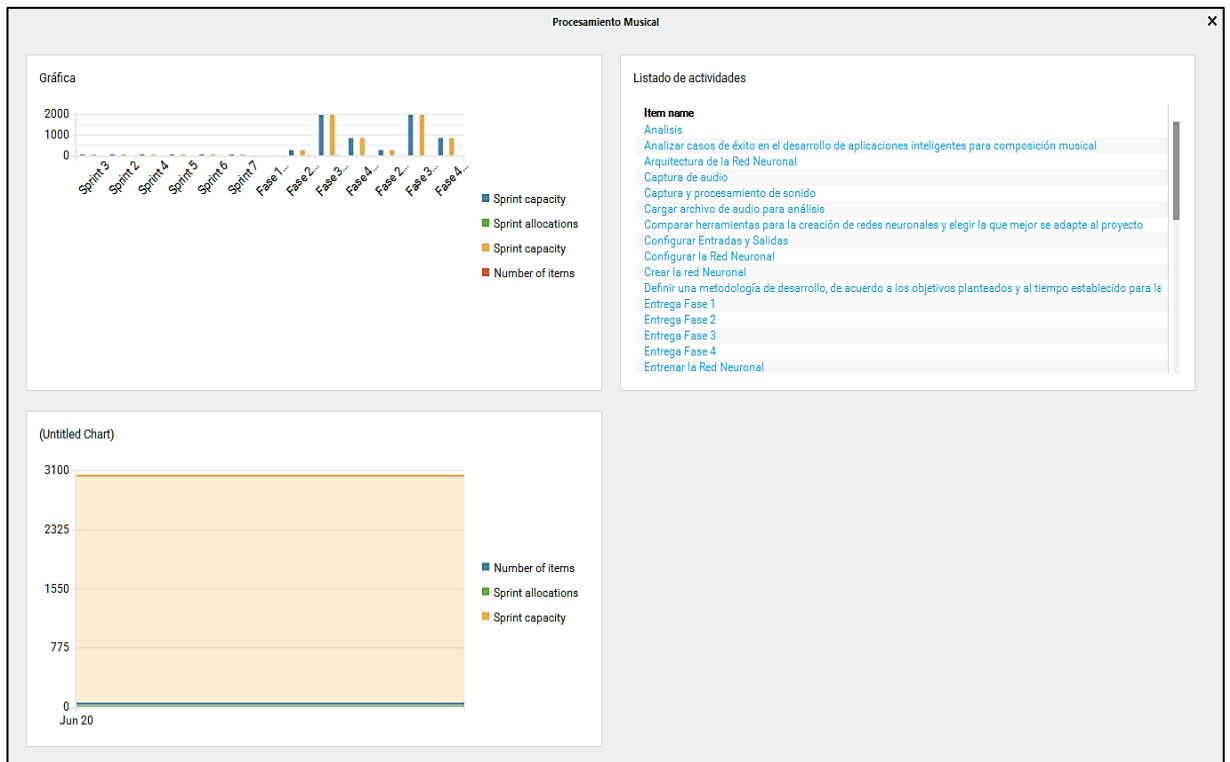


Figura 105: Visualización de diagramas del proyecto.

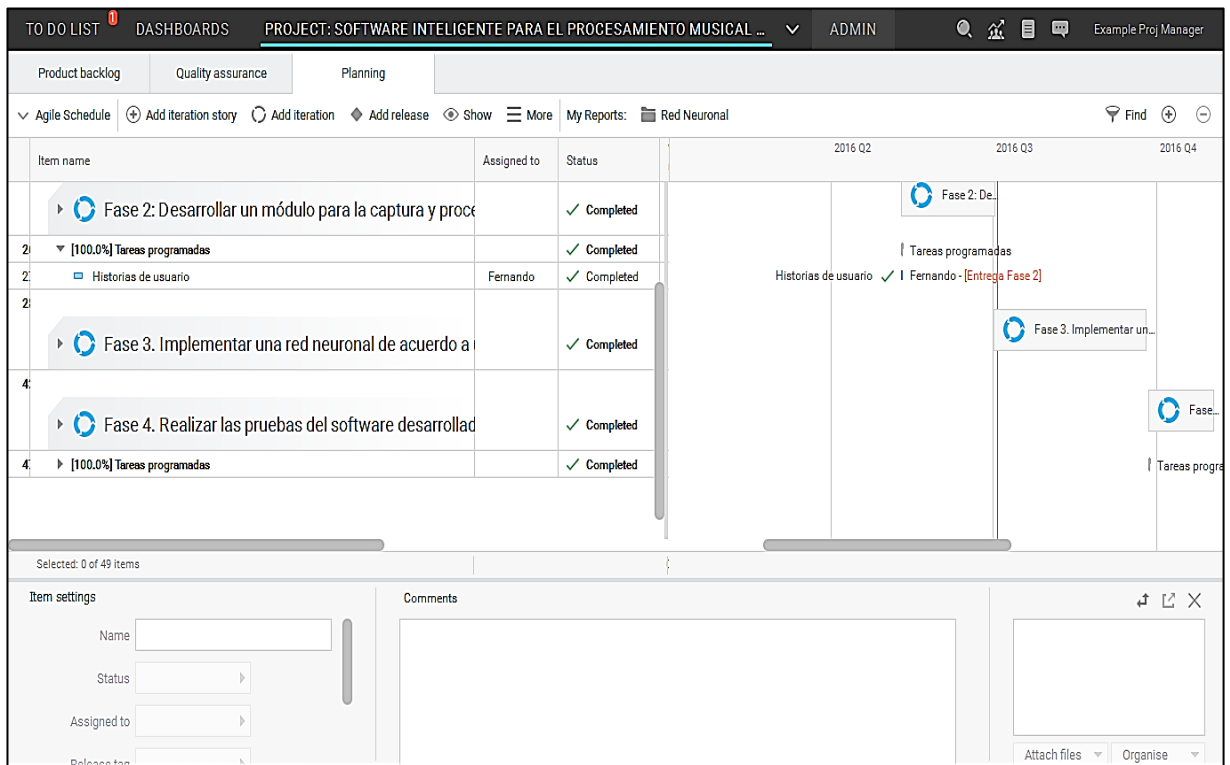


Figura 106: Creación de tareas programadas del proyecto.

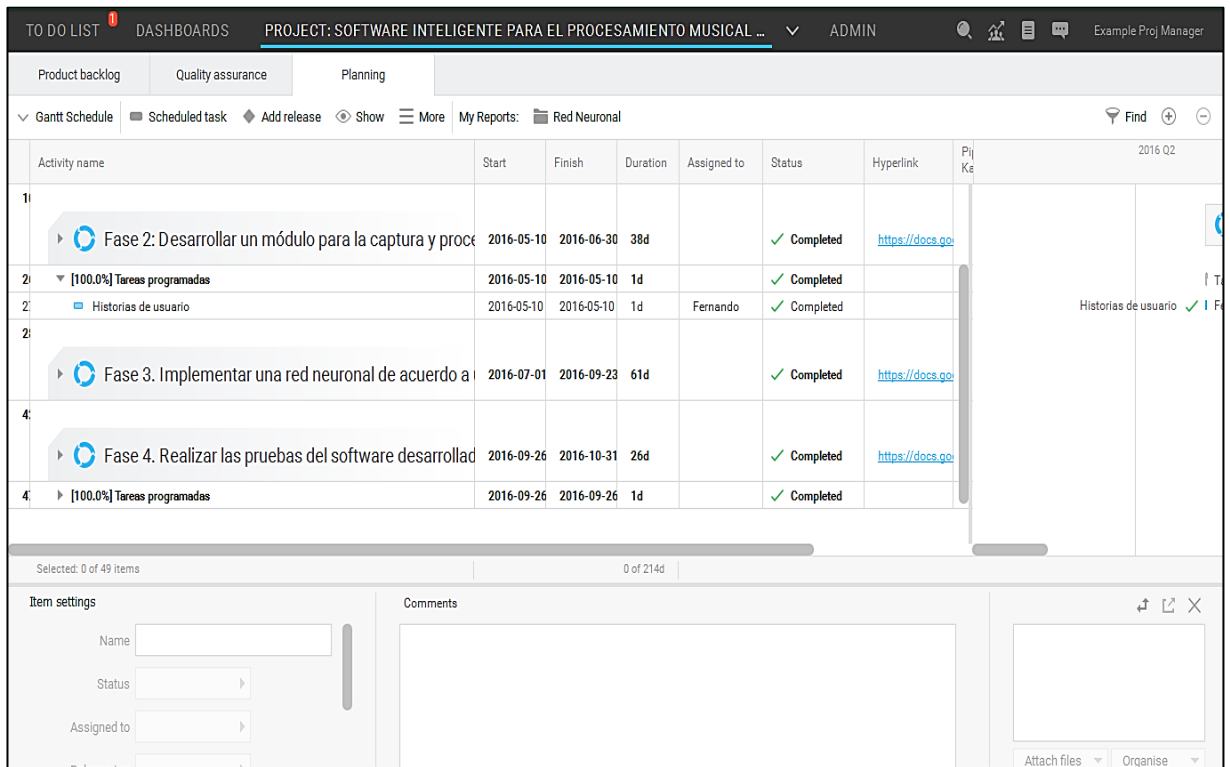


Figura 107: Creación de diagramas del proyecto.

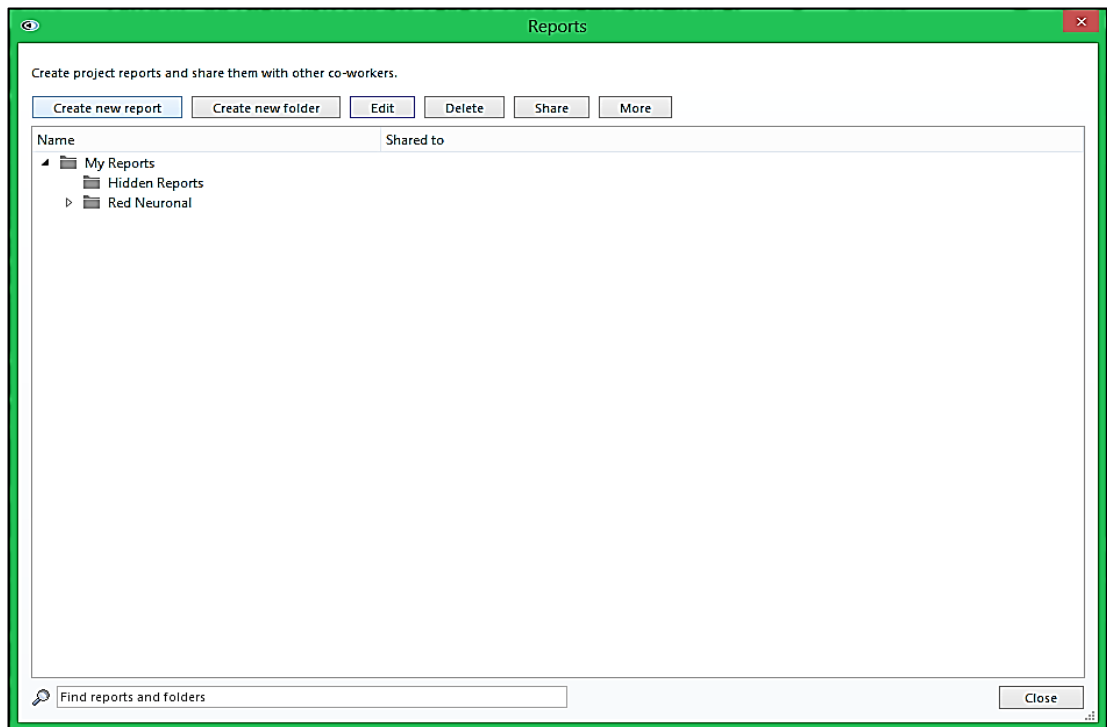


Figura 108. Creación de reportes del proyecto.

TO DO LIST DASHBOARDS PROJECT: SOFTWARE INTELIGENTE PARA EL PROCESAMIENTO MUSICAL ... ADMIN Example Proj Manager

Product backlog Quality assurance Planning

▼ Fase 1 + Add iteration story ↻ Add iteration ♦ Add release ⦿ Show ≡ More My Reports: Red Neuronal: Fase 1 Find

Item name	Assigned to	Status ↑	Work remaining	Iteration priority	Release tag	Hyp	2016 Q2
0 ♦ Releases: Entrega Fase 1		✓ Completed					
1 ♦ [100.0%] Releases: Entrega Fase 2		✓ Completed					
2 ♦ Releases: Entrega Fase 3		✓ Completed					
3 ♦ [100.0%] Releases: Entrega Fase 4		✓ Completed					
4							
Fase 1: Analizar las tecnologías existentes para el de		✓ Completed			Entrega Fase 1 :		
5 Fase 1: Analizar las tecnologías existentes para el desarrollo de la aplicación de tal man...		✓ Completed			Entrega Fase 1 :		
6 Analisis : Analizar casos de éxito en el desarrollo de aplicaciones inteligentes para con...	Mario	✓ Completed	0h	Very high priority	Entrega Fase 1 ht		
7 Analisis : Comparar herramientas para la creación de redes neuronales y elegir la que L...	Fernando	✓ Completed	0h	High priority	Entrega Fase 1 ht		
8 Analisis : Seleccionar las alternativas tecnológicas para la captura de sonido en tiemp...	Fernando	✓ Completed	0h	High priority	Entrega Fase 1		
9							
Fase 2: Desarrollar un módulo para la captura v. proce		✓ Completed			Entrega Fase 2: ht		

Selected: 1 of 49 items 0 of 0h

Sub project

Name Releases

Status Not done

Assigned to More...

Release tag

Comments

Add a comment...

Attach files Organise

Figura 109: Visualización del reporte de la Fase 1 del proyecto.

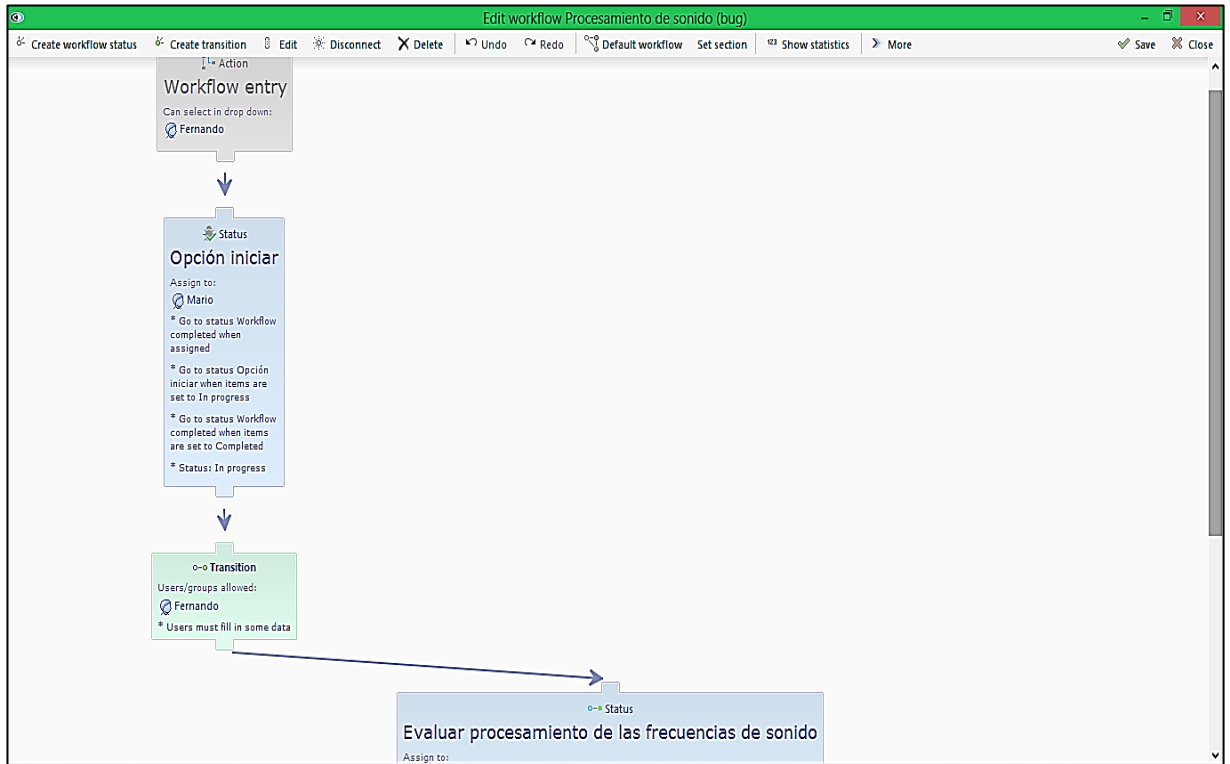


Figura 110: Creación de Workflow del proyecto.

TO DO LIST DASHBOARDS PROJECT: SOFTWARE INTELIGENTE PARA EL PROCESAMIENTO MUSICAL ... ADMIN Example Proj Manager

Product backlog Quality assurance Planning

Report new bug Show More My Reports:

Reports a new bug

Find by keywords Browse through statuses and more...

ID	Description	Assigned to	Status	Committed to iteration	Severity ↓
5	Presentación de resultados ■	Mario	✓ Completed	Fase 4. Realizar l	Severity C
1	Entrenamiento de la red Neuronal ■	Mario	✓ Completed	Fase 3. Impleme	Severity A
2	Captura de Audio ■	Mario	✓ Completed	Fase 2. Desarroll	Severity A
4	Procesamiento de Sonido ■	Mario	✓ Completed	Fase 2. Desarroll	Severity A

Selected: 0 of 4 items

Bug Description Status Assigned to

Detailed description Steps to reproduce Comments

Attach files Organise

Figura 111: Creación y visualización bugs del proyecto.

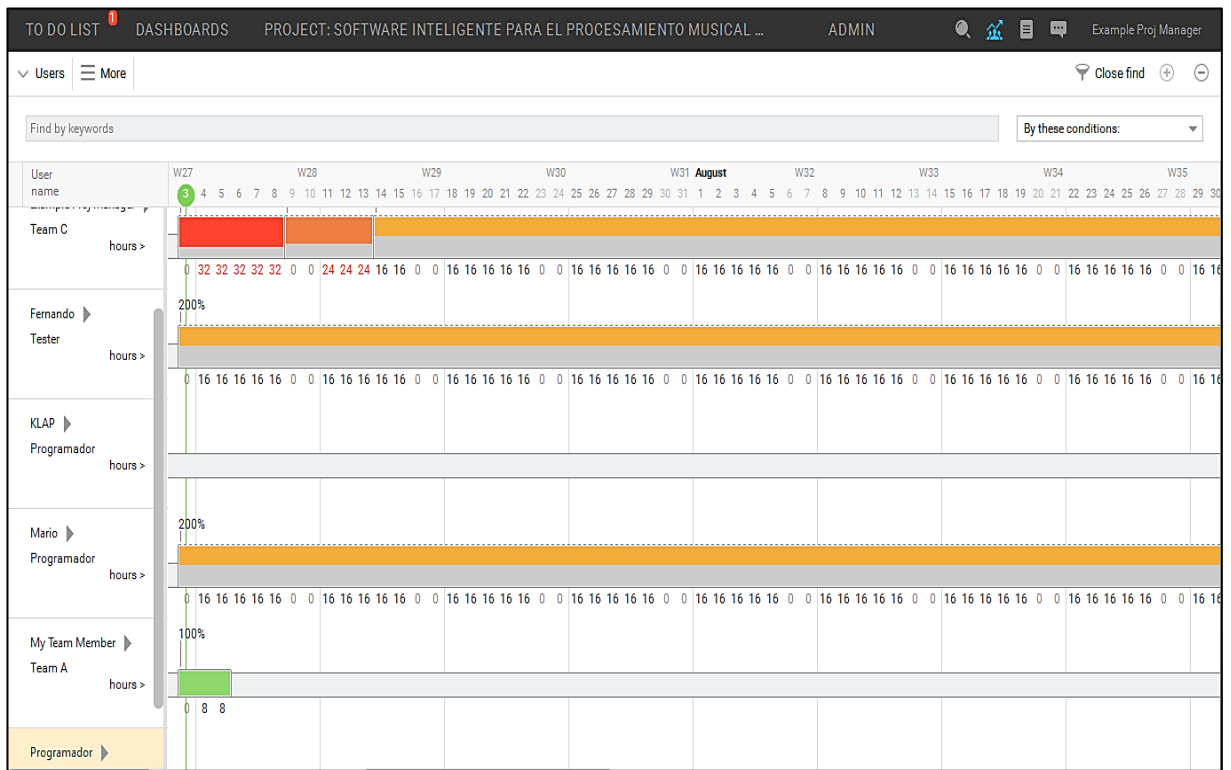


Figura 112: Visualización de las asignaciones de la cartera del proyecto.

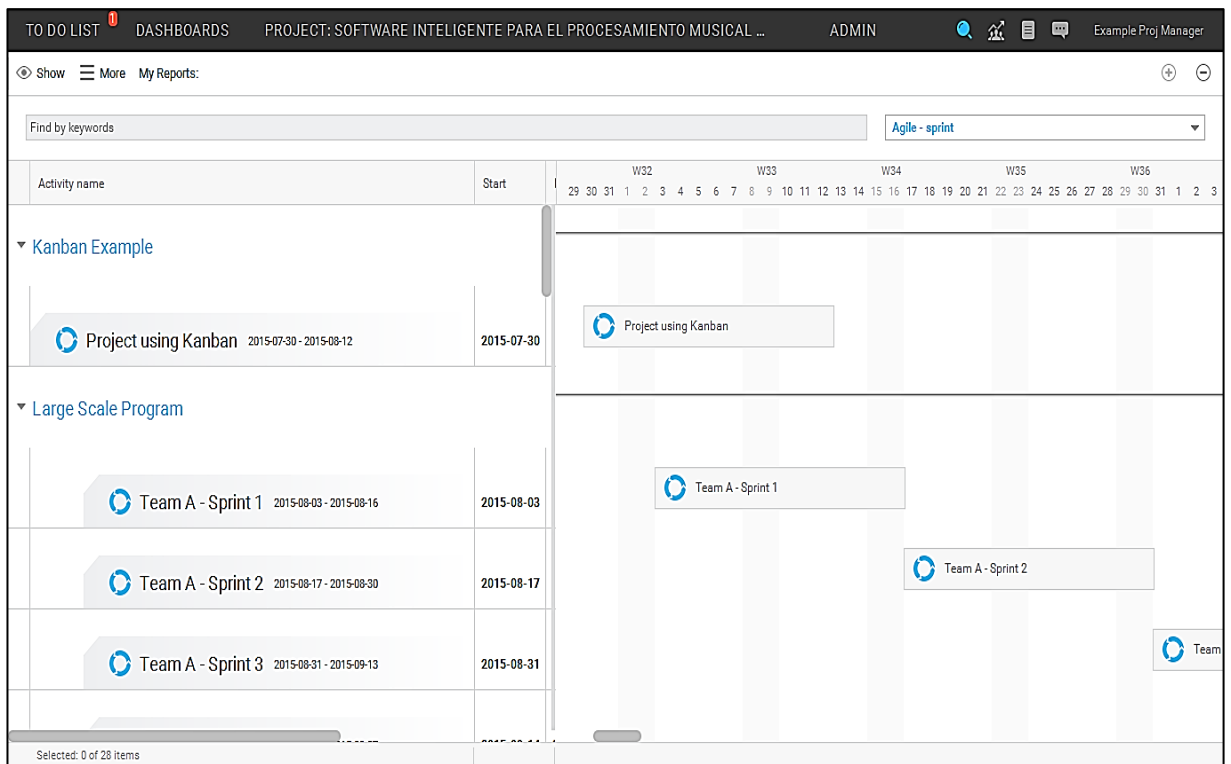


Figura 113: Visualización de todos los proyectos.

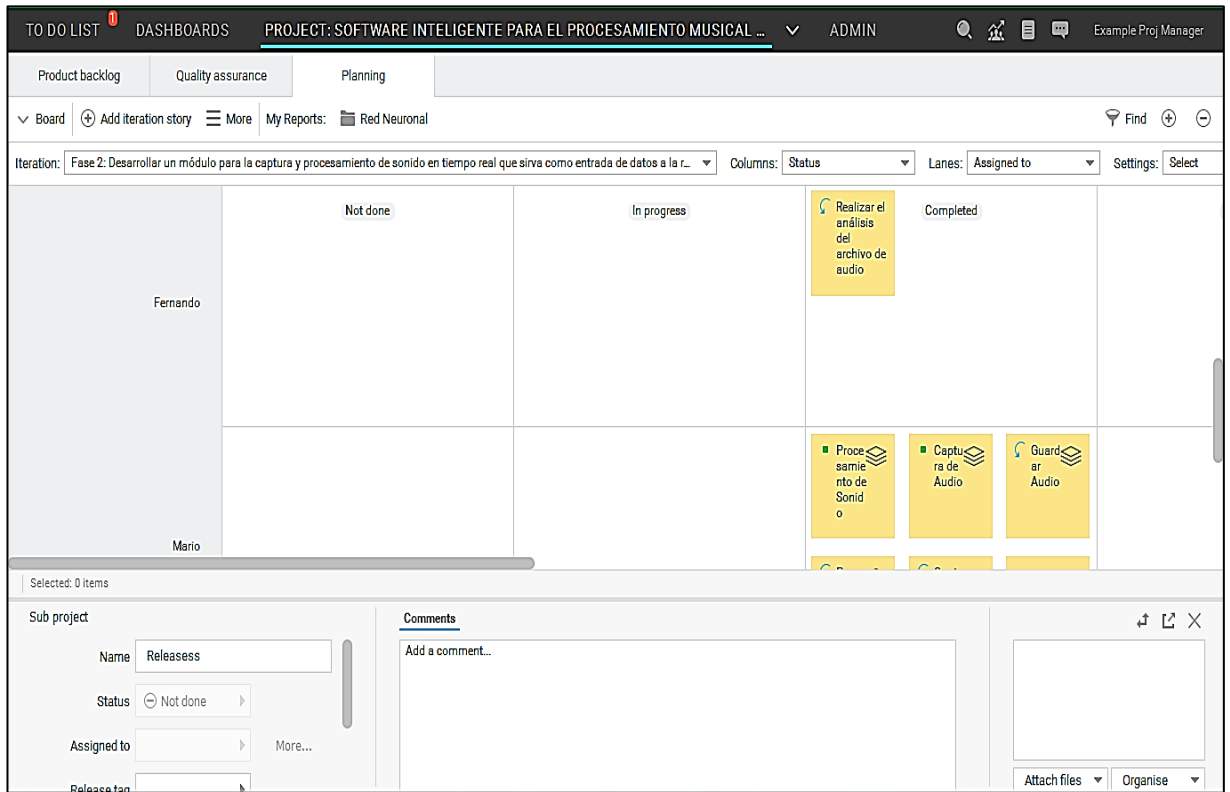


Figura 114: Visualización del tablero de actividades del proyecto.

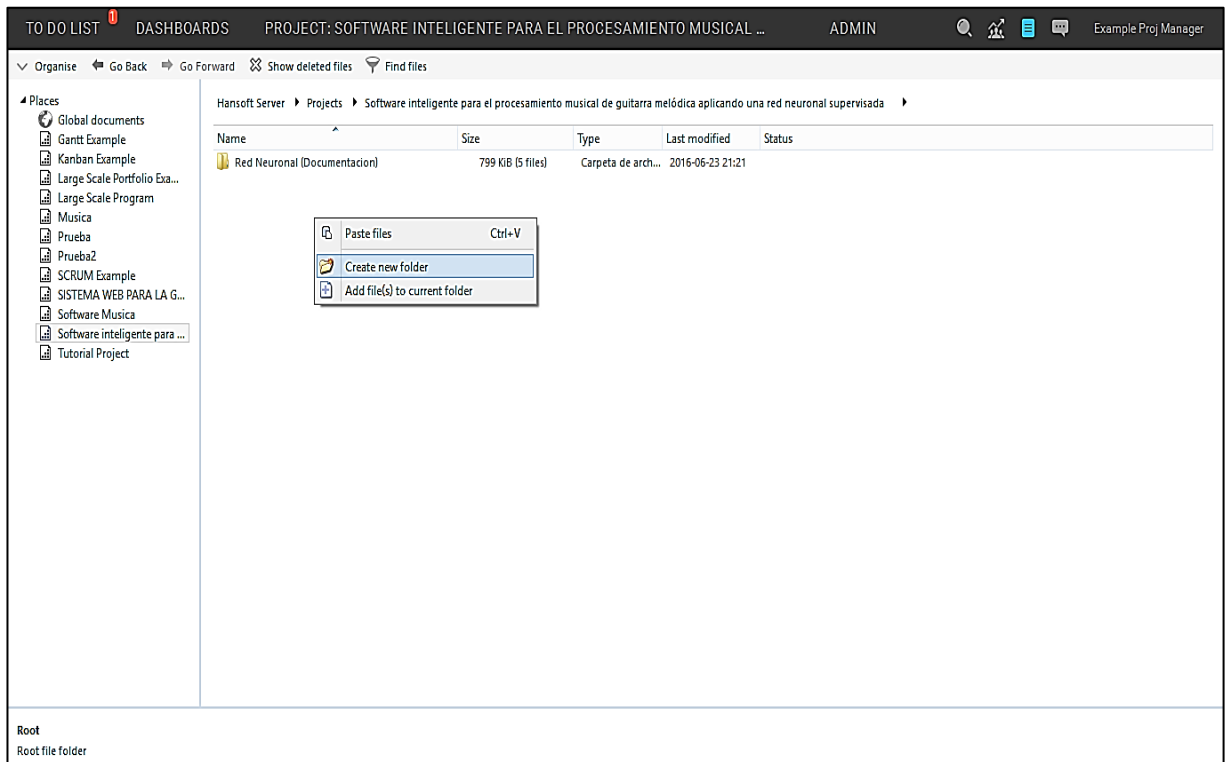


Figura 115: Creación de documentos del proyecto.

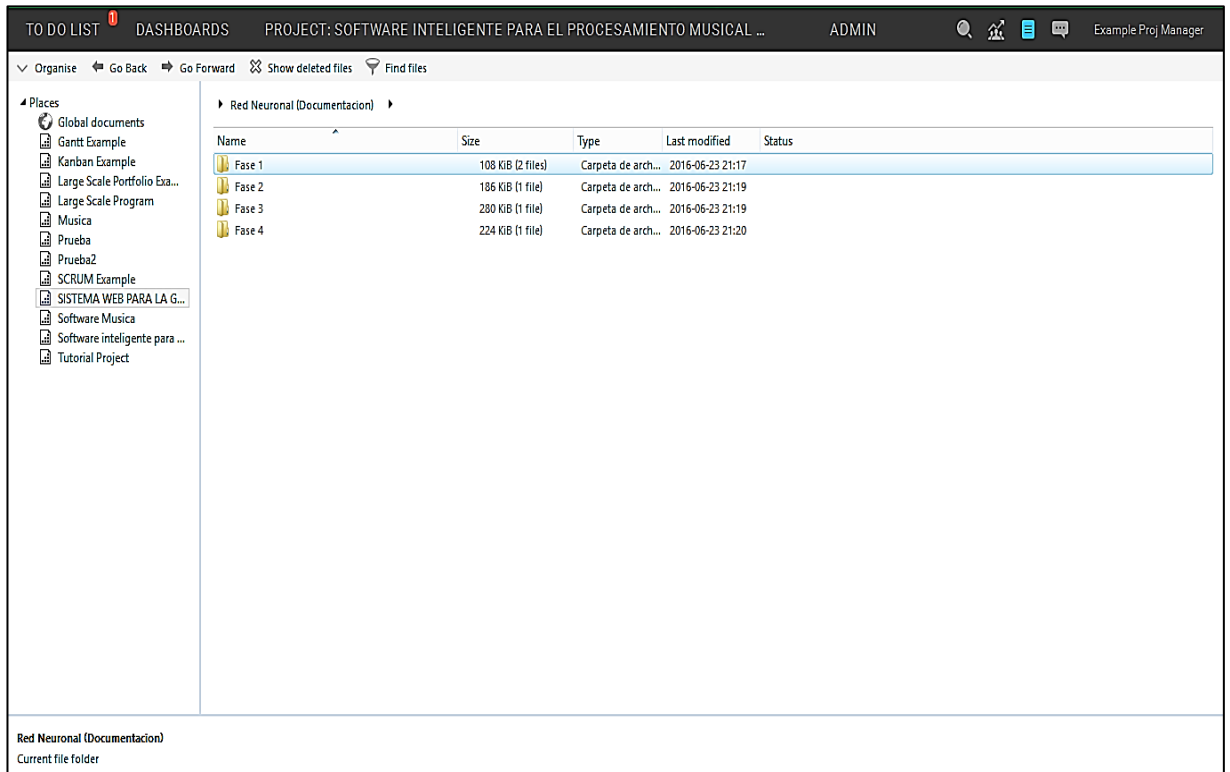


Figura 116: Visualización de documentos del proyecto.

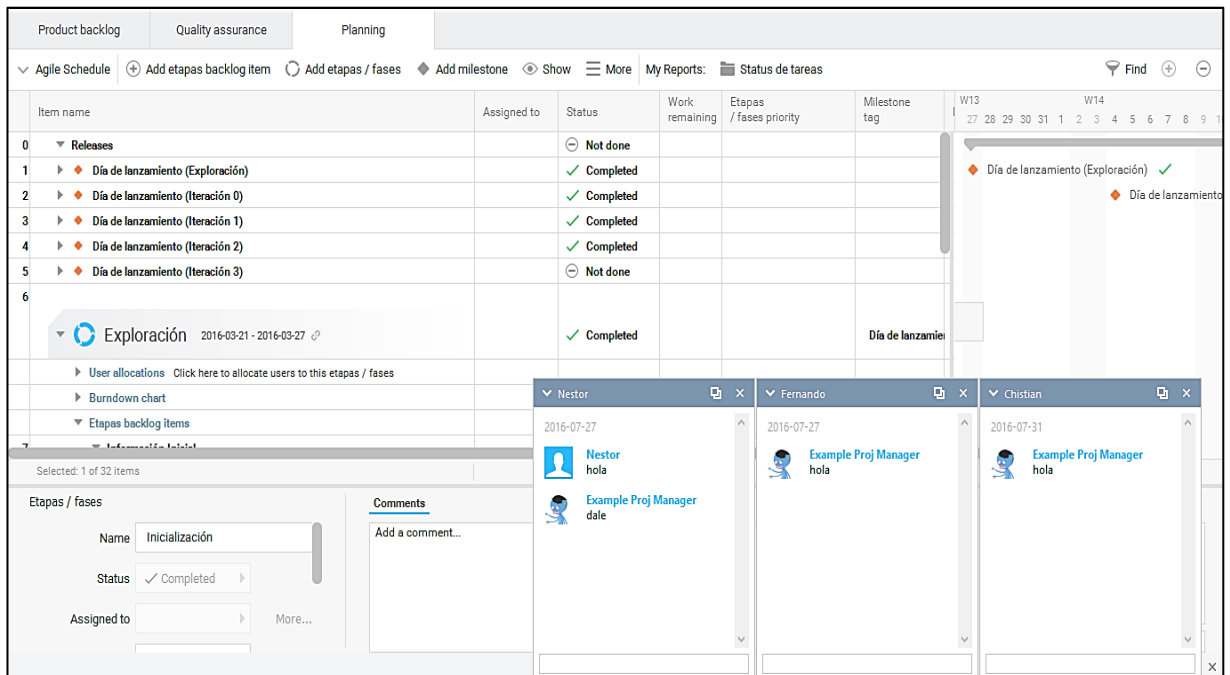


Figura 117: Comunicación / Chat.



Figura 118: Soporte de Hansoft en su página oficial.

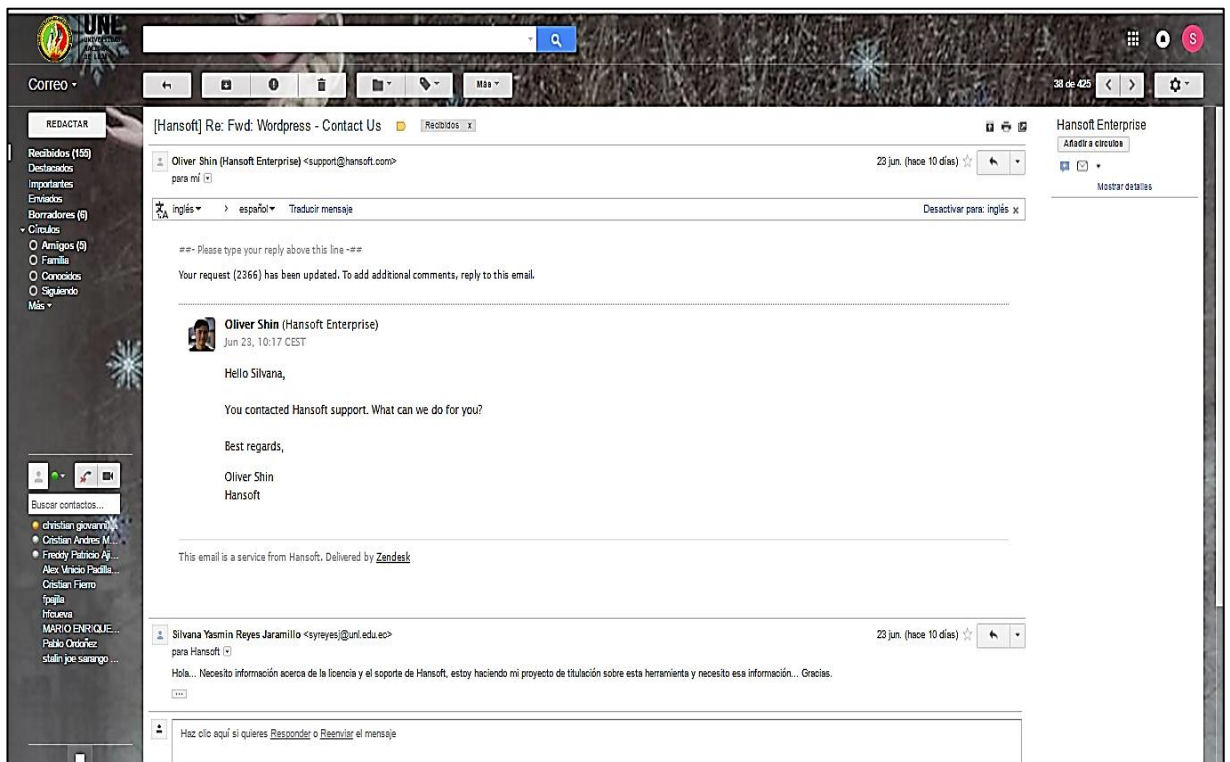


Figura 119: Soporte a través del envío de email.

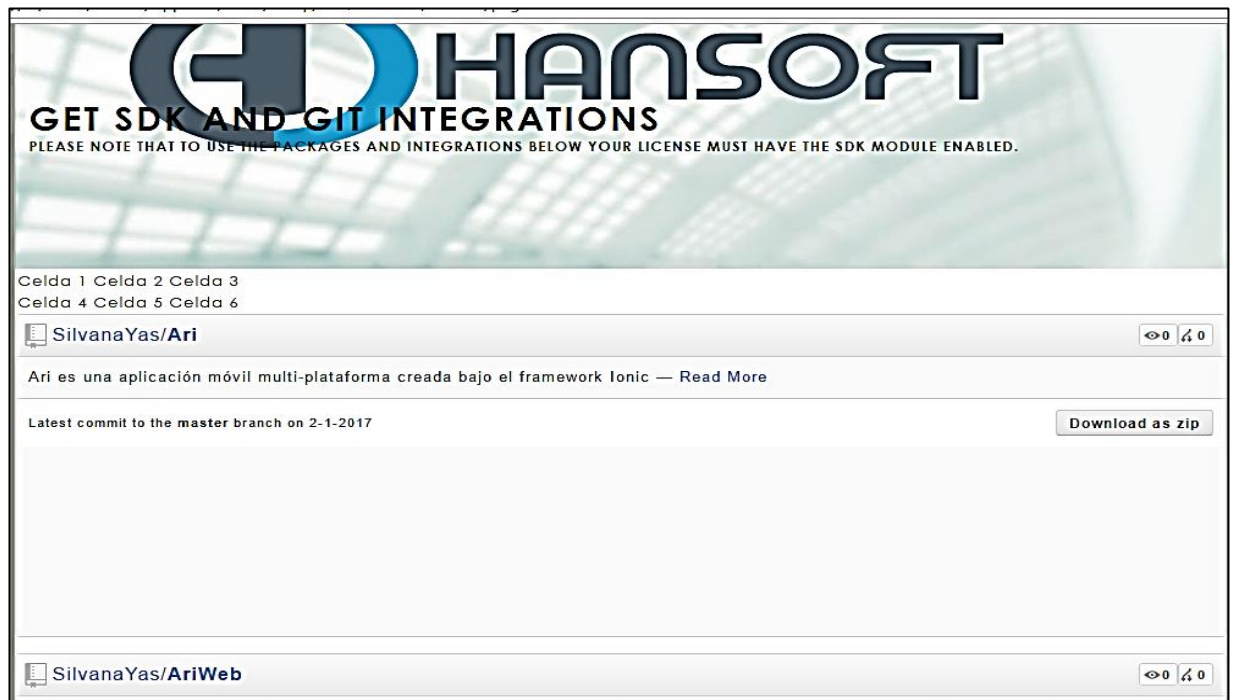


Figura 120: Versionamiento a través Git.

Tuleap

En este caso, Tuleap permite crear proyectos seleccionando una plantilla ágil o una por defecto. En esta ocasión en particular se seleccionó la ágil, el proyecto se está desarrollando con la metodología XP, permite asignar responsables, sprints, release, tareas, asignar fechas, reportes estadísticos, discusiones, bugs, añadir artefactos y crear gráficas.

Patrocinado por enalean

Mi página personal proyectos Ayuda Más Search Silvana Reyes

Describe su proyecto

Nombre Completo *
entre 3 y 40 caracteres

Nombre corto *
entre la longitud de 3 y 30 caracteres, con sólo caracteres, números y guiones. Debe comenzar con una letra. NO se permiten espacios.
info El nombre corto es un nombre técnico utilizado en diversas direcciones URL (URL del proyecto y repositorios de código fuente) y otras referencias.

Descripción breve *
¿Cuál es el propósito de su proyecto?

Completo Descripción del Proyecto *
***** NOTA IMPORTANTE** Asegúrese de que usted proporciona una descripción detallada que contiene todas las palabras clave que ayudarán a otros a encontrar y reutilizar su componente de software. No proporcionar una descripción precisa también puede retrasar la aprobación de su proyecto.

proyecto de privacidad Público
 Privado (limitar el acceso sólo a los miembros del proyecto)

Figura 121: Creación de un nuevo proyecto.

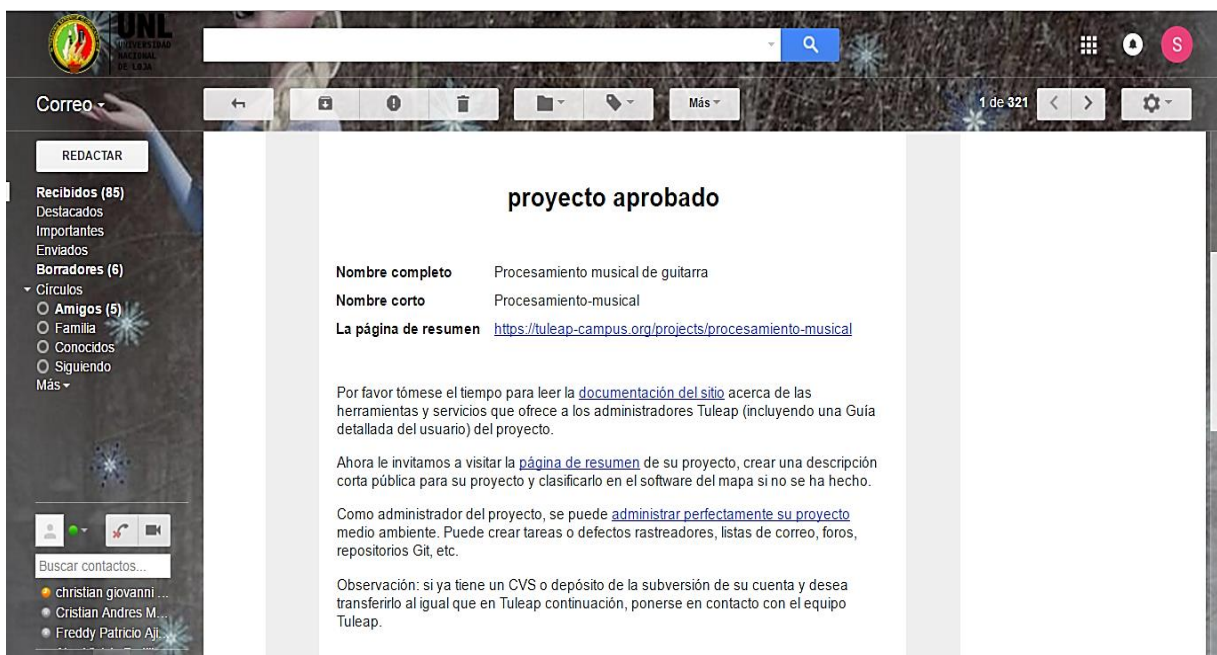


Figura 122: Correo de aprobación del nuevo proyecto.

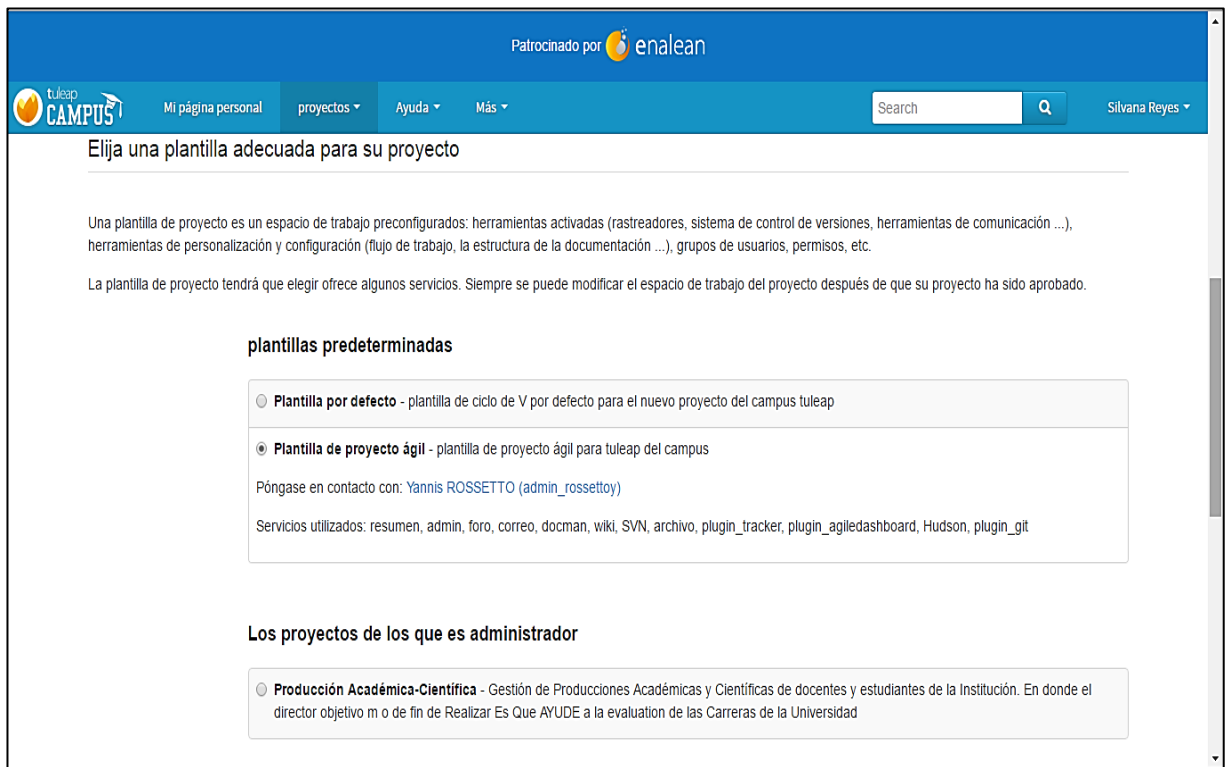


Figura 123: Selección de la plantilla para el proyecto.



Figura 124: Creación de sprints del proyecto.

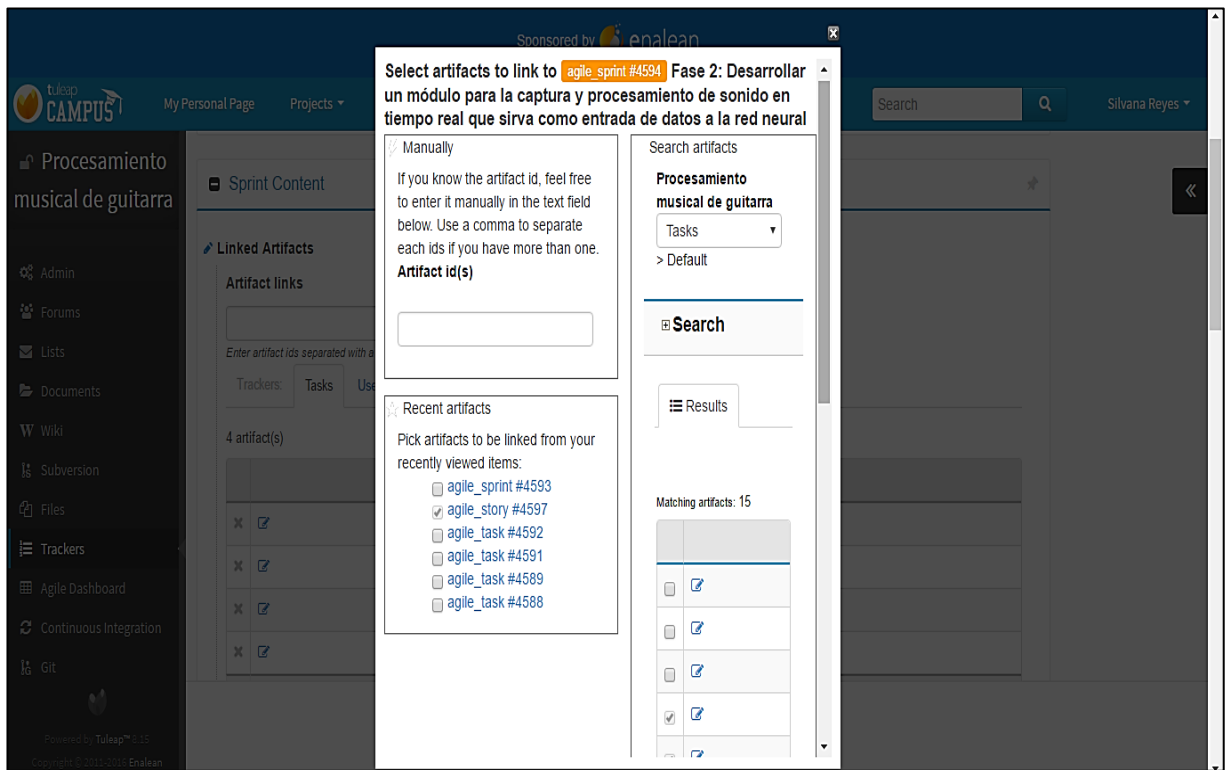


Figura 125: Adición de tareas al sprint del proyecto.

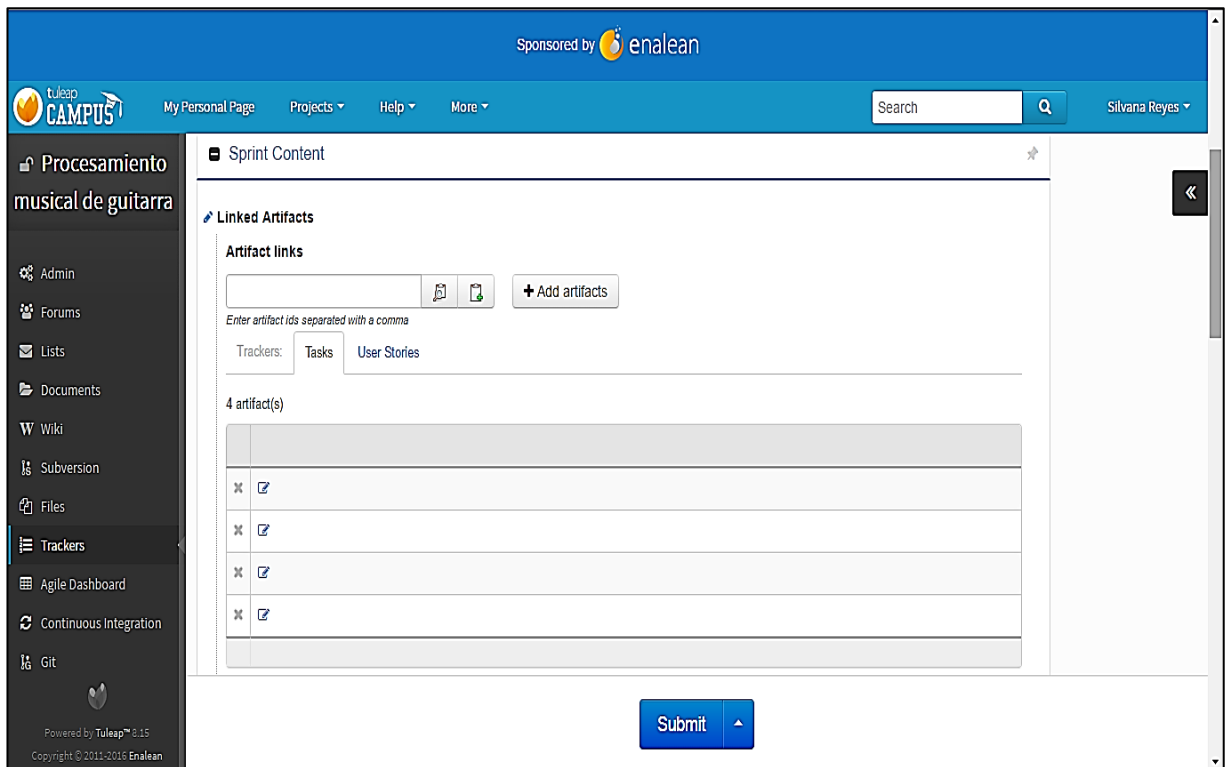


Figura 126: Visualización de los artefactos añadidos al sprint del proyecto.

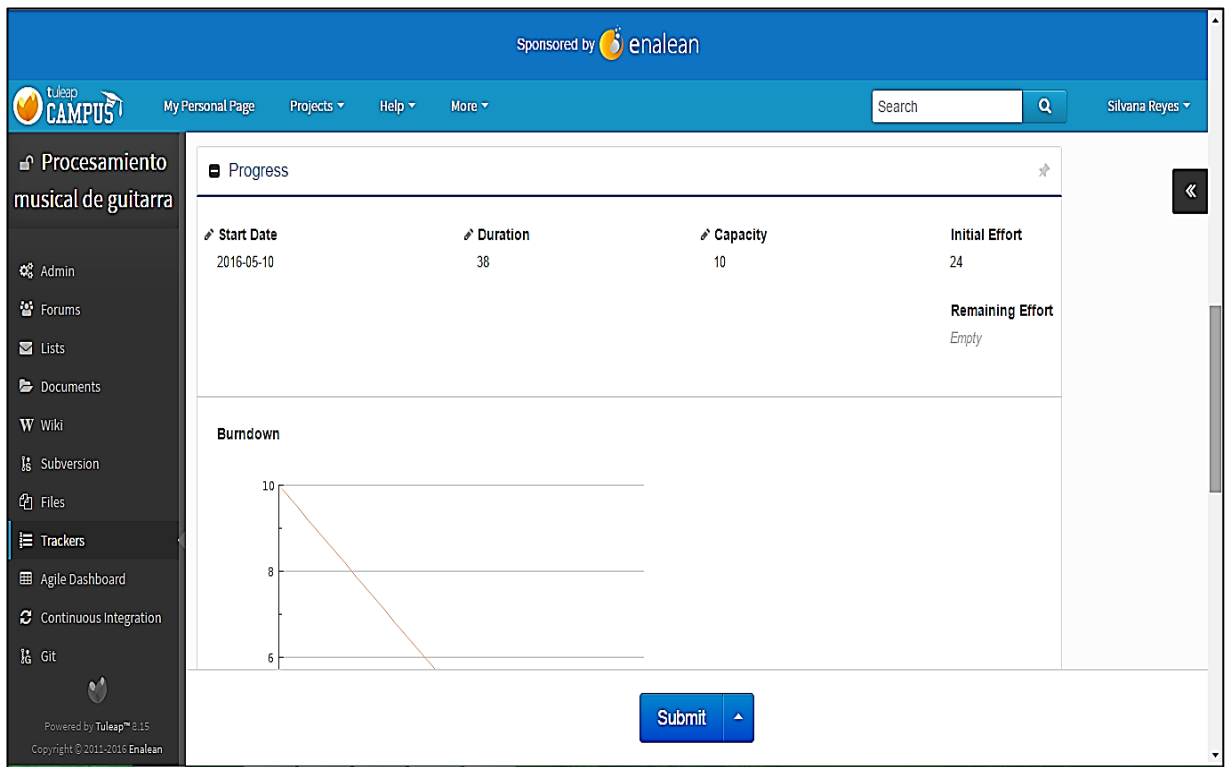


Figura 127: Ingreso de la información restante del sprint del proyecto.

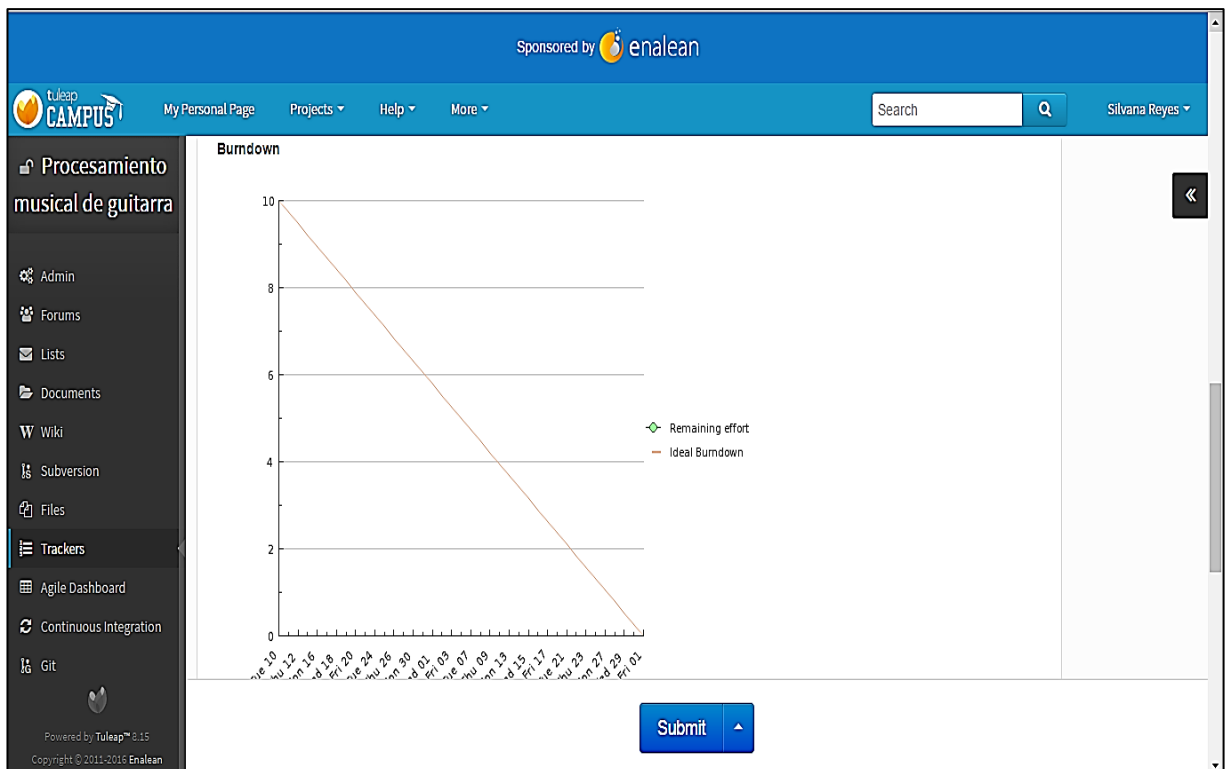


Figura 128: Visualización del avance del sprint del proyecto.

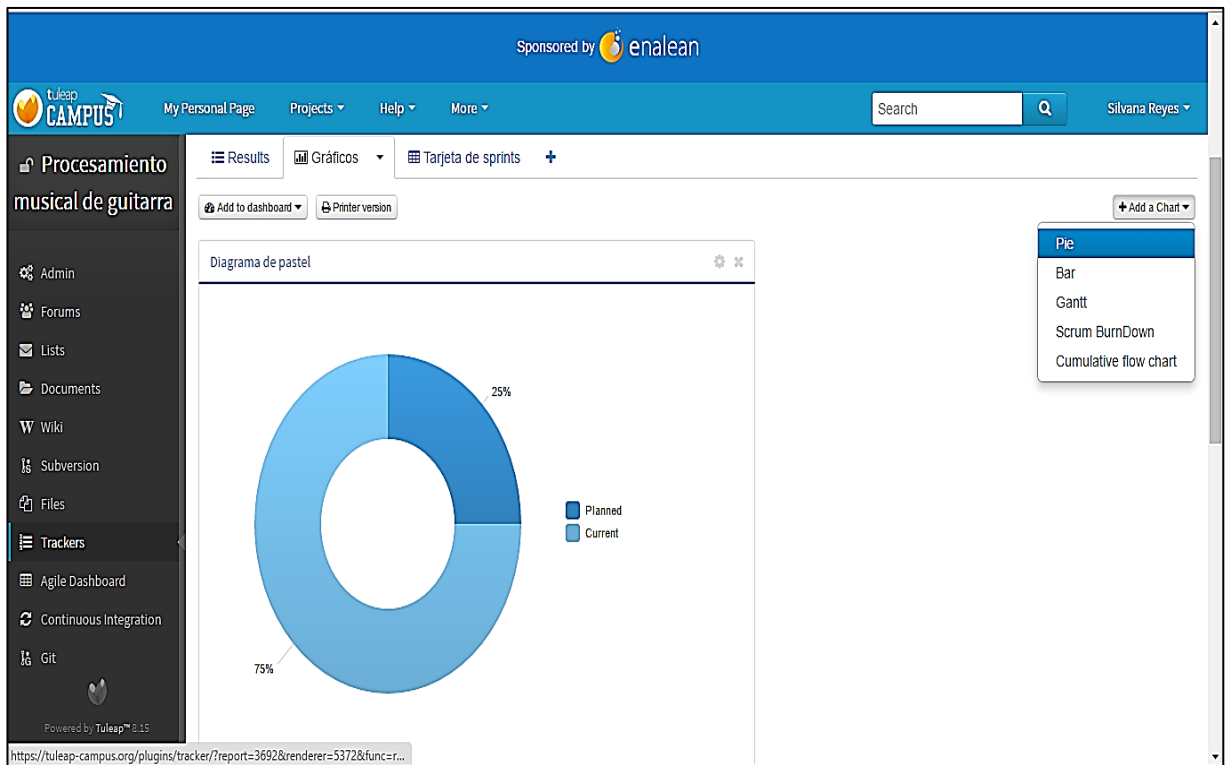


Figura 129: Creación de diagramas del proyecto.

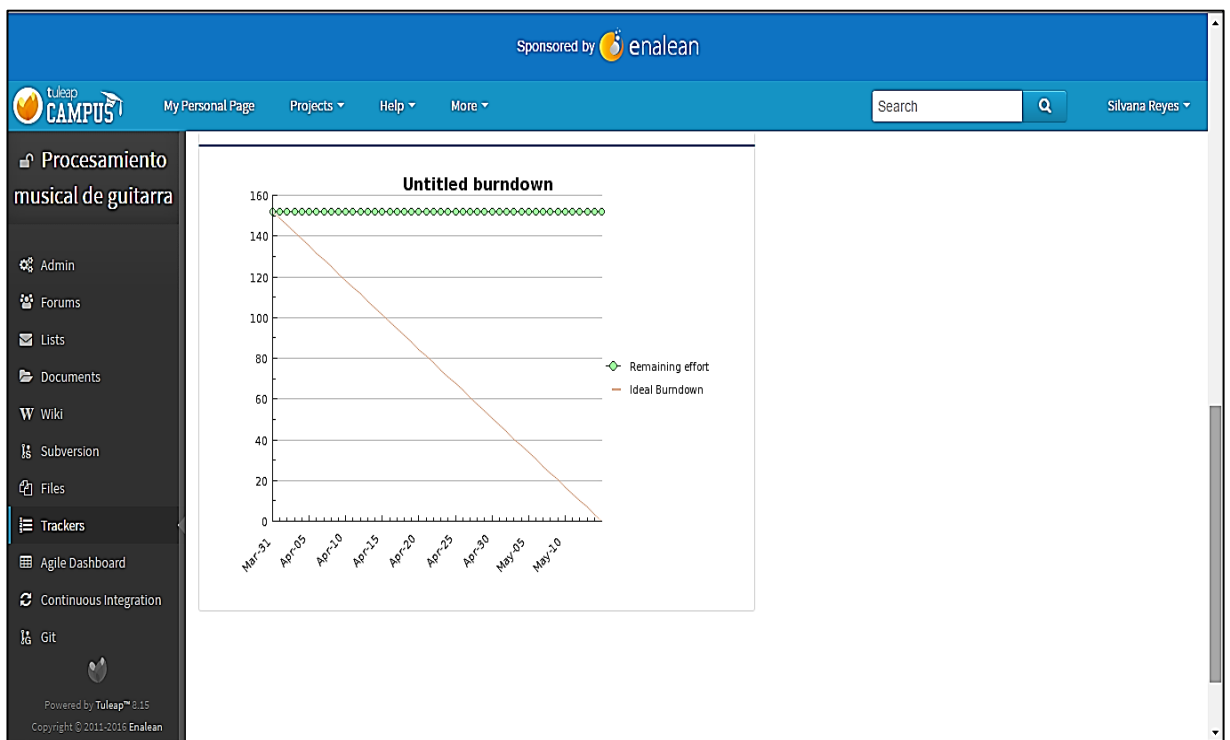


Figura 130: Creación de sprints del proyecto.

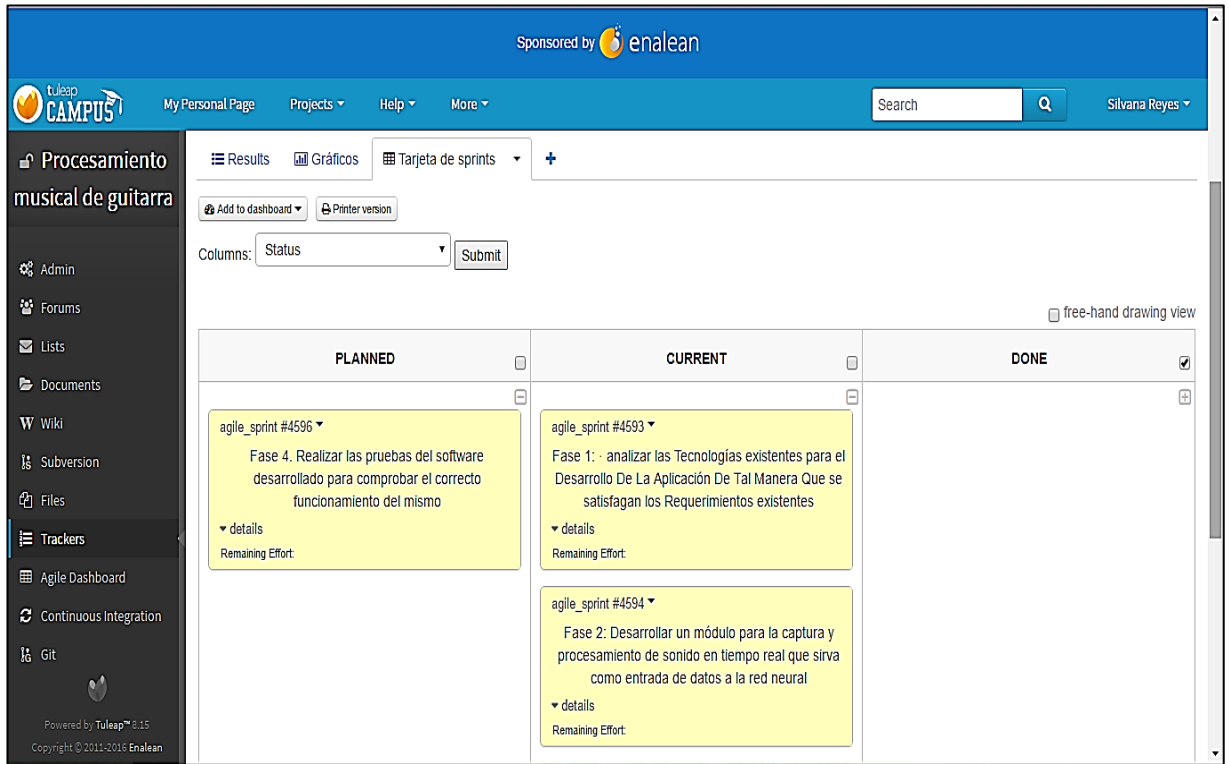


Figura 131: Creación de tarjetas en el tablero del proyecto.

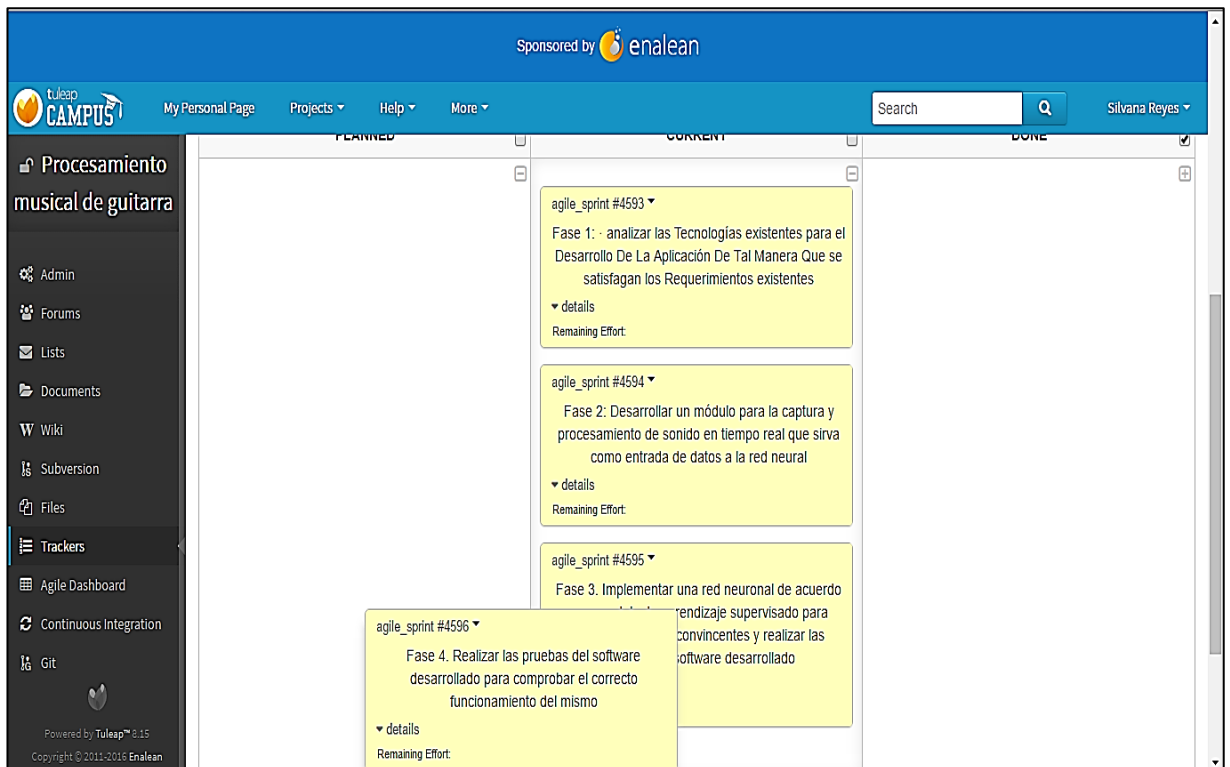


Figura 132: Visualización del cambio de estado de forma manual de las tarjetas en el tablero del proyecto.

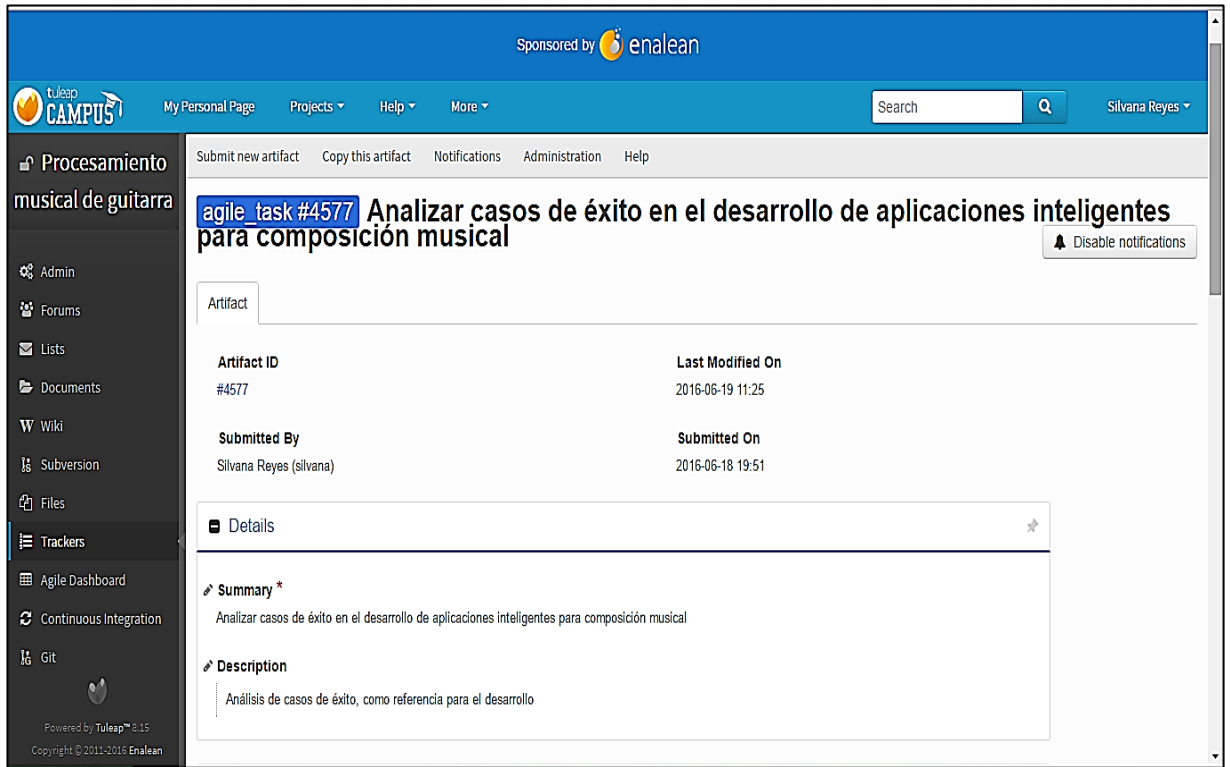


Figura 133: Creación de tareas del proyecto.

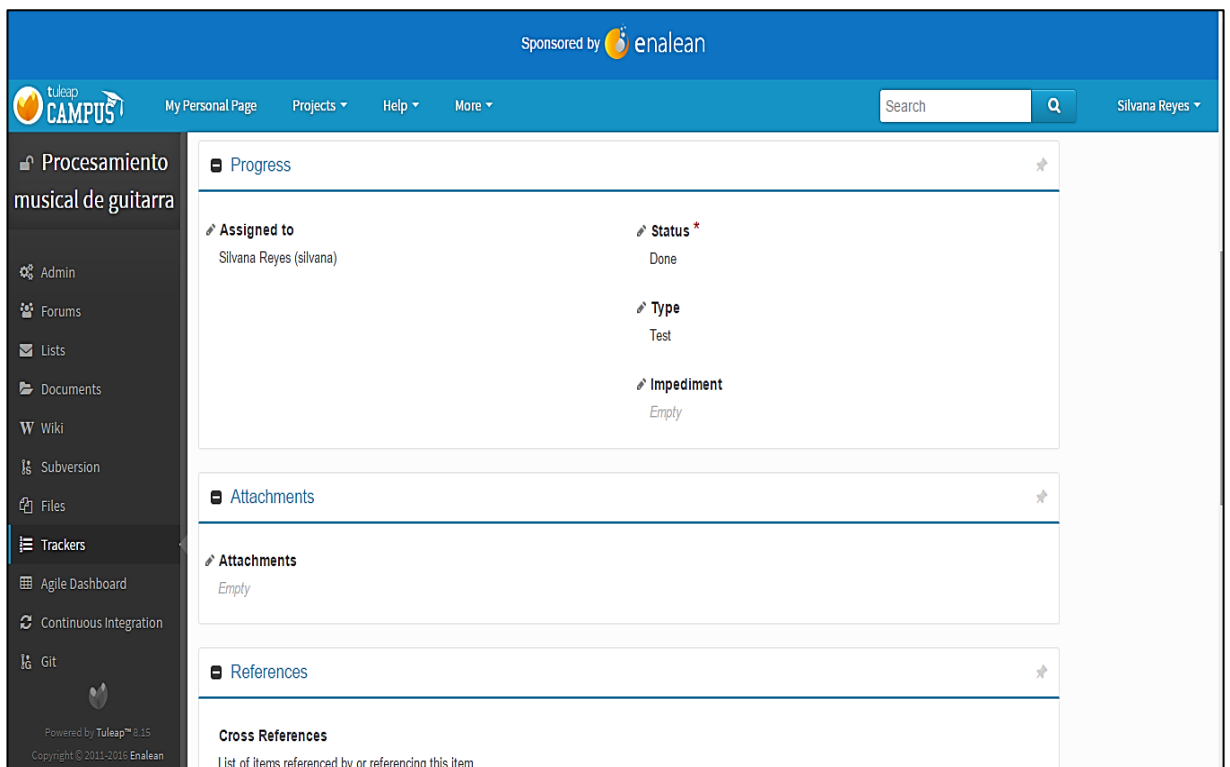


Figura 134: Asignación de datos a las tareas del proyecto.

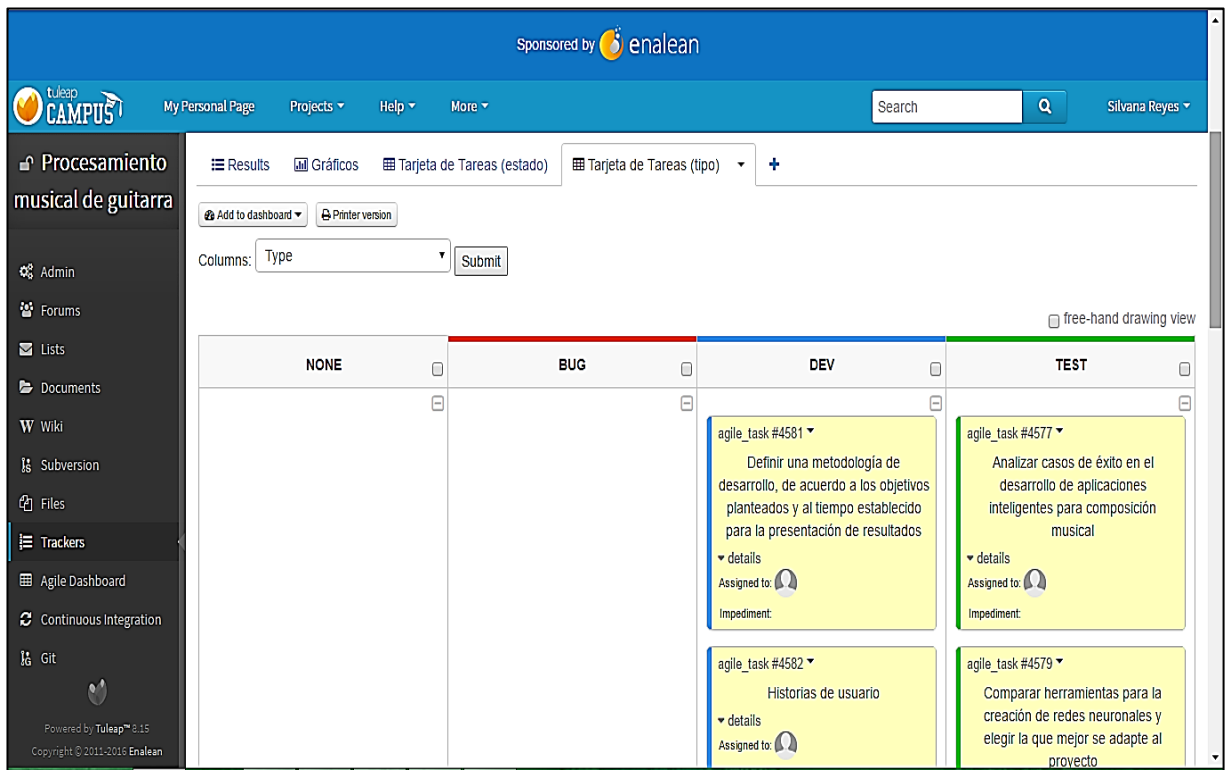


Figura 135: Creación de tarjetas en las tareas del proyecto.

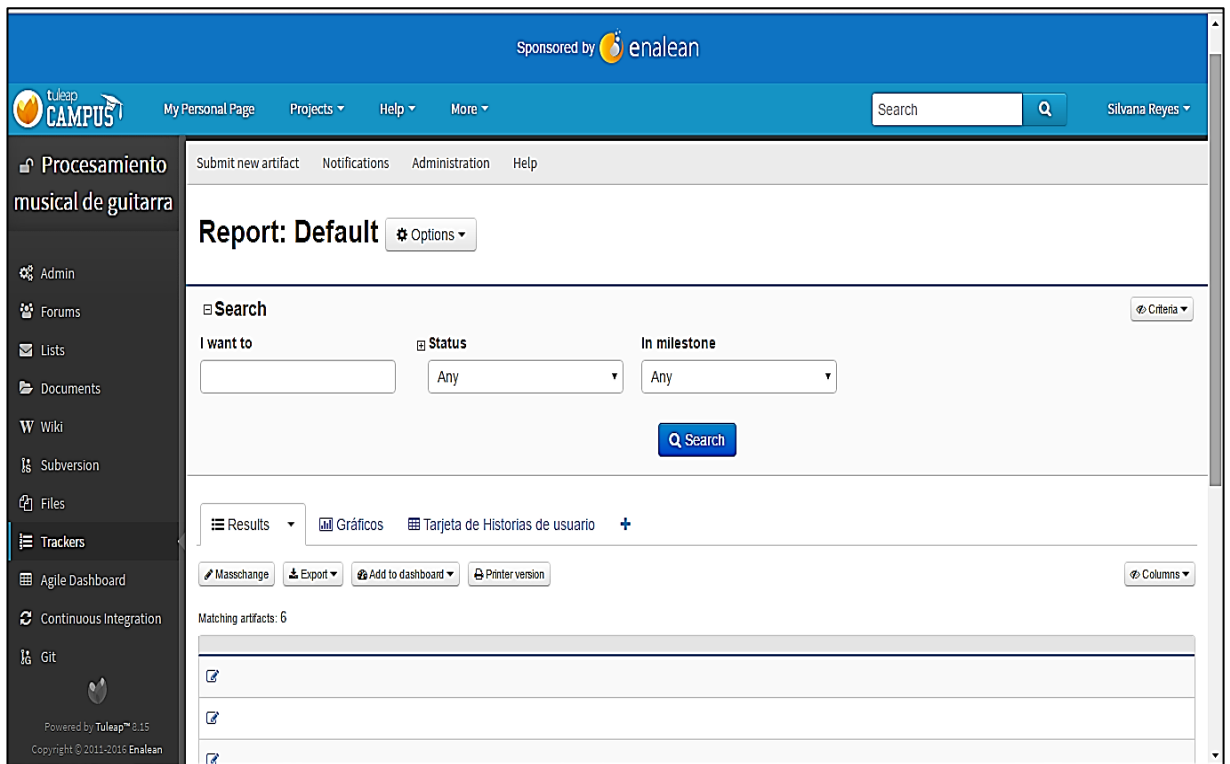


Figura 136: Visualización de los artefactos y gráficos de las tareas del proyecto.

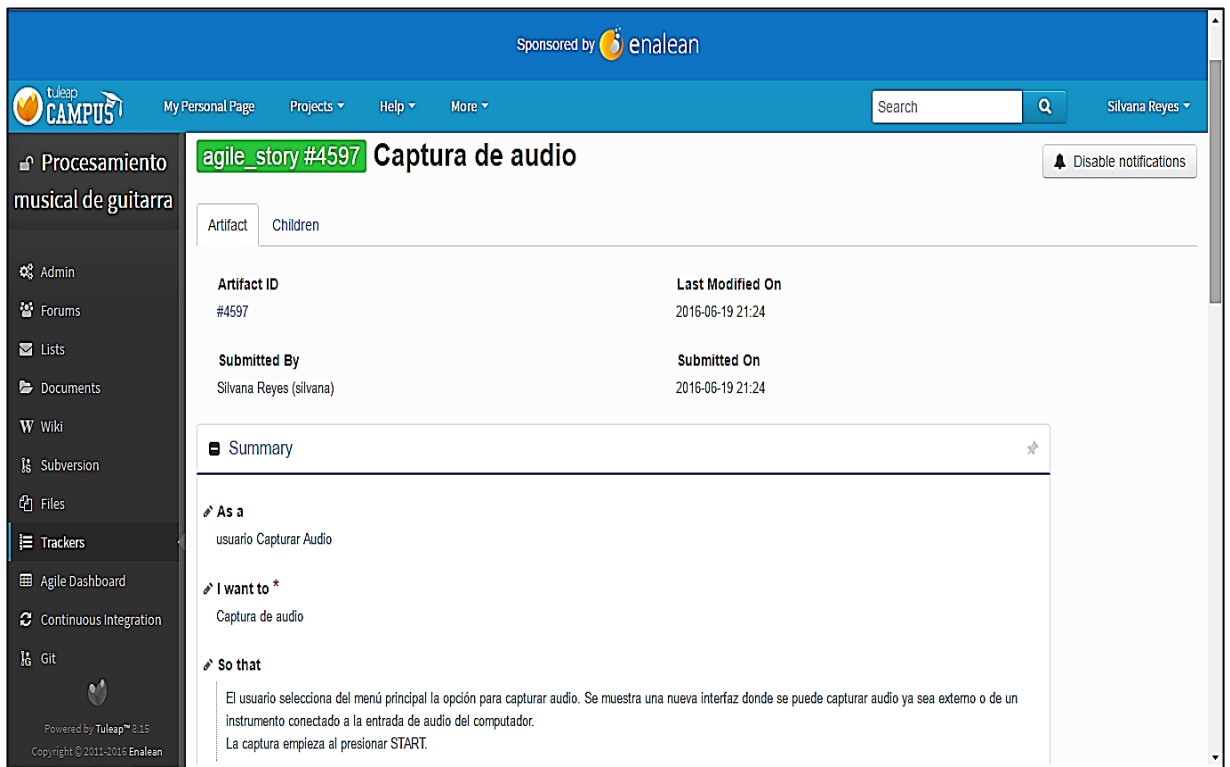


Figura 137: Creación de historias de usuario del proyecto.

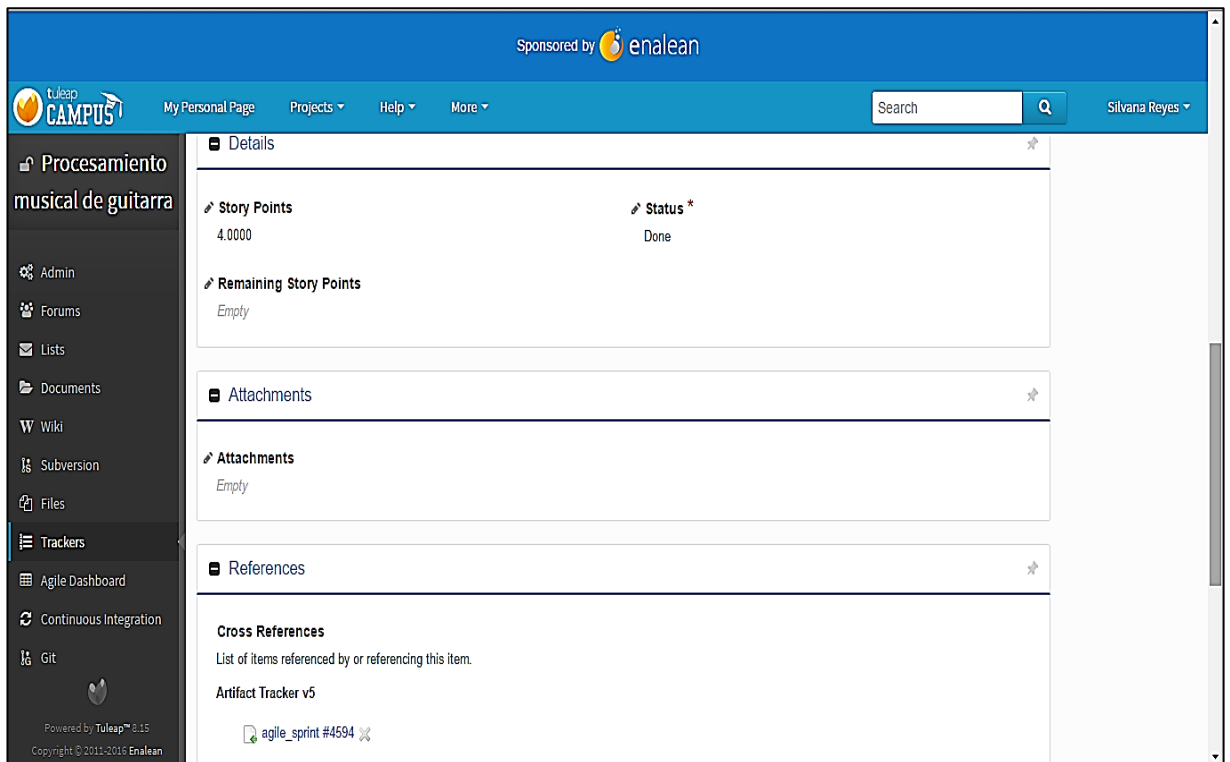


Figura 138: Referencia al sprint que pertenece la historia del proyecto.

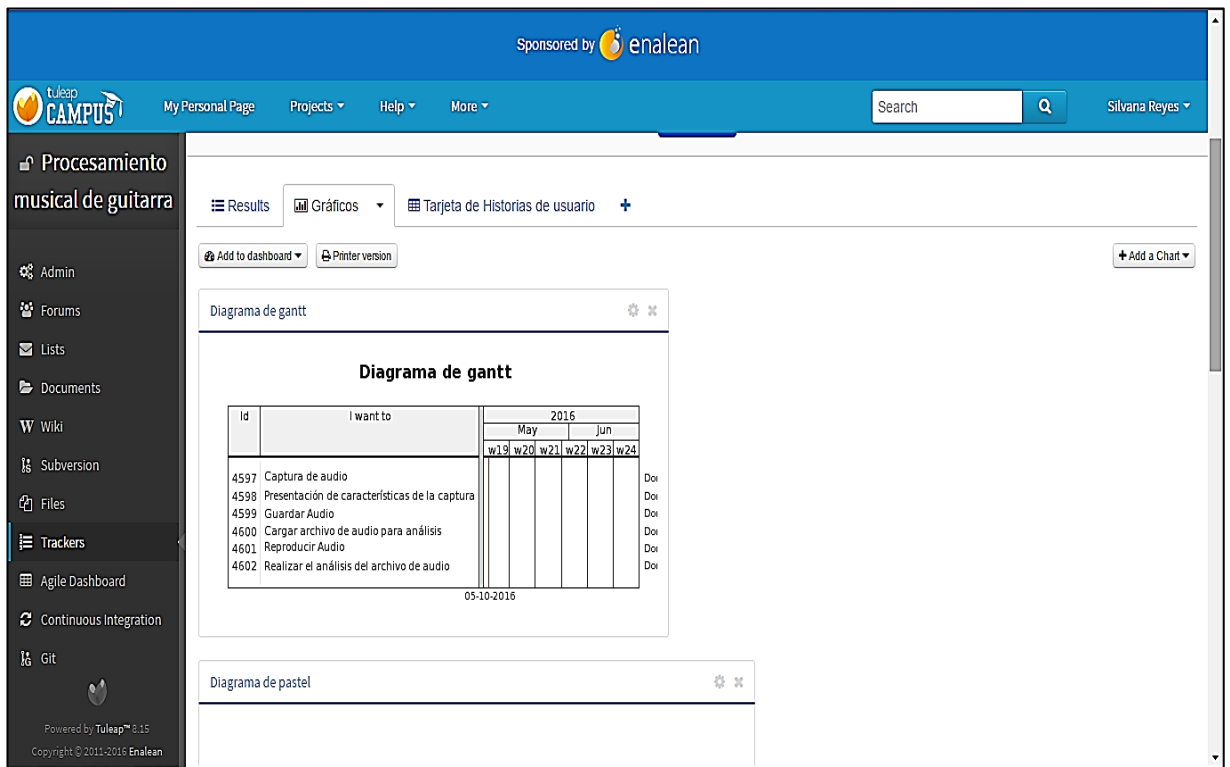


Figura 139: Visualización de los gráficos de las historias de usuario del proyecto.

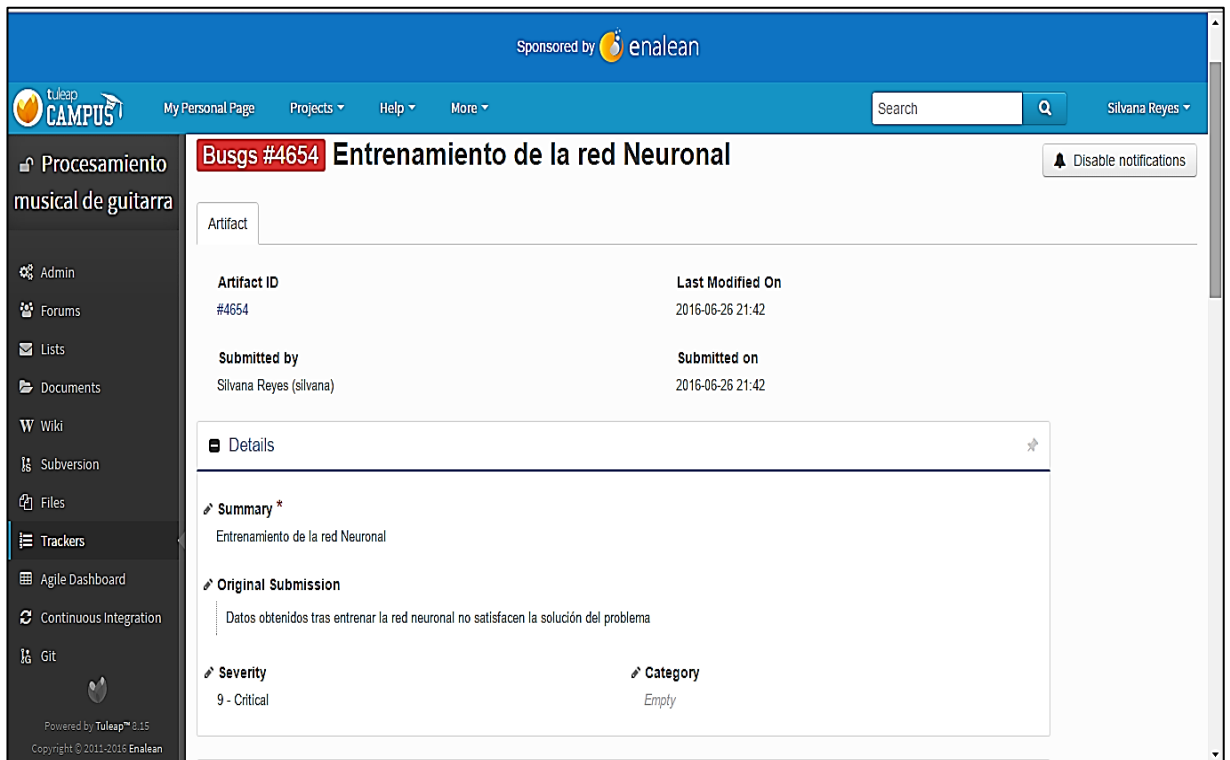


Figura 140: Creación de bugs del proyecto.

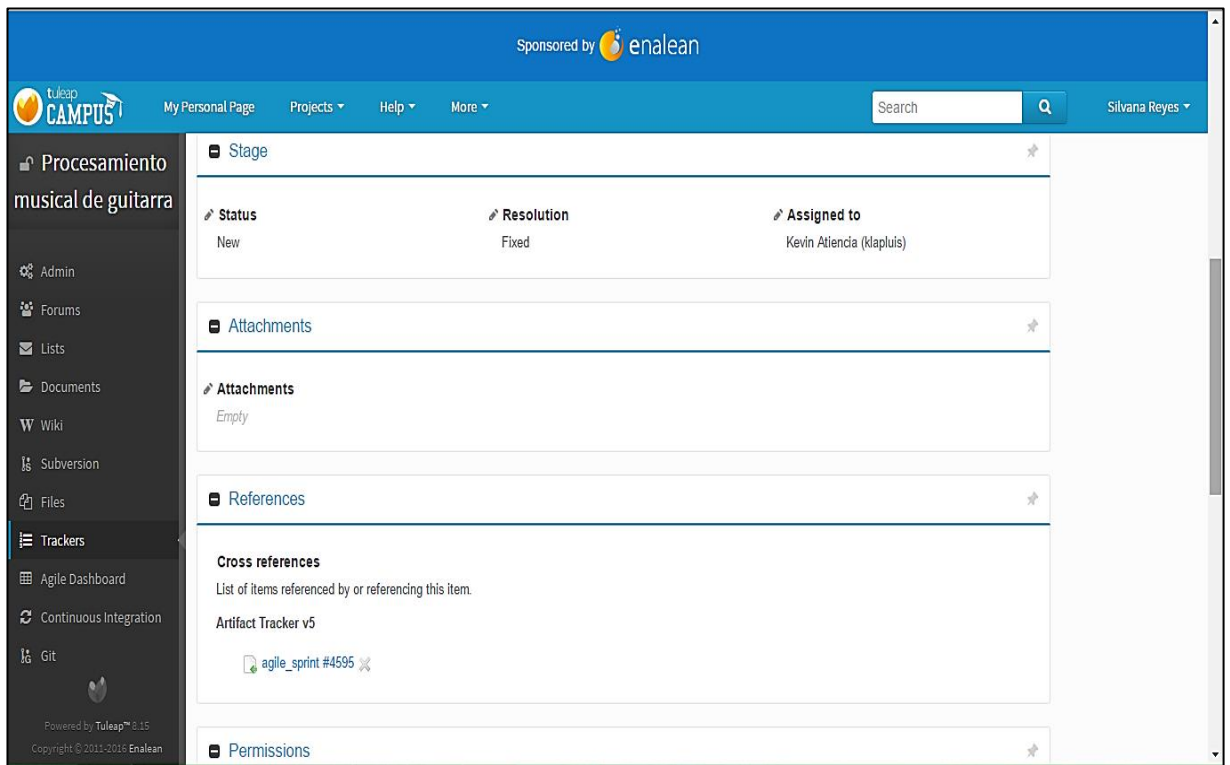


Figura 141: Ingreso de datos del bug del proyecto.

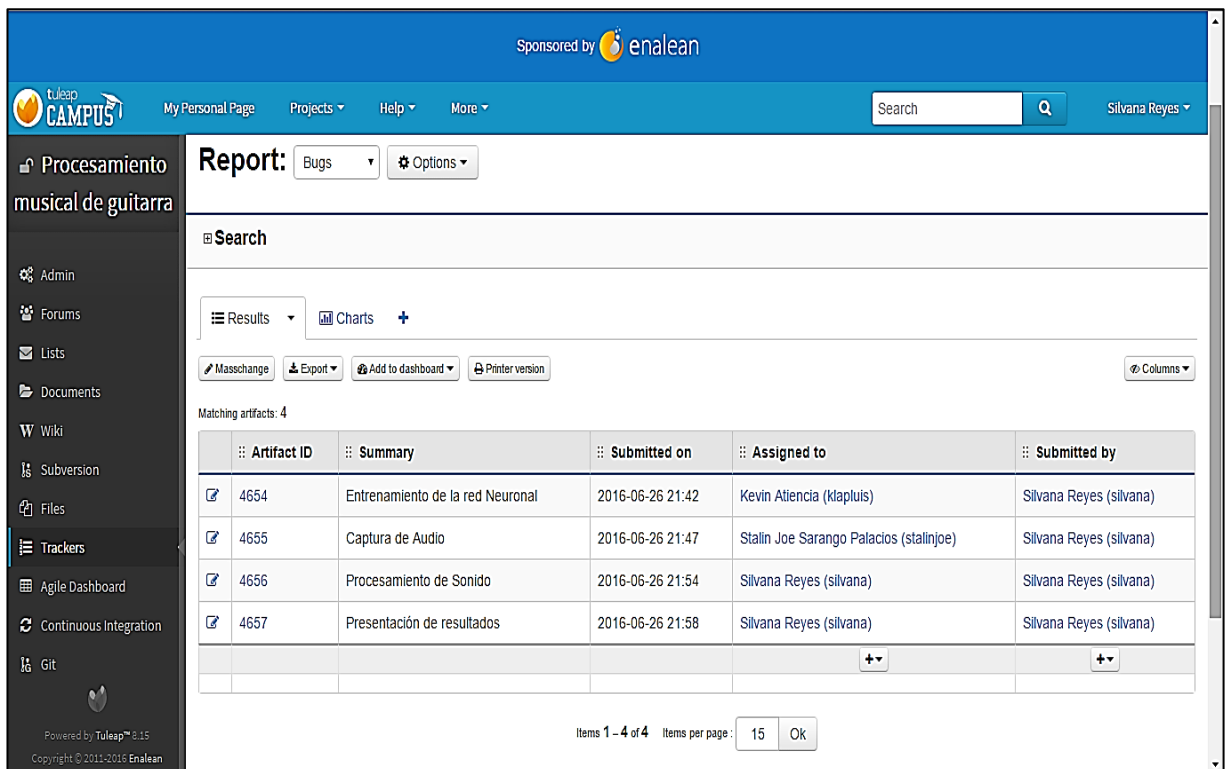


Figura 142: Visualización de los bugs del proyecto.

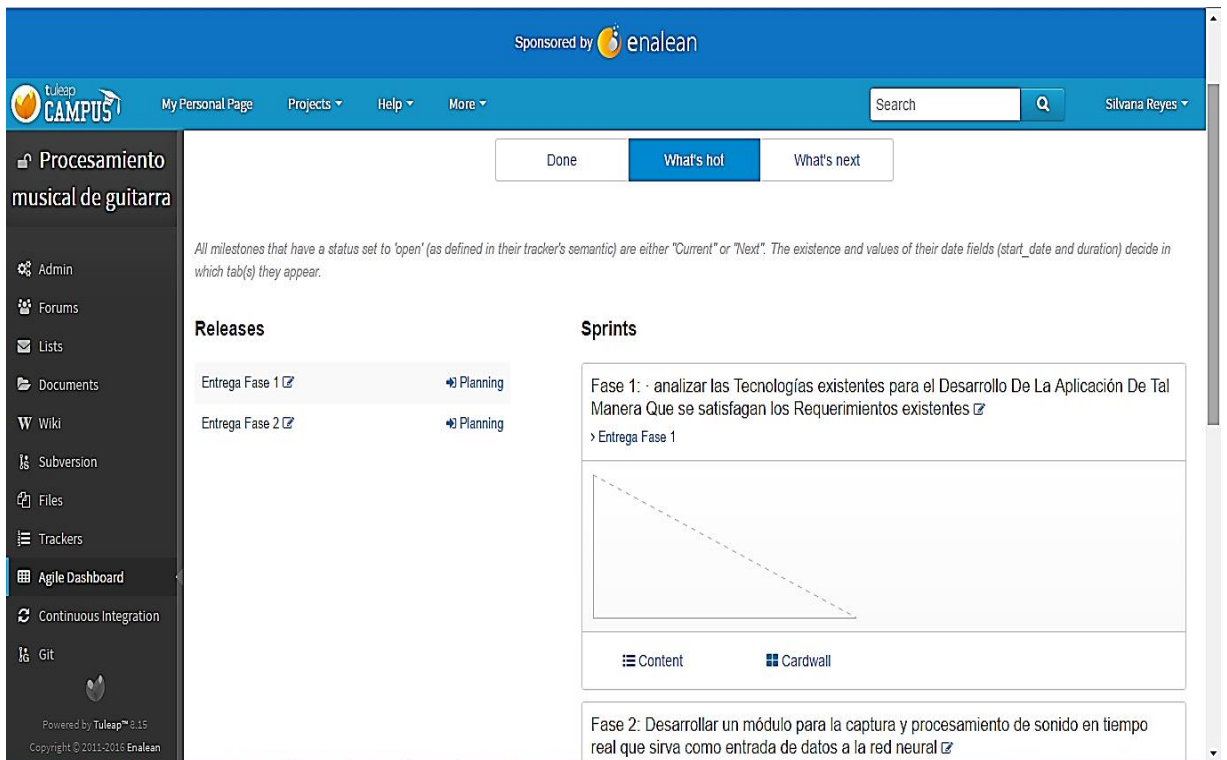


Figura 143: Visualización de las tareas cumplidas, en curso y por cumplirse del proyecto.

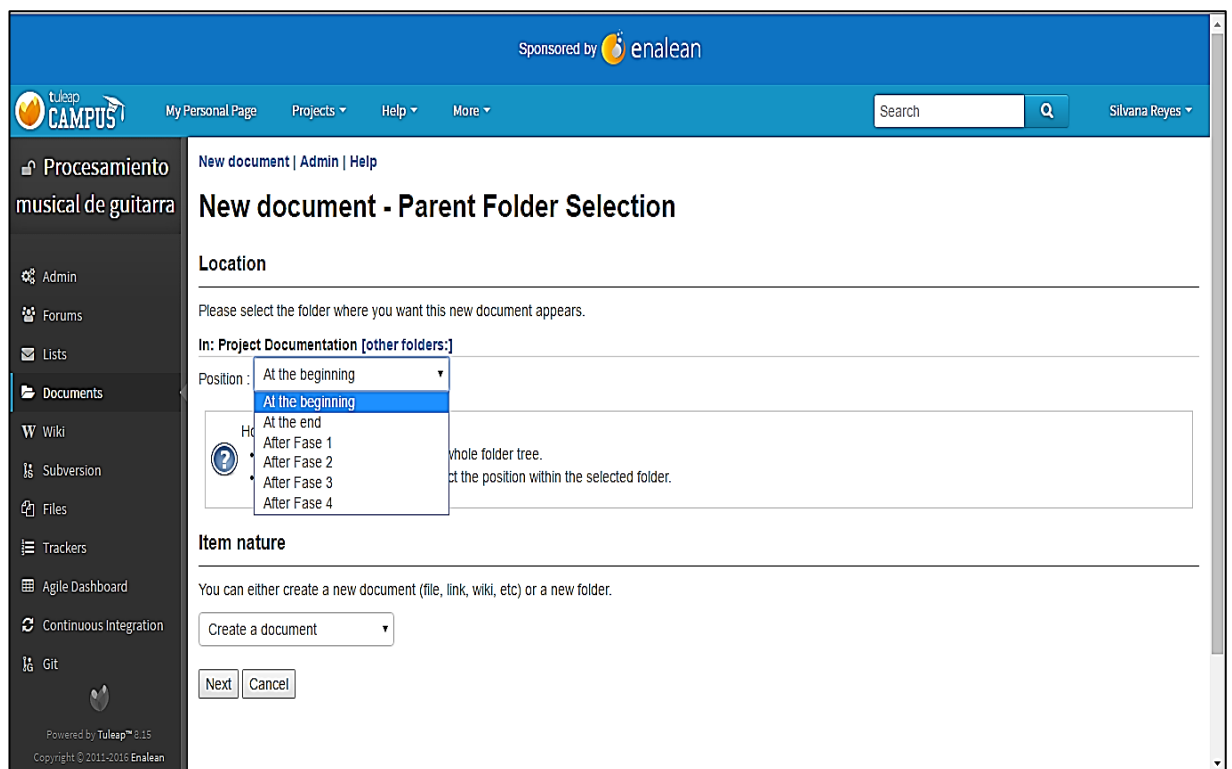


Figura 144: Creación de carpetas y/o documentos del proyecto, seleccionando su ubicación.

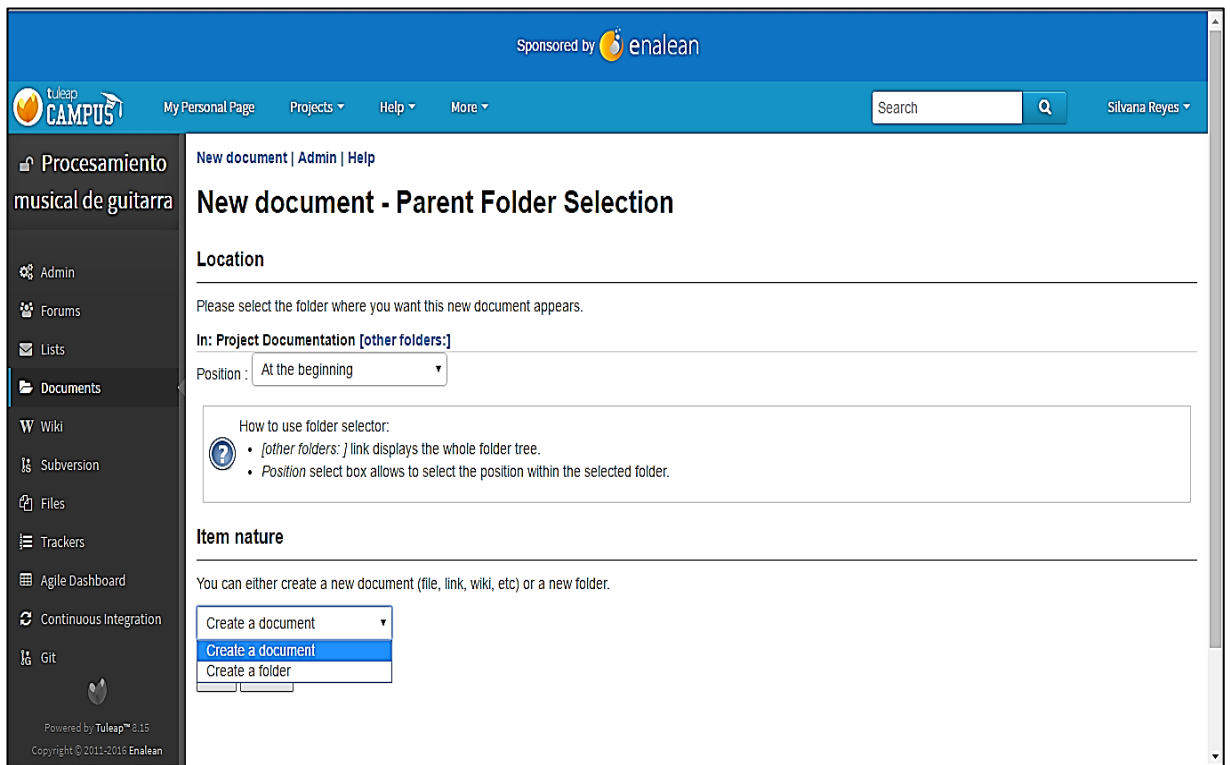


Figura 145: Selección entre la creación de una carpeta o documento en el proyecto.

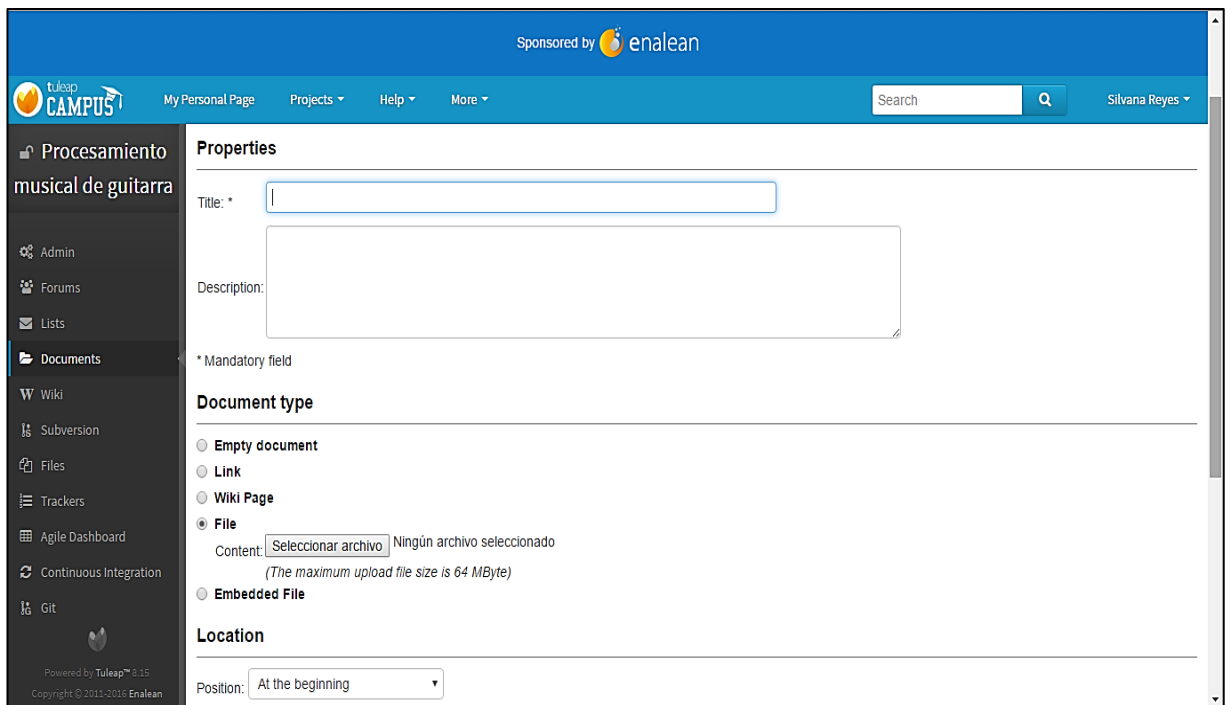


Figura 146: Ingreso de datos del documento del proyecto.

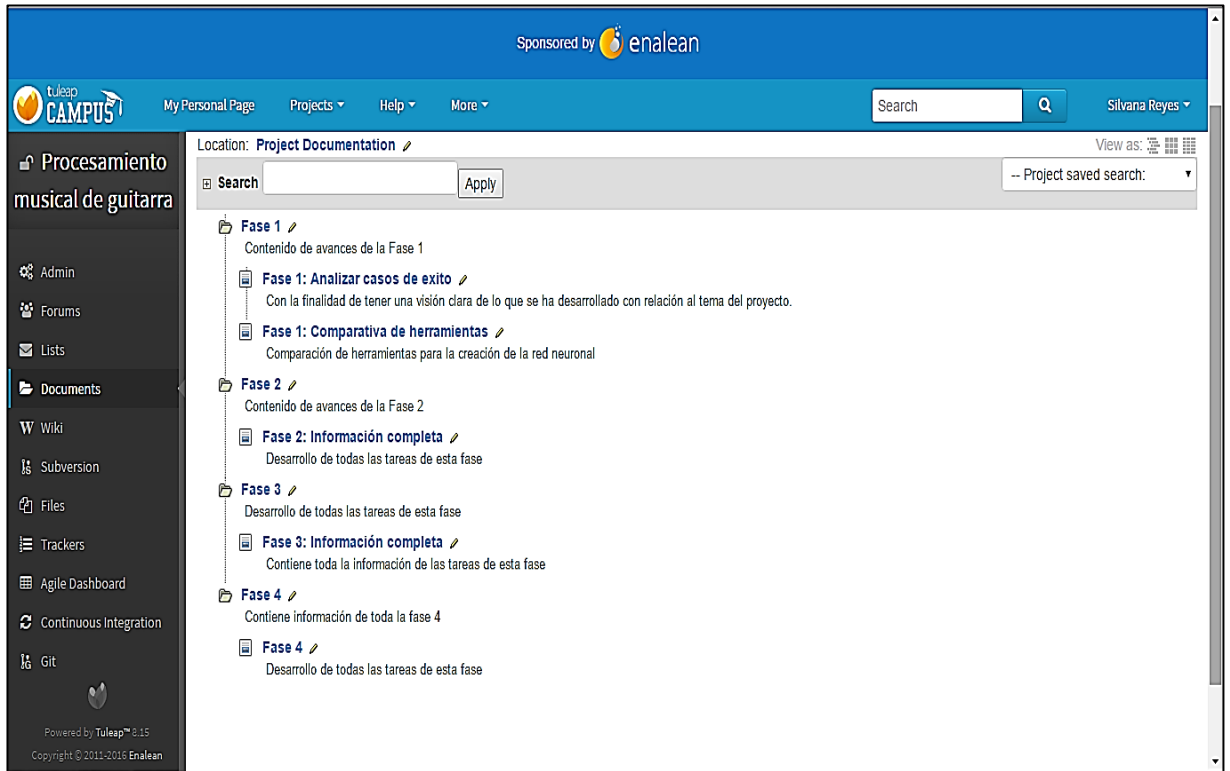


Figura 147: Visualización de carpetas y documentos del proyecto.

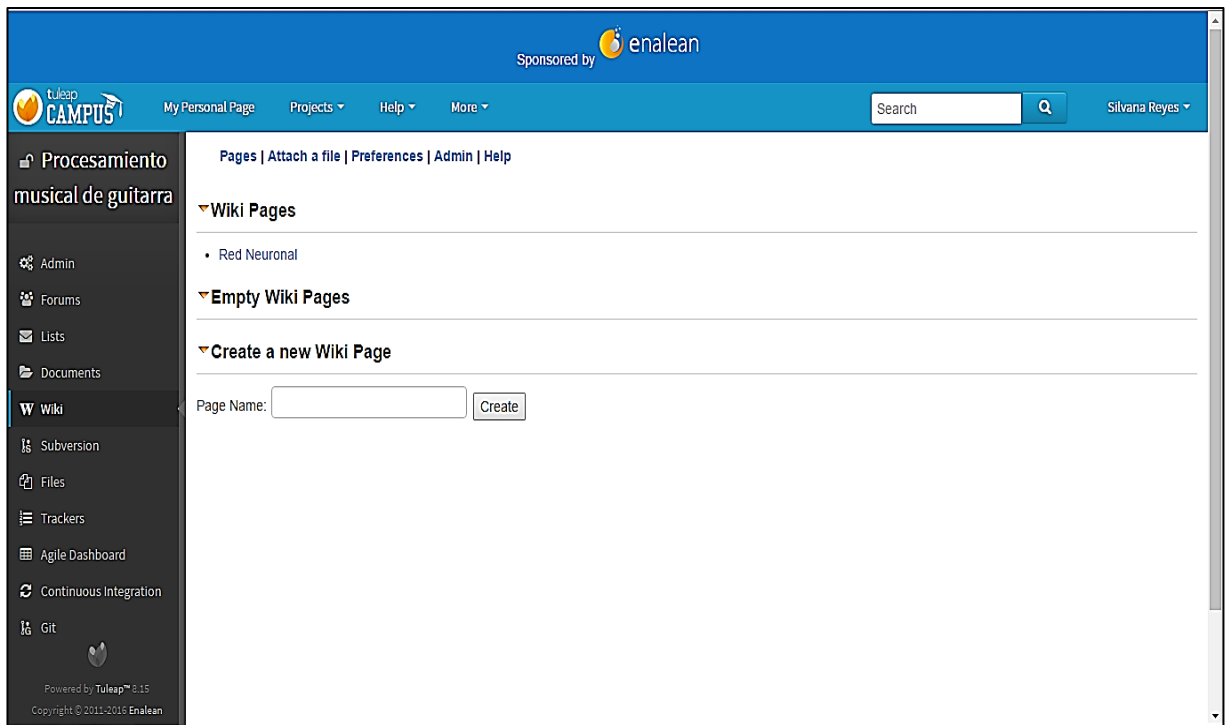


Figura 148: Creación de Wikis del proyecto.

The screenshot shows a Tuleap Wiki page for a project named 'Procesamiento musical de guitarra'. The page title is 'Red Neuronal'. The content is organized into sections: 2.1 Redes neuronales, 2.1.1 Fundamentos Biológicos, 2.1.2 Neurona Artificial, and 2.1.3 Redes Neuronales Artificiales. The text describes biological neurons and artificial neural networks, including their characteristics and functions. The page includes a sidebar with navigation options like Admin, Forums, Lists, Documents, Wiki, Subversion, Files, Trackers, Agile Dashboard, Continuous Integration, and Git. The top navigation bar shows 'Sponsored by enalean', 'My Personal Page', 'Projects', 'Help', 'More', a search bar, and the user name 'Silvana Reyes'.

Figura 149: Visualización de datos de la Wiki del proyecto.

The screenshot shows a Tuleap project page for 'Administración de la lista de correo'. The page title is 'Administración de la lista de correo'. The content includes sections for 'Añadir Lista de Correo' (Crear nuevas listas de correo) and 'Administrar Listas / Actualizar' (Manejo de correo existente (descripción del cambio, la privacidad ...)). The page includes a sidebar with navigation options like Administración, foros, Lista, Documentos, wiki, Subversión, archivos, rastreadores, Tablero de instrumentos ágiles, and Integración continua. The top navigation bar shows 'Patrocinado por enalean', 'Mi página personal', 'proyectos', 'Ayuda', 'Más', a search bar, and the user name 'Silvana Reyes'.

Figura 150: Creación de Listas del proyecto.

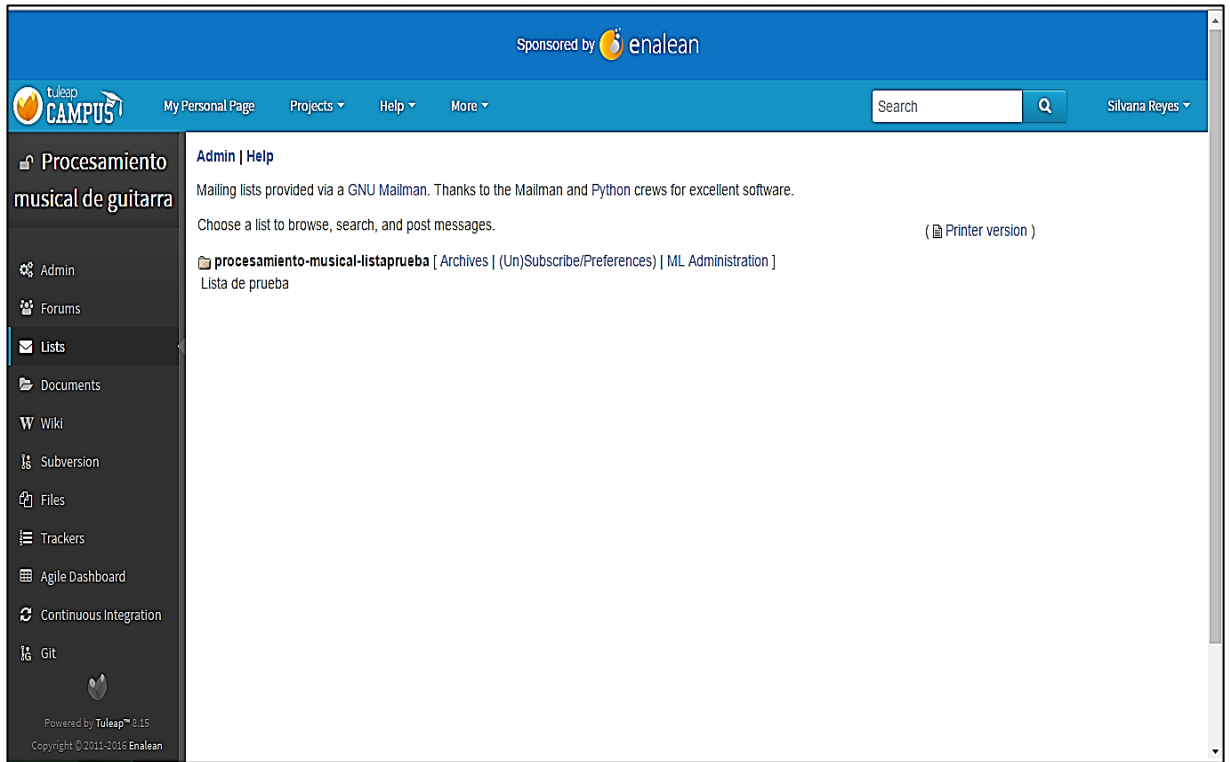


Figura 151: Visualización de las Listas del proyecto.

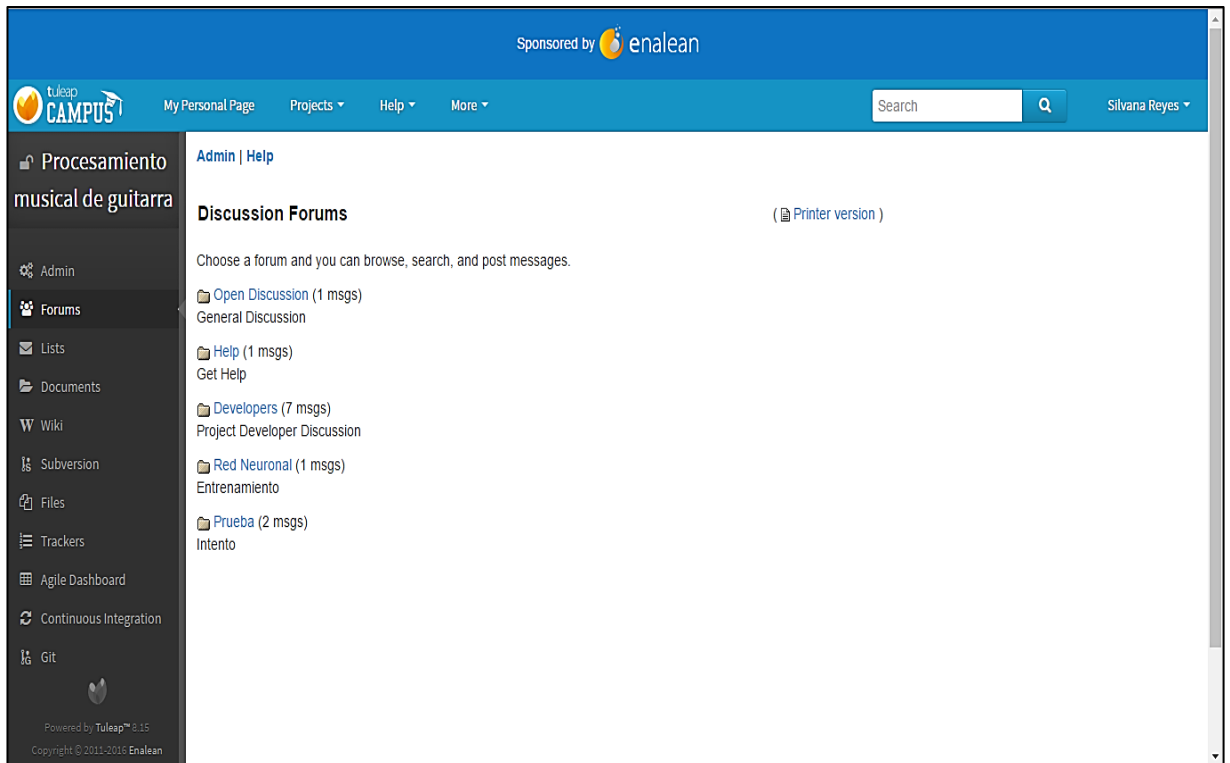


Figura 152: Visualización de los Foros del proyecto.

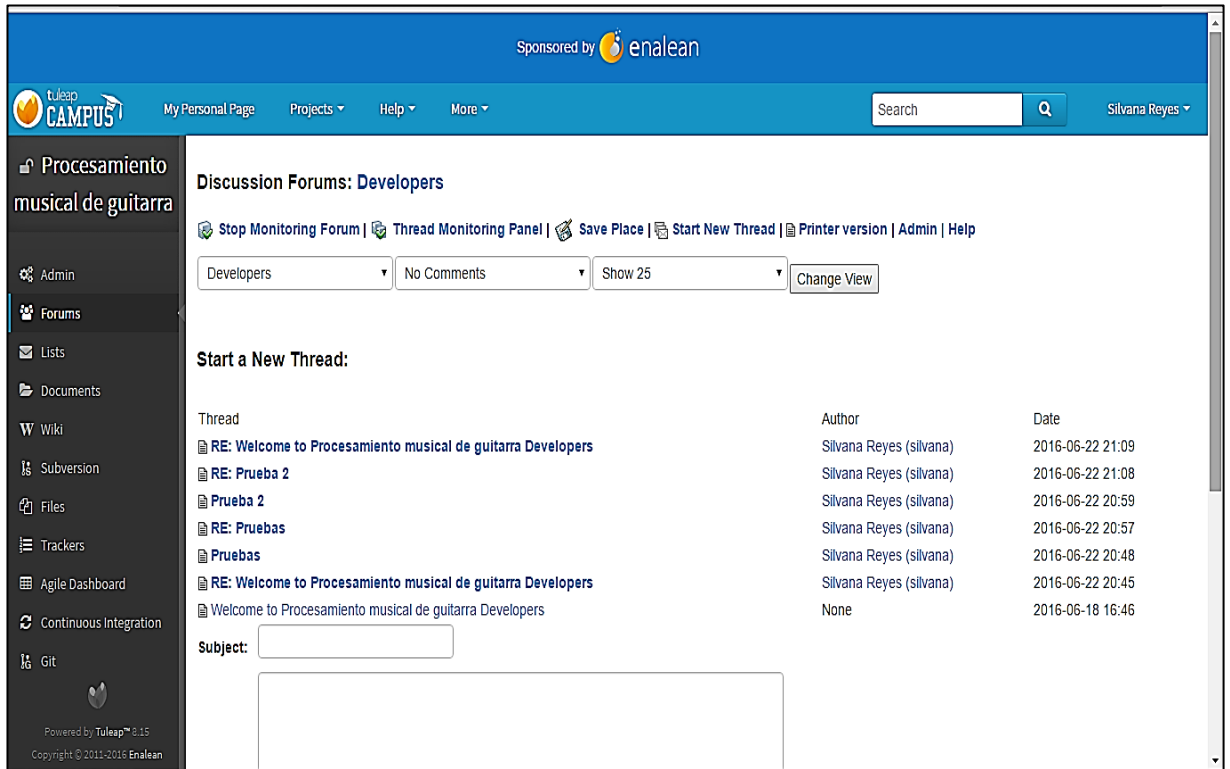


Figura 153: Visualización de los Foros del grupo de Desarrolladores del proyecto.

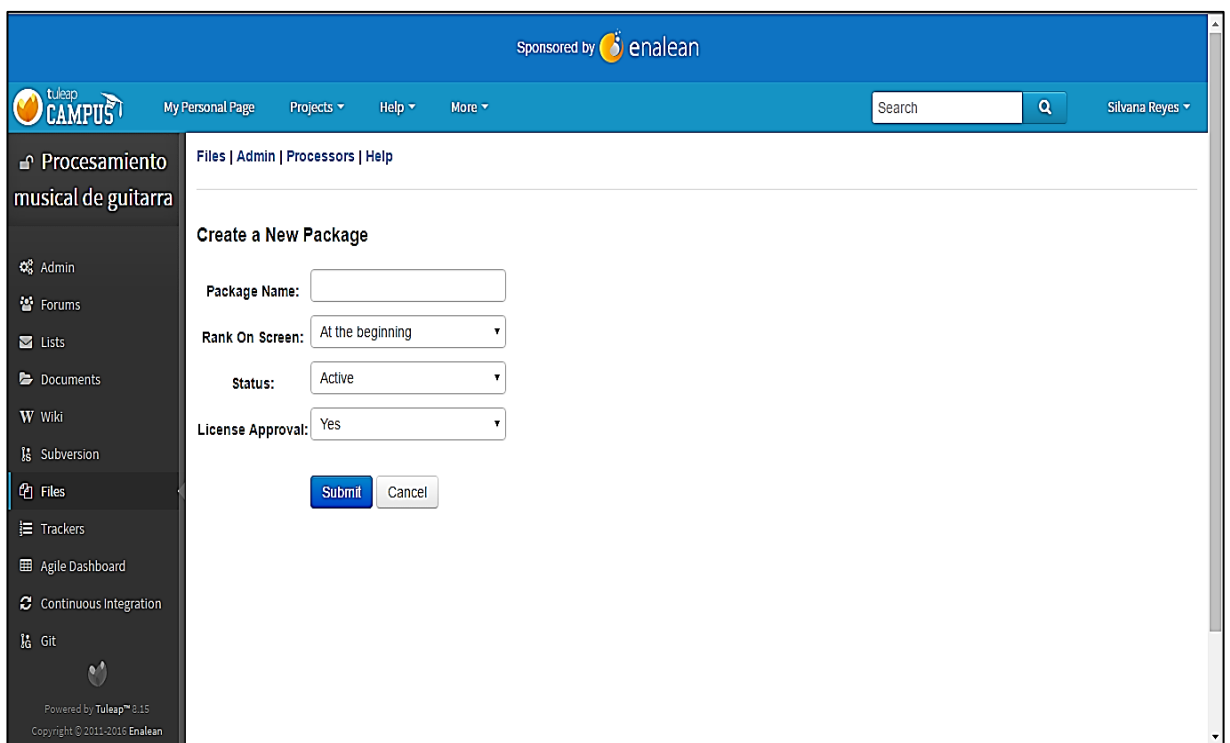


Figura 154: Creación de Archivos del proyecto.

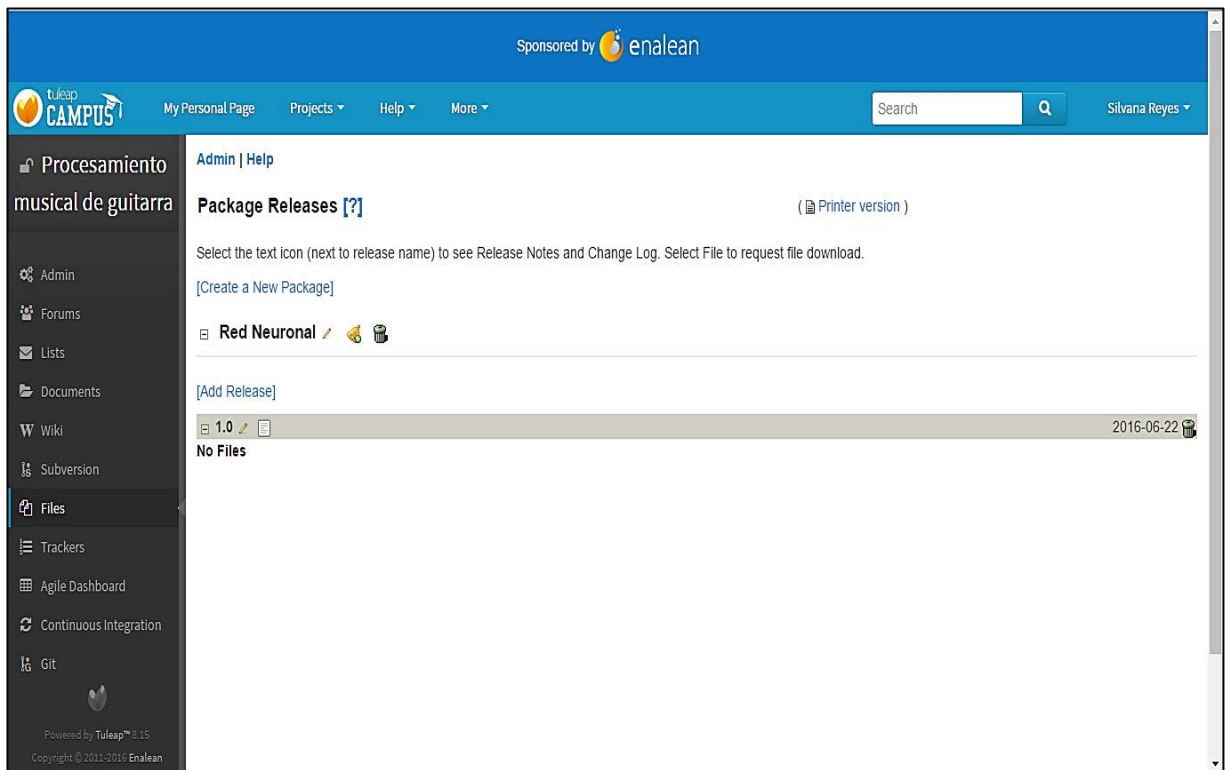


Figura 155: Visualización de los Archivos del proyecto.

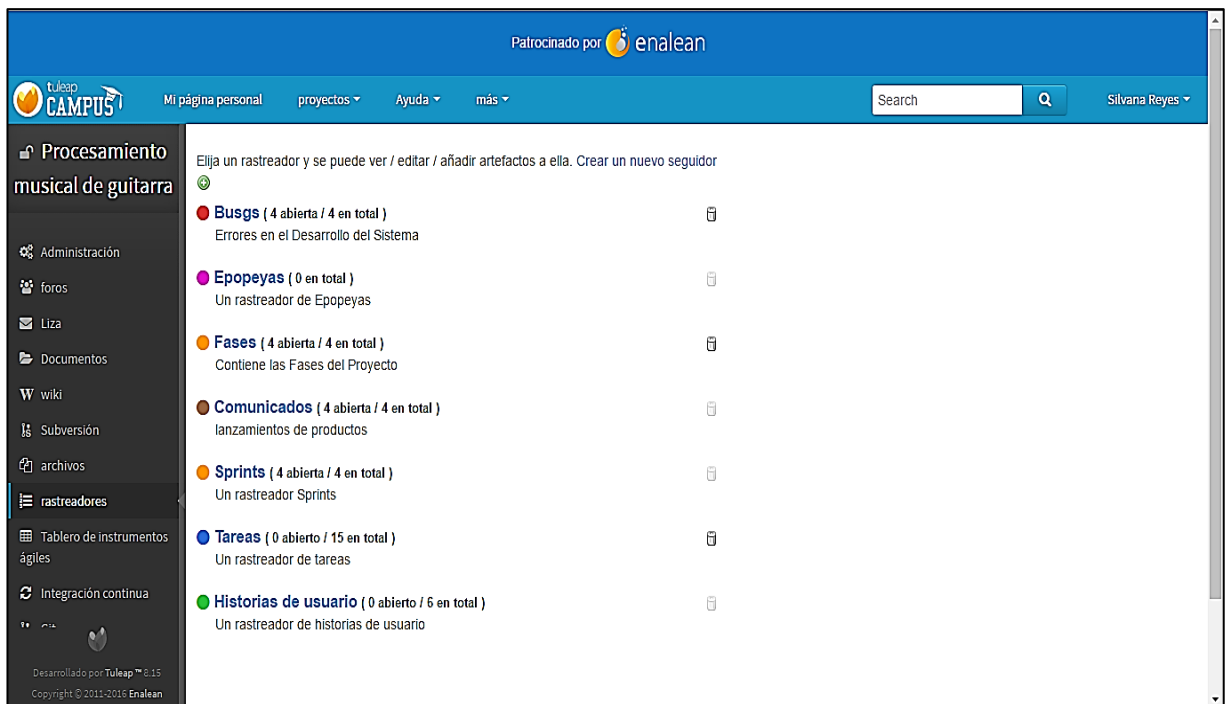


Figura 156: Visualización de los Rastreadores del proyecto.

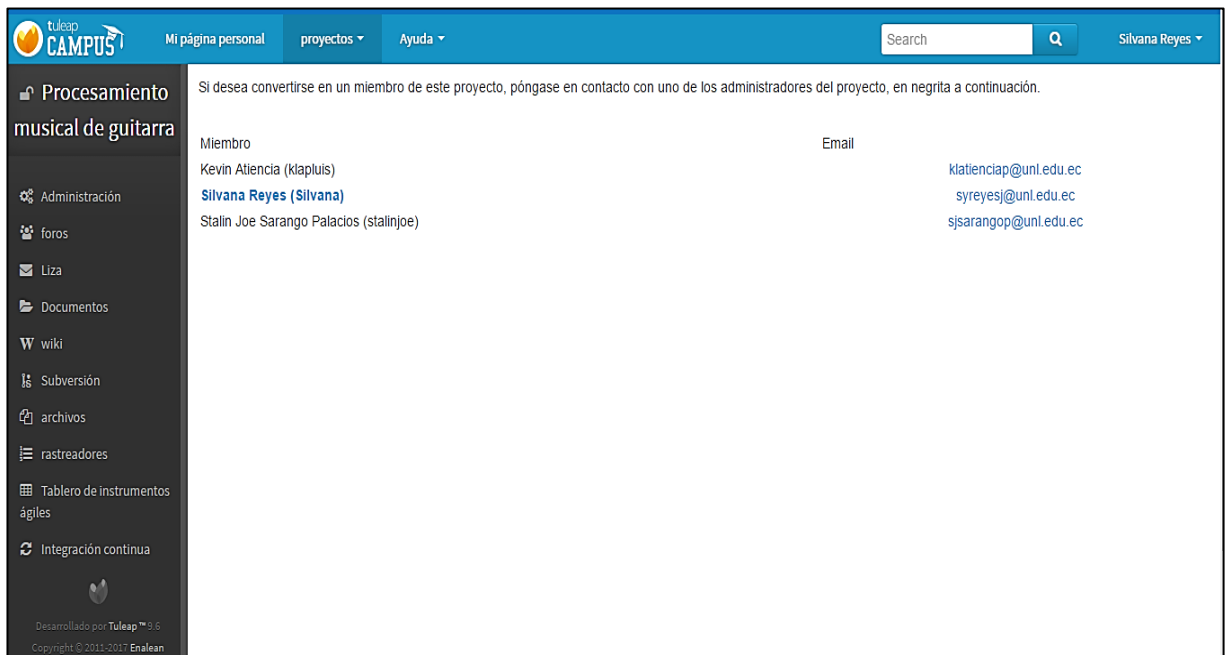


Figura 157: Asignación de miembros al proyecto.

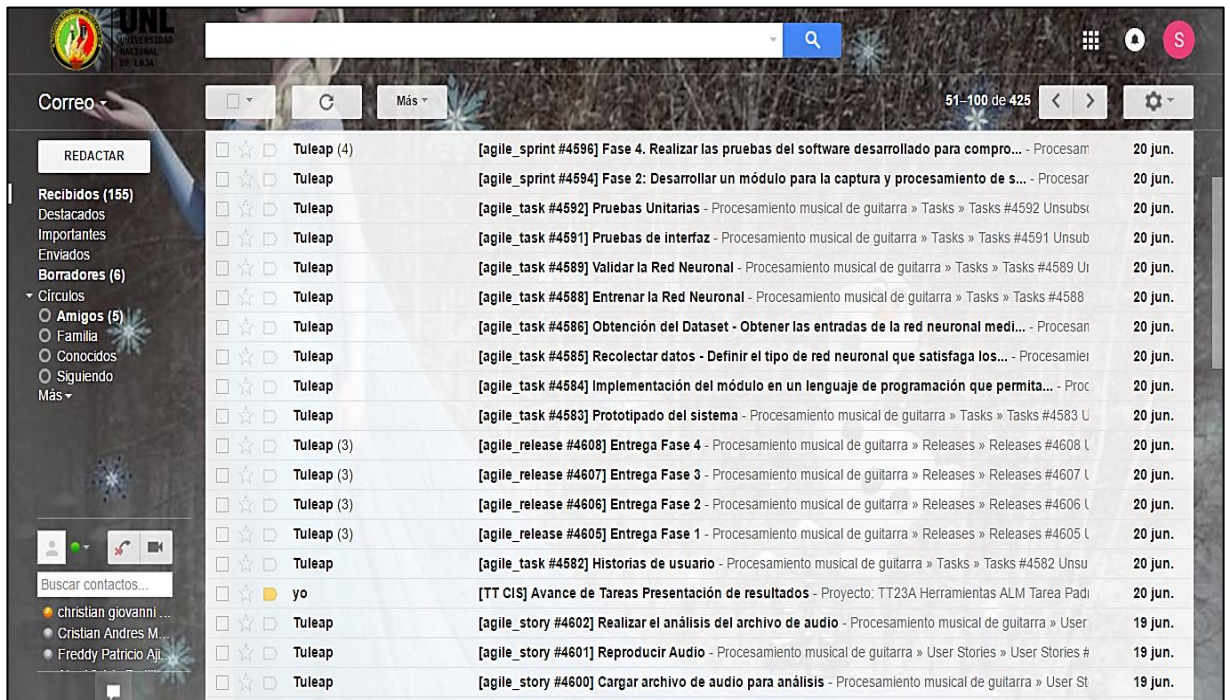


Figura 158: Visualización de las Notificaciones del proyecto.



Figura 159: Soporte de Tuleap en su página oficial.

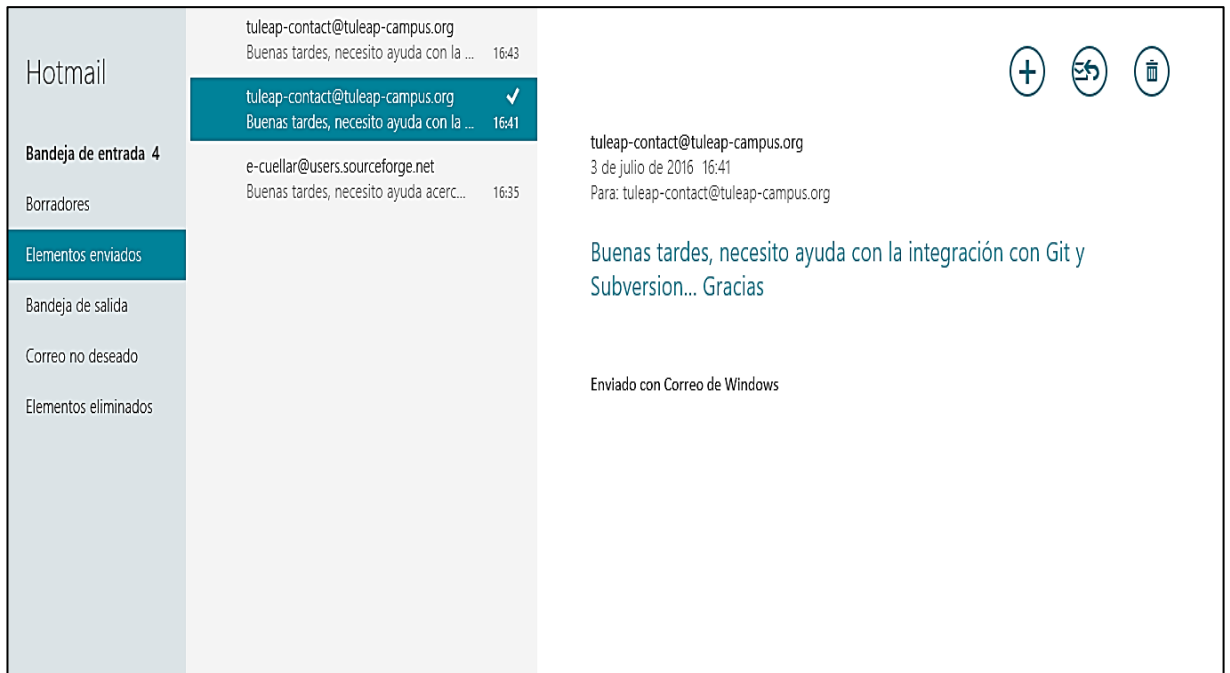


Figura 160: Soporte a través del envío de email.

Repository list Fork repositories Help Admin

Project repositories ?

Project repositories ▾

Repository	Description	Last push
mfernando93	-- Default description --	5 months ago by Silvana Reyes (silvana)
ninjaPeoject	-- Default description --	
redNeuronal	entrenamiento	

Reference creation ?

Name:

Figura 161: Versionamiento a través Git.

