



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**INSTITUTO DE POSGRADO**



**MAESTRÍA EN INGENIERÍA DE SOFTWARE**

“Impacto de la implementación del software de gestión para la fase de análisis de requerimientos funcionales en la Cooperativa Financiera Atuntaqui”

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

DIRECTORA:

Msc. Cathy Guevara

AUTOR:

ERIC DANIEL GUZMÁN CHAMORRO

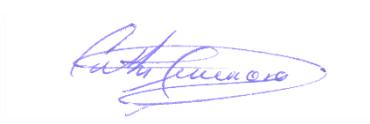
IBARRA - ECUADOR

**2018**

## APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de grado: **“Impacto de la implementación del software de gestión para la fase de análisis de requerimientos funcionales en la Cooperativa Financiera Atuntaqui”**, presentado por el **Ing. Eric Daniel Guzmán Chamorro**, para optar por el grado de Magister en Ingeniería de Software, ha sido guiado y revisado periódicamente y cumple normas establecidas en el Reglamento de Estudiantes de la Universidad Técnica del Norte, por lo que doy fe que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación pública y evaluación por parte del Jurado Examinador que se designe.

En la ciudad de Ibarra, 24 de abril del 2018



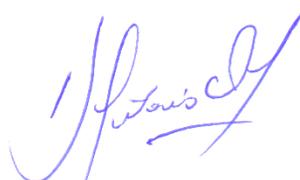
**Ing. Cathy Pamela Guevara Vega; MSc.**  
C.I.1002334835  
**TUTORA DE TRABAJO DE GRADO**

## APROBACIÓN DEL ASESOR

**“Impacto de la implementación del software de gestión para la fase de análisis de requerimientos funcionales en la Cooperativa Financiera Atuntaqui”**

Por: Ingeniero Eric Daniel Guzmán Chamorro.

Trabajo de grado de Maestría aprobado en nombre de la Universidad Técnica del Norte, por el Asesor, a los veinte y cuatro días del mes de abril de 2018.



**Ing. José Antonio Quiña Mera; MSc**  
C.I. 1002322384  
**ASESOR**

## DECLARACIÓN DE RESPONSABILIDAD

Ing. Eric Daniel Guzmán Chamorro.

### DECLARO QUE:

El proyecto de grado denominado “**Impacto de la implementación del software de gestión para la fase de análisis de requerimientos funcionales en la Cooperativa Financiera Atuntaqui**”, y bajo juramento que el contenido e información que se encuentra en el presente trabajo de investigación, ha sido desarrollado con base a una investigación exhaustiva y de mi autoría, respetando derechos intelectuales de terceros conforme se menciona en la sección bibliográfica de éste trabajo.



**Ing. Eric Daniel Guzmán Chamorro.**

C.I.: 1002785416

**AUTOR**

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

### 1. Identificación de la Obra.

La Universidad Técnica del Norte dentro del proyecto de Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

<b>DATOS DEL CONTACTO</b>			
<b>Cedula de ciudadanía:</b>	1002785416		
<b>Apellidos y Nombres:</b>	Guzmán Chamorro Eric Daniel		
<b>Dirección:</b>	Av. Luis Felipe Borja 1-69 y Alberto Haro		
<b>Email:</b>	e_cucho@hotmail.com		
<b>Teléfono fijo:</b>	062955169	<b>Móvil:</b>	0985553856
<b>DATOS DE LA OBRA</b>			
<b>Título:</b>	"Impacto de la implementación del software de gestión para la fase de análisis de requerimientos funcionales en la Cooperativa Financiera Atuntaqui"		
<b>Autor:</b>	Eric Daniel Guzmán Chamorro		
<b>Fecha:</b>	24-04-2018		
<b>SOLA PARA TRABAJOS DE GRADO</b>			
<b>Programa:</b>	Pregrado <input type="checkbox"/> Postgrado <input checked="" type="checkbox"/>		
<b>Título por el que opta:</b>	Magíster en Ingeniería de Software		
<b>Asesor/Director:</b>	Ing. Cathy Pamela Guevara Vega; MSc.		

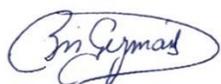
## **2. Autorización de uso a favor de la Universidad**

Yo, Eric Daniel Guzmán Chamorro, con cédula de identidad Nro. 100278541-6, en calidad de autor y titular de los derechos patrimoniales del trabajo de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

## **3. Constancia**

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

Ibarra, a los 24 días del mes de abril 2018



**Eric Daniel Guzmán Chamorro**  
CI:100278541-6

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

Yo, Eric Daniel Guzmán Chamorro, con cédula de ciudadanía N°. 100278541-6, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **“Impacto de la implementación del software de gestión para la fase de análisis de requerimientos funcionales en la Cooperativa Financiera Atuntaqui”**, trabajo de investigación elaborado para optar por el título Magister en Ingeniería de Software en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.



**Eric Daniel Guzmán Chamorro**  
CI:100278541-6  
**AUTOR**

**DEDICATORIA**

*A Dios, por haberme bendecido con su infinita bondad  
y amor, permitiéndome llegar a concluir con éxito esta  
etapa de vida académica, brindándome salud y  
paciencia para lograr mis objetivos.*

*A mi padre José, mi madre Miriam, mi hermana  
Frances y mi hermano Santiago, quienes con su  
ejemplo me han inspirado a ser mejor persona y por  
enseñarme que nunca se deja de aprender; a toda mi  
familia por brindarme siempre la fuerza y motivación  
para culminar la meta propuesta.*

*A mi tío Germán, por su lucha constante.*

*A mis amigos!, qué sería de la vida sin ellos.*

***Eric***

## AGRADECIMIENTO

*A la Universidad Técnica del Norte por brindarme la oportunidad de crecer en mi vida profesional, permitiéndome actualizar mis conocimientos y formación académica.*

*Al Ing. Roberto Peñafiel, Jefe de Sistemas de la Cooperativa Financiera Atuntaqui, por la apertura y el apoyo en el desarrollo de este trabajo.*

*A mi directora de tesis, MSc. Cathy Pamela Guevara Vega, por su dedicación, paciencia y tiempo en la guía y asesoría de la elaboración del presente trabajo.*

***Eric***

## ÍNDICE DE CONTENIDOS

Aprobación del tutor .....	ii
Aprobación del asesor .....	iii
Autoría.....	iv
Autorización de uso y publicación a favor de la universidad técnica del norte .....	v
Cesión de derechos del autor del trabajo de grado a favor de la universidad técnica del norte ...	vii
Dedicatoria.....	viii
Agradecimiento.....	ix
Indice de figuras.....	xiii
Índice de tablas .....	xiv
Resumen .....	xv
Abstract .....	xvi
<b>CAPÍTULO I: EL PROBLEMA.....</b>	<b>17</b>
<b>INTRODUCCIÓN .....</b>	<b>17</b>
1.1.    Antecedentes .....	18
1.2.    Planteamiento del problema.....	20
1.3.    Formulación del problema .....	23
1.4.    Justificación de la Investigación .....	23
1.5.    Objetivos de la Investigación.....	26
1.5.1.    Objetivo General.....	26
1.5.2.    Objetivos Específicos.....	26
1.6.    Hipótesis o Proposición .....	26
1.7.    Situación actual de la gestión de requerimientos funcionales .....	26
<b>CAPÍTULO II. MARCO REFERENCIAL.....</b>	<b>28</b>
2.1.    Marco Teórico .....	28
2.1.1.    Requerimientos.....	28
2.1.2.    Tipos de requerimientos .....	28
2.1.3.    Características de los requerimientos.....	31
2.1.4.    Técnicas de recopilación de requerimientos .....	32
2.1.5.    Requerimientos funcionales .....	34

2.1.6.	Ingeniería de requerimientos .....	34
2.1.7.	Importancia de la ingeniería de requerimientos .....	36
2.1.8.	Estándar IEEE 830-1998.....	37
2.1.9.	Estándar ISO/IEC/IEEE 29148-2011 .....	41
2.1.10.	Software de gestión.....	53
2.1.11.	Metodología SCRUM .....	53
2.2.	Marco Legal .....	54
CAPÍTULO III. METODOLOGÍA .....		56
3.1.	Descripción del área de estudio.....	56
3.2.	Tipo de investigación .....	56
3.3.	Diseño de la investigación .....	56
3.3.1.	Modalidad de investigación.....	56
3.3.2	Tipos o Niveles de Investigación.....	56
3.4.	Población y Muestra.....	57
3.5.	Métodos .....	57
3.6.	Estrategias Técnicas .....	58
3.7.	Instrumentos.....	58
CAPÍTULO IV. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS .....		59
4.1	Comparativa entre el estándar IEEE 830 y el estándar ISO/IEC/IEEE 29148....	59
4.2	Interpretación de resultados .....	62
CAPÍTULO V. PROPUESTA.....		64
5.1.	Antecedentes .....	64
5.1.1.	Análisis del proceso actual de desarrollo .....	64
5.2.	Propuesta de mejora al proceso de gestión de requerimientos .....	66
5.3.	Desarrollo del prototipo utilizando SCRUM .....	69
5.4.	Definición de Requerimientos .....	69
5.4.1.	Historias de usuario .....	69
5.4.2.	Product Backlog .....	73
5.4.3.	Sprint Planning .....	74
5.4.4.	Implementación y pruebas .....	75
5.5.	Medición del impacto del software de gestión.....	81
5.6.	Resultado de la propuesta .....	86
CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES .....		87
6.1	Conclusiones .....	87

6.2	Recomendaciones.....	88
	REFERENCIAS BIBLIOGRÁFICAS.....	89
	Anexo A: Glosario de Términos Técnicos.....	92
	Anexo B: Otras versiones del IEEE 830.....	94
	Anexo C: Ejemplo del documento de especificación de requerimientos.....	97
	Anexo D: Encuesta I.....	105
	Anexo E: Encuesta II.....	106
	Anexo F: Encuesta III.....	107
	Anexo G: Documento de Especificación de Requerimientos del Prototipo.....	108

## ÍNDICE DE FIGURAS

<i>Figura 1.</i> Relaciones entre tipos de requisitos. (Wiegers & Beatty, 2013) .....	30
<i>Figura 2.</i> Técnicas de gestión de requerimientos (Ibañez, 2010). .....	32
<i>Figura 3.</i> Esquema de Especificación de Requerimientos (IEEE 830). .....	37
<i>Figura 4.</i> Esquema de Especificación de Requerimientos. (ISO/IEC/IEEE) .....	44
<i>Figura 5</i> Visión General de SCRUM .....	54
<i>Figura 6</i> Proceso actual para solicitar requerimientos sobre un proyecto de software .....	65
<i>Figura 7</i> Sugerencia de mejora al proceso actual de gestión de requerimientos. ....	68
<i>Figura 8</i> Product Backlog del prototipo. ....	74
<i>Figura 9</i> Tiempos de carga de la aplicación. ....	77
<i>Figura 10</i> Vista de conexiones TCP realizadas al sistema. ....	77
<i>Figura 11</i> Vista de solicitudes por segundo realizadas al sistema. ....	78
<i>Figura 12</i> Vista del total de solicitudes realizadas al sistema. ....	78
<i>Figura 13</i> Gráfico Burndown.....	80

## ÍNDICE DE TABLAS

Tabla 1. <i>Cambios a los requerimientos iniciales de proyectos en producción.</i> .....	22
Tabla 2. <i>Tipos de requerimientos</i> .....	29
Tabla 3. <i>Características de requerimientos Estándar IEEE-830.</i> .....	31
Tabla 4. <i>Población y muestra de investigación.</i> .....	57
Tabla 5. <i>Comparación de características entre IEEE-830 y ISO/IEC/IEEE 29148.</i> .....	60
Tabla 6. <i>Sprint número uno.</i> .....	74
Tabla 7. <i>Sprint número dos.</i> .....	74
Tabla 8. <i>Sprint número tres.</i> .....	74
Tabla 9. <i>Escenarios de prueba uno.</i> .....	79
Tabla 10. <i>Escenarios de prueba dos.</i> .....	79
Tabla 11. <i>Escenarios de prueba tres.</i> .....	80
Tabla 12. <i>Escenarios de prueba cuatro.</i> .....	80
Tabla 13. <i>Escenarios de prueba cinco.</i> .....	80
Tabla 14. <i>Medición del impacto del protoipo</i> .....	82
Tabla 15. <i>Cálculo de valores para medir el impacto del prototipo</i> .....	84

**UNIVERSIDAD TÉCNICA DEL NORTE****INSTITUTO DE POSTGRADO****PROGRAMA DE MAESTRÍA EN INGENIERÍA DE SOFTWARE****“Impacto de la implementación del software de gestión para la fase de análisis de requerimientos funcionales en la Cooperativa Financiera Atuntaqui”**

**Autor:** Eric Daniel Guzmán Chamorro

**Tutor:** Cathy Pamela Guevara Vega

**Año:** 2018

**RESUMEN**

La presente investigación presenta el resultado del estudio realizado al proceso de desarrollo de software en la Cooperativa Financiera Atuntaqui. Determinando el impacto que tuvo la implementación de un software de gestión para la fase de análisis de requerimientos funcionales del desarrollo de software. La investigación tiene un enfoque cualitativo de carácter documental, descriptivo, exploratorio y de campo. La fase de análisis de requerimientos de software, es la etapa que lleva mayor cuidado y la que presenta mayor índice de impacto sea bueno o malo cuando el sistema se encuentra en producción. Para conocer este impacto se realizó un análisis del proceso actual en la gestión de requerimientos para así poder identificar el problema. La propuesta se enfoca en el desarrollo de un prototipo de software para gestionar los requerimientos funcionales. El objetivo es desarrollar un prototipo de software que permita al equipo de desarrollo tener un control adecuado de los requerimientos, conocer el estado de desarrollo, el número de modificaciones y mediante reportes poder tomar las decisiones adecuadas si algún proyecto tiene constantes cambios. Para el desarrollo del proyecto se utilizó: la metodología ágil Scrum, estándares internacionales para el desarrollo de software más arquitectura de software, que permiten finalizar proyectos de software de calidad. El proceso de desarrollo se basa en iteraciones o incrementos representados en los Sprint con sus respectivos requerimientos (historias de usuario y tareas), permitiendo generar un entregable del módulo correspondiente funcional. Finalmente se presenta el prototipo de software a los usuarios y se ejecuta las pruebas de funcionalidad necesarias generando informes de resultados en tiempo real que ayudan a la detección de problemas en el proceso de gestión de requerimientos.

**Palabras claves:** *Impacto, Gestión de requerimientos, Requerimientos funcionales, Desarrollo de Software, Metodología SCRUM, Estándares para el desarrollo de software.*

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

**“Impacto de la implementación del software de gestión para la fase de análisis de requerimientos funcionales en la Cooperativa Financiera Atuntaqui”**

**Autor:** Eric Daniel Guzmán Chamorro

**Tutor:** Cathy Pamela Guevara Vega

**Año:** 2018

**ABSTRACT**

The present investigation presents the result of the study carried out to the software development process in Cooperativa Financiera Atuntaqui. Determining the impact that the implementation of a management software had for the analysis phase of functional requirements of software development. The research has a qualitative approach of documentary, descriptive, exploratory and field nature. The software requirements analysis phase is the stage that takes the greatest care and the one with the highest impact index, whether good or bad when the system enters production. In order to know this impact, an analysis of the current process in requirements management has been carried out in order to identify the problem. The proposal focuses on the development of a prototype software to manage the functional requirements. The objective is to develop a software prototype that allows the development team to have an adequate control of the requirements, to know the development status, the number of modifications and through reports to be able to make the appropriate decisions if a project has constant changes. For the development of the project we used: the Scrum agile methodology, international standards for software development plus software architecture, which allow to finalize quality software projects. The development process is based iterations or increments represented in the Sprint with their respective requirements (user stories and tasks), allowing to generate a deliverable of the corresponding functional module. Finally, the software prototype is presented to the users and the necessary functionality tests are carried out generating real-time results reports that help to detect problems in the requirements management process.

**Keywords:** *Impact, Requirements Management, Functional Requirements, Software Development, SCRUM Methodology, Standards for software development..*

## **CAPÍTULO I: EL PROBLEMA**

### **INTRODUCCIÓN**

Las organizaciones que su giro del negocio no es producir software, deben tomar conciencia de la importancia que tiene realizar un proceso adecuado para realizar un sistema.

Todas las etapas tienen un cierto grado de importancia que se lo debe realizar adecuadamente para que el producto final no presente fallas cuando ya se encuentre en funcionamiento.

La utilización del estándar IEEE-830 ha ayudado a la mayor parte de empresas a seguir los lineamientos adecuados para un correcto proceso de desarrollo de software. Si bien es cierto que el estándar se publicó en el año 1998, su alcance y adaptación en la actualidad tiene un impacto beneficioso para aquellas empresas que todavía lo aplican.

En la primera parte del proyecto se realizó un estudio al proceso de desarrollo de software, se analizaron las ventajas y desventajas del documento de gestión de requisitos, se analizaron las causas por las que algunos proyectos presentan varias modificaciones a sus requerimientos originales, se establecieron los efectos de no aplicar normas internacionales en el proceso y se analizaron las posibles soluciones.

En la siguiente fase del proyecto, se aplicó la guía para el documento de especificación de requerimientos basado en el estándar IEEE-830, debido a que es el que se ajustaba mejor al tamaño del equipo de desarrollo de la institución. Se desarrolló un prototipo de software que ayude a la gestión de requerimientos funcionales, para tener un control adecuado de cada proyecto, requisitos, personal a cargo y de ser el caso, poder tomar acciones correctivas si se tiene varios cambios a uno o más requisitos, de esta manera se distribuye mejor los tiempos en la finalización de un proyecto.

En la última fase se aplicó la fórmula para medir el impacto que tuvo el prototipo de software de gestión de requerimientos para verificar si el prototipo complementa de mejor manera el desarrollo de software de calidad en la institución.

### 1.1. Antecedentes

En los últimos años, el interés de las empresas de Latinoamérica por desarrollar software de calidad y mejorar el nivel de producción, se ha visto reflejado en los datos que muestra el CMMI (Capability Maturity Model Integration) Institute, el cual ofrece un modelo que ayuda a las empresas en más de setenta países a mejorar y alcanzar sus objetivos comerciales. Este instituto permite a las organizaciones certificarse en varios niveles de acuerdo con el grado de madurez en los procesos de desarrollo del producto, es así como Brasil, México, Colombia, Argentina y Chile encabezan la lista de los países de Latinoamérica con mayores certificaciones en calidad del software. En la lista, Ecuador posee dos certificaciones. (CMMI Institute, 2016, pp. 26-27).

Las empresas de software que se destacan en Ecuador son COBISCORP y Kruger Corporation con niveles de madurez CMMI 3, las cuales han madurado los procesos y han agregado mejor calidad a sus productos. Lo positivo de esta iniciativa es que se convierten en modelos hacia otras empresas para que adopten mejoras a los procesos, permitiendo que el software desarrollado en el país sea competitivo a nivel mundial (CMMI, 2016).

La importancia del software bien desarrollado radica en la Ingeniería de requerimientos por ser la fase inicial y la que va a servir como pilar a las siguientes fases del desarrollo del sistema. (Simbaña Saransig, Simbaña Quinsasamin, Hinojosa Raza, & Ron Egas, 2015, pp. 26-27).

Según los autores (López Echeverry, Cabrera, & Valencia Ayala, Introducción a la calidad del software, 2008, p. 328) indica varios aspectos por los cuales un software no alcanza niveles de calidad y menciona los siguientes aspectos que se debe evitar:

*“la construcción de software presenta dificultades tales como insuficiencia en la especificación de requisitos, diseño poco profundo, mala gestión de la configuración, poca flexibilidad para la incorporación de cambios, prolongado tiempo de duración y aumento en los costos”.*

*“En el desarrollo de software, el control de la calidad es realizado por el mismo desarrollador, que dispone de poco tiempo, cuando lo tiene”.*

Como estos factores inciden directamente en la calidad del producto, se debe delegar la revisión al rol específico para agregar un valor adicional al producto final.

De igual manera la generación de requerimientos no tiene la importancia necesaria en algunas empresas de desarrollo, en las cuales los tiempos aplicados a esta fase son cortos, no tienen el personal con los conocimientos necesarios y tampoco aplican el estándar adecuado; eso como consecuencia lleva a que el proceso de desarrollo sufra cambios considerables y los tiempos de entrega sean modificados a sus cronogramas ya establecidos. (López Echeverry, Cabrera, & Valencia Ayala, *Introducción a la calidad del software*, 2008, p. 328).

Por esta razón es necesario la aplicación de los modelos de madurez del CMMI y la adopción de un estándar en la fase de requerimientos funcionales, para que sea tomada de manera consciente por parte de las empresas mejorando la planificación y ejecución del ciclo de vida del software (López Echeverry, Cabrera, & Valencia Ayala, *Introducción a la calidad del software*, 2008, p. 328).

La Cooperativa Financiera Atuntaqui, al ser una institución financiera que está a la vanguardia del Sistema Nacional Cooperativo de todo el Ecuador, a demostrado a lo largo de sus 54 años de vida institucional los siguientes valores: honestidad, confianza, solvencia, seriedad y responsabilidad social al servicio de todos sus socios y clientes; factores que han permitido un desarrollo constante, gracias al trabajo efectuado con capacidad y eficiencia de directivos y funcionarios que han puesto en práctica el slogan : "La Caja Fuerte del Ecuador", posee nueve oficinas ubicadas en la zona norte del Ecuador, que comprenden las provincias de Imbabura y Pichincha, en las cuales los socios y clientes han depositado la confianza en esta grande institución financiera desde hace varios años atrás.

En esta institución el área tecnológica es importante debido a los diferentes tipos de transacciones que se realiza, por lo que el área de desarrollo de software se constituye un pilar fundamental para los aplicativos internos que se adaptan al giro de negocio de cada departamento. Se utiliza la metodología RUP y SCRUM para el desarrollo de los diferentes proyectos de software limitando las funciones de los roles de las metodologías a ser distribuidos entre el personal del equipo de desarrollo.

Como la Cooperativa Financiera Atuntaqui no es una empresa de desarrollo, los recursos humanos y el conocimiento en temas claves del proceso de calidad de software no son los adecuados, la aplicación de normas o estándares no se la realiza en las diferentes fases del desarrollo porque el tiempo asignado a cada proyecto es corto y el personal a cargo es escaso; por

esta razón se ha visto la necesidad de mejorar el proceso de la gestión de requerimientos debido a que al ser la fase inicial de desarrollo, es la que va a influir de manera directa con las siguientes fases en el desarrollo del sistema.

Al poder crear conciencia con el personal involucrado sobre la importancia de aplicar un software de gestión para la fase de análisis de requerimientos, se puede conseguir una mejora significativa en la funcionalidad del software, satisfacer completamente las necesidades del cliente y reducir los costos de mantenimiento.

Además de aplicar un estándar, se debe comprender de manera íntegra lo que el documento de la norma trata de enseñar, de igual manera se debe conocer los beneficios que trae al producto final, es decir, que al existir una mejora continua del proceso y que, al estudiar las mejores prácticas de los estándares, el software al ser entregado al cliente tiene un mejor rendimiento y mejor funcionalidad.

Apoyarse de una herramienta de gestión que pueda mejorar el actual proceso de requerimientos, beneficia el trabajo del actual personal y de igual manera en el caso de existir nuevos integrantes en el equipo de desarrollo, ellos comprenderán todo el proceso en la revisión de los documentos previamente para el sistema.

Si el proceso de gestión mejora, tendrá un impacto positivo en el equipo de trabajo, mejorará la autoestima del personal y promoverá la capacitación constante en las áreas estratégicas de procesos de calidad y de gestión del software.

## **1.2. Planteamiento del problema**

En el departamento de desarrollo de software de la Cooperativa Financiera Atuntaqui, una persona cumple varias funciones como: analista, arquitecto, diseñador, programador y usuario de pruebas, los nombres de los roles pueden cambiar dependiendo de la metodología de desarrollo que se va a utilizar para un sistema, puede ser RUP o SCRUM, esto ocasiona que el trabajo desempeñado en ese rol se lo realice con limitada experiencia y conocimientos básicos, estos factores inciden directamente en el tiempo de desarrollo, los cronogramas de entregables se modifican con frecuencia, el cambio a los requerimientos cuando el producto ya se encuentra en

producción, el mantenimiento correctivo constante generando molestias en el equipo de desarrollo y al usuario final porque el software no satisface las necesidades al problema propuesto.

Cada una de las etapas de desarrollo tienen su importancia y relevancia, la etapa inicial de análisis de requerimientos necesita mayor atención y trabajo porque es la que guiará a las siguientes fases del ciclo de vida del software; obtener requerimientos es una parte importante y difícil en los proyectos de software, la mala calidad de cada uno de ellos motiva al fracaso del proyecto (Hinojosa, Raura, Fonseca, & Dieste, 2015, pp. 3-5). la gestión de requerimientos la debe realizar el personal calificado y con experiencia para que los requerimientos sean:

*“correctos, completos, consistentes, inequívoco, categorizados por relevancia, verificables, modificables y trazables” (IEEE-830, 1998).*

En un conversatorio mantenido en el área de sistemas de la Cooperativa Financiera Atuntaqui, se pudo observar que actualmente las causas que hacen deficiente el proceso en la fase de gestión de requerimientos son las siguientes:

1. La solicitud de requerimientos por parte de los usuarios contiene errores. Los usuarios crean los requisitos generalizados y sin detallarlos adecuadamente ya que lo realizan sin el soporte del especialista, dificultando la comprensión de la necesidad para el problema propuesto.
2. El tiempo para gestionar requerimientos no es el adecuado, es corto y limitado. Los usuarios presentan el documento con los requerimientos en pocos minutos y con ideas generales de sus necesidades.
3. No se cuenta con un especialista en requerimientos. Los desarrolladores cumplen varios roles sin tener los conocimientos y la experiencia necesaria para desempeñar las funciones de gestionar los requerimientos y trabajar con los usuarios.
4. Capacitación limitada en las áreas de gestión de requerimientos. No se realiza capacitaciones en procesos de desarrollo de software.
5. No se aplica un estándar documental para la toma de requerimientos funcionales. El documento de gestión de requerimientos no se basa en ningún estándar para la gestión de requerimientos.
6. Limitado conocimiento en los estándares, normas internacionales e instrumentos utilizados para la gestión de requerimientos en el desarrollo de software.

A partir de estas causas, en el conservatorio de igual manera se pudo identificar los siguientes efectos:

1. El tiempo de desarrollo de la aplicación se torna extenso y molesto para el equipo de desarrollo. El cronograma de desarrollo es modificado con frecuencia y los proyectos de igual manera han tenido que ser aplazados.
2. Gran cantidad de modificaciones después del desarrollo. En producción se realiza constantemente cambios y modificaciones para corregir los defectos del producto.
3. Generación de un ambiente tenso y no productivo con el equipo de trabajo. Las molestias en el equipo de desarrollo son notorias porque se debe realizar doble trabajo, mayor esfuerzo al construir el software y corregir las fallas que presentan.
4. El software no cumple con estándares de calidad. Cuando ya se encuentra en producción presenta defectos y es necesario un mantenimiento permanente y en algunos casos no satisface las necesidades del usuario.

Como casos puntuales para sustentar lo descrito anteriormente se tiene un Proyecto A en el cual se realizaron nueve cambios cuando el software estuvo en producción, estos cambios corresponden a que en los requerimientos originales no se contemplaron escenarios entorno al giro del negocio, estos cambios provocaron la suspensión temporal de la aplicación y generaron una disminución en el resultado proyectado por las áreas operativas.

De igual manera sucedió con el Proyecto B, en donde hubo quince cambios apenas el software estuvo en producción, en este proyecto los requerimientos fueron variables durante la fase de desarrollo y no se lograba llegar a un acuerdo y definir las funcionalidades claras del requerimiento porque en este caso, existió una rotación de jefaturas quienes estaban a cargo del proyecto.

A continuación, se detalla una tabla con los cambios de algunos proyectos para una mejor comprensión de los cambios realizados:

**Tabla 1.** *Cambios a los requerimientos iniciales de proyectos en producción.*

Nombre	Total de Requerimientos	Cambios a los Requerimientos	Porcentaje
--------	----------------------------	---------------------------------	------------

<b>Proyecto A</b>	22	15	68.18%
<b>Proyecto B</b>	18	9	50%
<b>Proyecto C</b>	25	2	8%
<b>Proyecto D</b>	10	9	90%
<b>Proyecto E</b>	6	1	16.66%

**Fuente:** Autor

Como proyectos medianos los cambios pueden ser mínimos, pero a la final afectan al producto final, la siguiente afirmación del Dr. Raymond Yeh de hace varios años atrás tiene mucha coherencia con lo que actualmente sucede con el desarrollo de un sistema:

*“La carencia de buenos requisitos ha sido la causa del fracaso de proyectos con presupuestos de millones de dólares, ha impedido el desarrollo productivo, y ha sido el mayor contribuyente de los costos elevados del mantenimiento del software” (Yeh, 1990).*

Por esta razón, si no realizan acciones correctivas en la fase análisis de requerimientos, el software que se desarrolla seguirá presentando falencias a la hora de poner en producción, continuará generando molestias al equipo de desarrollo y a los usuarios finales, además de retrasar los tiempos de entrega, elevar el costo y continuar con el mantenimiento correctivo repercutiendo en la planificación de otros proyectos.

### **1.3. Formulación del problema**

¿La deficiencia en la gestión de la fase de análisis de requerimientos afecta a la calidad funcional del software en la Cooperativa Financiera Atuntaqui?

### **1.4. Justificación de la Investigación**

En el trabajo presentado por Sanguino Marco de título “Formulación de un marco de referencia para la personalización de requerimientos en una fábrica de desarrollo de software bancario para mejorar la atención de las entidades financieras” (2014), concluye que una buena gestión de requerimientos basándose en estándares y normas internacionales permite la reducción del tiempo de desarrollo, mejorando la productividad y evitando la sobrecarga de trabajo al equipo de desarrollo.

Además, concluye que los efectos positivos de una correcta gestión de requerimientos son la confianza de los socios hacia la institución y la confianza de los empleados en los sistemas desarrollados internamente. (Sanguino, 2014).

Como el proceso de gestionar los requerimientos lleva tiempo y tiene su complejidad (Dávila, Melendez, & Flores, 2006), el uso de herramientas se ha convertido en un aspecto importante al desarrollar un sistema. Considerando el tamaño y complejidad del desarrollo, el uso de las herramientas se ha vuelto esencial (Rosique, Jiménez, & Sánchez, 2010, pp. 47-48).

Estas herramientas son llamadas CASE (Computer-Aided Software Engineering) las cuales son mencionadas en el trabajo de Trung Hung Vo con el título “Introduction to Software Engineering” (2009) en donde se refiere a las herramientas CASE como programas que se utilizan para apoyar las actividades de proceso de software, tales como análisis de requisitos, modelado de sistemas, depuración y pruebas.

*“The acronym CASE stands for Computer-Aided Software Engineering. It covers a wide range of different types of program which are used to support software process activities such as requirements analysis, system modelling, debugging and testing.”* (Vo, 2009, pp. 18-19).

En este tipo de herramientas CASE se puede encontrar a los programas de gestión de requerimientos, los cuales permiten mejorar el proceso de la primera etapa de desarrollo y de todo el ciclo de vida del software.

En el artículo presentado por Arturo W. Laredo titulado “Indicadores de calidad en el desarrollo de Software” (2011) concluye que se debe mejorar y optimizar constantemente los procesos en cada etapa del proyecto de desarrollo de software, lo que garantiza que el producto propuesto alcance la calidad esperada.

Otro artículo importante es el de Michael Arias Chaves, “La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software” (2006). En el cual hace dos conclusiones importantes:

“La aparición de herramientas automatizadas para la administración de requerimientos, sirven de apoyo a los procesos de Ingeniería de Software, que se concentran en capturar

requerimientos, administrarlos y producir una especificación de requisitos. Permiten tener un mayor control en proyectos complejos, reducir costos y retrasos en los proyectos, ayudan a determinar la complejidad y los esfuerzos necesarios; sin duda alguna, una gran ayuda para establecer ideas claras de lo que realmente se necesita para llevar a cabo una exitosa Ingeniería de Requerimientos y, por ende, un comienzo prometedor cuando se quiere tener éxito con un proyecto de software.” (Arias, 2006, pp. 10-11).

“Es necesario dar a conocer que alrededor del mundo existen estándares enfocados en el mejoramiento de los procesos de desarrollo de software, citando entre ellos a los estándares propuestos por la IEEE, el SEI (Software Engineering Institute) que propone el modelo de capacidad de madurez, mejor conocido como CMM (The Capability Maturity Model), el PMI (Project Management Institute), que ofrece certificaciones para el área de administración del proyectos y las ya conocidas normas ISO, en cuyas normas también se involucran apartados referentes al desarrollo de software.” (Arias, 2006, pp. 10-11).

A esta mejora constante de los procesos, la utilización y adaptación de estándares internacionales y a la aplicación de un software de gestión en la fase de requerimientos, benefician al equipo de desarrollo en la construcción de un software de calidad, basados en requerimientos funcionales correctamente definidos, evitando así el fracaso del proyecto, reduciendo costos de mantenimiento y garantizan un producto de acorde a las expectativas del cliente.

En el trabajo de los autores (Dávila, Melendez, & Flores) titulado “Determinación de los Requerimientos de Calidad del Producto Software Basados en Normas Internacionales” recalcan la importancia y la influencia que tiene elaborar requerimientos de calidad, de igual manera promueve la aplicación de los estándares como complemento al desarrollar un software con niveles aceptables de funcionalidad para el usuario final.

Factibilidad técnica. - existe el conocimiento necesario y los recursos disponibles para desarrollar el presente proyecto.

Factibilidad Operativa. - Este estudio se lo realizará por medio de un prototipo en donde se gestionará la documentación de cada ítem del estándar aplicado.

Factibilidad Económica. - El financiamiento de estudio será por parte del investigador para realizar la aplicación.

## **1.5. Objetivos de la Investigación**

### **1.5.1. Objetivo General**

- Implementar el software de gestión en la fase de análisis de requerimientos funcionales, para medir el impacto en la mejora del desarrollo de software de la Cooperativa Financiera Atuntaqui.

### **1.5.2. Objetivos Específicos**

- Analizar los procesos y la metodología aplicada a la fase de análisis de requerimientos funcionales.
- Desarrollar el prototipo de gestión para la fase de análisis de requerimientos funcionales.
- Validar los resultados obtenidos con la implementación del software de gestión.

## **1.6. Hipótesis o Proposición**

Un sistema de gestión para la fase de análisis de requerimientos mejora la calidad funcional de los sistemas desarrollados en la Cooperativa Financiera Atuntaqui.

### **1.6.1. Preguntas Directrices**

- ¿Cómo se maneja la fase de análisis de requerimientos funcionales?
- ¿Se utiliza una metodología de desarrollo?
- ¿Cuál es el tiempo promedio asignado para obtener los requerimientos funcionales?
- ¿Cuáles son los roles de los integrantes del equipo de requerimientos funcionales?
- ¿El documento base de análisis de requerimientos funcionales se rige a estándares?

## **1.7. Situación actual de la gestión de requerimientos funcionales**

Esta encuesta se la realizó con el fin de tener una idea clara de cómo se lleva actualmente el proceso para la fase de requerimientos; con los datos tabulados y con los resultados obtenidos podremos proponer las soluciones óptimas para mejorar el procedimiento.

La encuesta se encuentra en el anexo D, misma que se la realizó a los jefes departamentales de la Cooperativa Financiera Atuntaqui, quienes son responsables de solicitar los requerimientos acerca de un nuevo proyecto o la mejora a un sistema que se encuentra en producción.

Con los datos obtenidos se puede realizar las siguientes conclusiones y recomendaciones:

a) Conclusiones

- Los usuarios realizan de manera generalizada y con un bajo nivel de detalle los requisitos.
- La capacitación en temas de mejora a los procesos de desarrollo de software es limitada.
- Existe una gran cantidad de cambios a los requerimientos originales.
- El desconocimiento en temas sobre calidad y gestión de requerimientos se evidencia en los resultados presentados.
- No se aplica normas y estándares internacionales para la gestión de requerimientos.

b) Recomendaciones

- Se debe conformar un equipo de trabajo entre el personal con conocimientos en el tema de gestión de requerimientos para que brinde ayuda al usuario.
- No se debe dejar que el usuario realice los requerimientos sin la ayuda del personal capacitado.
- Se debe capacitar al personal de desarrollo en temas de gestión de procesos.
- Se debe realizar un documento guía basado en algún estándar para gestionar los requerimientos.

## CAPÍTULO II. MARCO REFERENCIAL

### 2.1. Marco Teórico

#### 2.1.1. Requerimientos

Los requisitos del software expresan las necesidades y restricciones impuestas a un producto de software que contribuyen a la solución de un problema del mundo real (Bourque & Fairley, 2014, p. 32).

En un análisis del autor Roger Pressman, reflexiona sobre lo que abarca un requerimiento, para esto realiza una revisión rápida con las siguientes preguntas: ¿Qué es? – Es un listado con los impactos que va a tener el software hacia el giro del negocio de la institución; ¿Quién lo realiza? – Los ingenieros de software, analistas de TI u otros Stakeholders del proyecto (managers, clientes y usuarios finales); ¿Por qué es importante? – Para diseñar y construir un elegante sistema que resuelva un problema. (Pressman & Maxim, 2015, p. 166).

Una vez analizados estas preguntas entonces los requerimientos especifican lo que el sistema debe hacer, las funciones, los servicios que proporciona y las restricciones sobre su funcionamiento. Estos requisitos reflejan las necesidades de los clientes para un sistema que sirve para un determinado propósito (Sommerville, 2011).

#### 2.1.2. Tipos de requerimientos

En la bibliografía de los autores, Ian Sommerville, Roger Pressman y el SWEBOK clasifican a los requerimientos en funcionales y no funcionales.

Los requisitos funcionales describen las funciones que el software debe ejecutar; por ejemplo, formatear un texto o modular una señal. A veces se conocen como capacidades o características (Bourque & Fairley, 2014, p. 34).

Los requisitos no funcionales son los que actúan para restringir la solución. Los requisitos no funcionales a veces se conocen como restricciones o requisitos de calidad. Pueden clasificarse en requisitos de rendimiento, mantenimiento, seguridad, confiabilidad e interoperabilidad (Bourque & Fairley, 2014, p. 34).

Otra clasificación de requerimientos se muestra en la siguiente tabla de acuerdo con el criterio de los autores (Wieggers & Beatty, 2013).

**Tabla 2.** *Tipos de requerimientos*

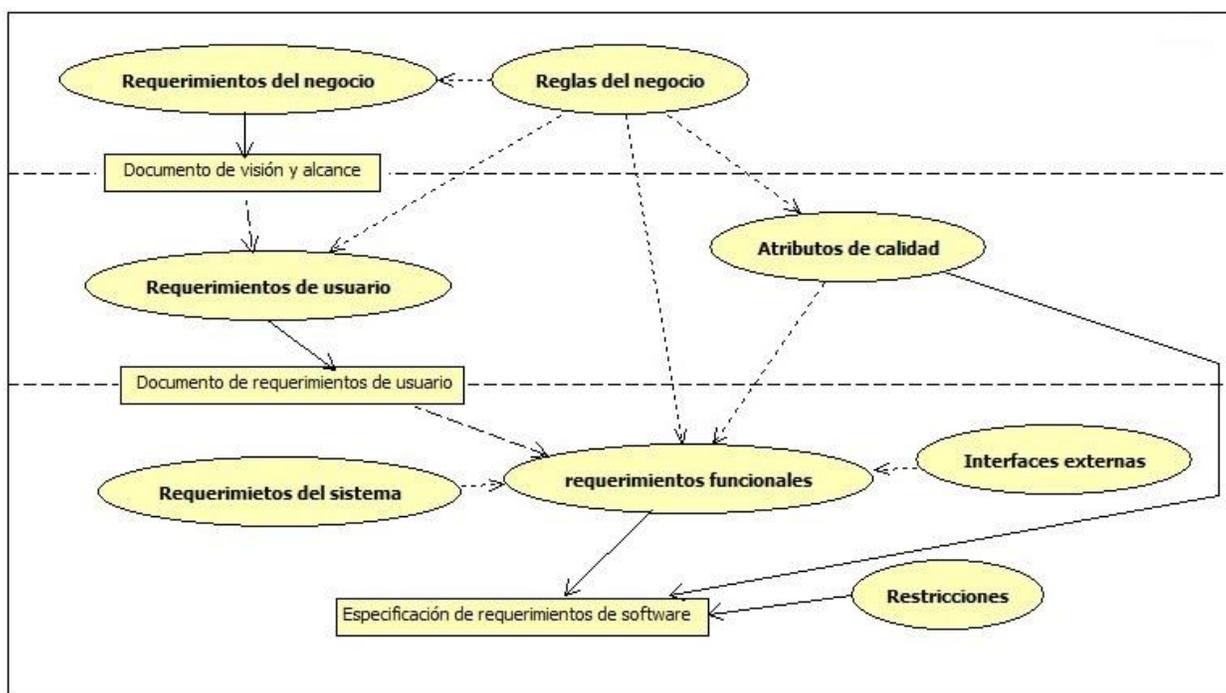
<b>Término</b>	<b>Definición</b>
<b>Requisito de negocio</b>	Un objetivo de negocio de la organización que construye un producto o de un cliente que lo adquiere.
<b>Regla de negocio</b>	Una política, una directriz, un estándar o un reglamento que define algún aspecto del negocio.
<b>Restricción</b>	Una política, una directriz, un estándar o un reglamento que restringe algún aspecto del negocio.
<b>Requisito de interfaz externa</b>	Descripción de una conexión entre un sistema y un usuario, otro sistema o un dispositivo de hardware.
<b>Característica</b>	Una o más capacidades del sistema lógicamente relacionadas que proporcionan valor a un usuario y se describen mediante un conjunto de requisitos funcionales.
<b>Requerimiento funcional</b>	Una descripción de un comportamiento que un sistema exhibirá bajo condiciones específicas.
<b>Requisito no funcional</b>	Una descripción de una propiedad o característica que un sistema debe exhibir o una restricción que se debe respetar.
<b>Atributo de calidad</b>	Un tipo de requisito no funcional que describe un servicio o característica de rendimiento de un producto.
<b>Requisitos del sistema</b>	Un requisito para un producto que contiene varios subsistemas, que podría ser todo software o software y hardware.

<b>Requisitos de usuario</b>	Un objetivo o una tarea que determinadas clases de usuarios puede realizar con un sistema o un atributo de producto deseado.
------------------------------	--

**Fuente:** (Wieggers & Beatty, 2013, p. 40)

Los requisitos del software incluyen tres niveles: requisitos comerciales, del usuario y funcionales. Además, cada sistema tiene una variedad de requisitos no funcionales.

El modelo de la Figura 1 muestra la relación que tiene estos diversos tipos de requisitos (Wieggers & Beatty, 2013).



*Figura 1.* Relaciones entre varios tipos de información de requisitos. Las flechas sólidas significan "se almacenan en"; las flechas punteadas significan "son el origen de" o "influencia". **(Wieggers & Beatty, 2013)**

Los óvalos representan tipos de información de requisitos y los rectángulos indican documentos en los que se debe almacenar esa información. Las flechas sólidas indican que cierto tipo de información se almacena en el documento indicado. Las reglas empresariales y los requisitos del sistema se almacenan aparte de los requisitos de software, como en un catálogo de reglas de negocio o una especificación de requisitos del sistema, respectivamente. Las flechas punteadas indican que un tipo de información es el origen o influye en otro tipo de requisito. Los

requisitos de datos no se muestran explícitamente en este diagrama. Las funciones de manipulación de datos, por lo que los requisitos de datos pueden aparecer en los tres niveles (**Wieggers & Beatty, 2013**).

### 2.1.3. Características de los requerimientos

Los requerimientos permiten a los desarrolladores comprender lo que el cliente necesita para el sistema, también muestra a los diseñadores las funcionalidades y características que va a tener; de igual manera indican al equipo de pruebas qué simulaciones llevar a cabo para convencer al cliente de que el sistema que se le entrega es lo solicitado. Para esto, los requerimientos deben cumplir algunas características para que sean comprensibles y cumplan el objetivo de ayudar a los desarrolladores a crear el sistema de la manera correcta.

Las características de los requerimientos mencionados en el estándar IEEE-830 los explica (Gómez Fuentes, 2011, p. 6) en la siguiente tabla.

**Tabla 3.** *Características de requerimientos Estándar IEEE-830*

<b>Característica</b>	<b>Descripción</b>
<b>Correctos.</b>	Tanto el cliente como el desarrollador deben revisarlos para asegurar que no tienen errores.
<b>Consistentes.</b>	Dos requerimientos son inconsistentes cuando es imposible satisfacerlos simultáneamente.
<b>Completo.</b>	El conjunto de requerimientos está completo si todos los estados posibles, cambios de estado, entradas, productos y restricciones están descritos en alguno de los requerimientos.
<b>Realistas.</b>	Todos los requerimientos deben ser revisados para asegurar que son posibles y que inciden directamente en la resolución del problema del cliente.
<b>Verificables.</b>	Se deben poder preparar pruebas que demuestren que se han cumplido los requerimientos.
<b>Rastreables.</b>	¿Se puede rastrear cada función del sistema hasta el conjunto de requerimientos que la establece?

**Fuente:** (Gómez Fuentes, 2011, p. 6)

### 2.1.4. Técnicas de recopilación de requerimientos

La recopilación de requerimientos es un paso importante para poder determinar necesidades viables para resolver un problema. Un error o mala interpretación de un requisito en esta etapa propagará el problema a través del ciclo de vida de desarrollo (Ibañez, 2010).

Además de afectar en costos y tiempos, el retraso para otros proyectos significará un impacto negativo para la empresa por lo que a continuación se tiene las siguientes técnicas, las cuales debe realizar un experto en el área:

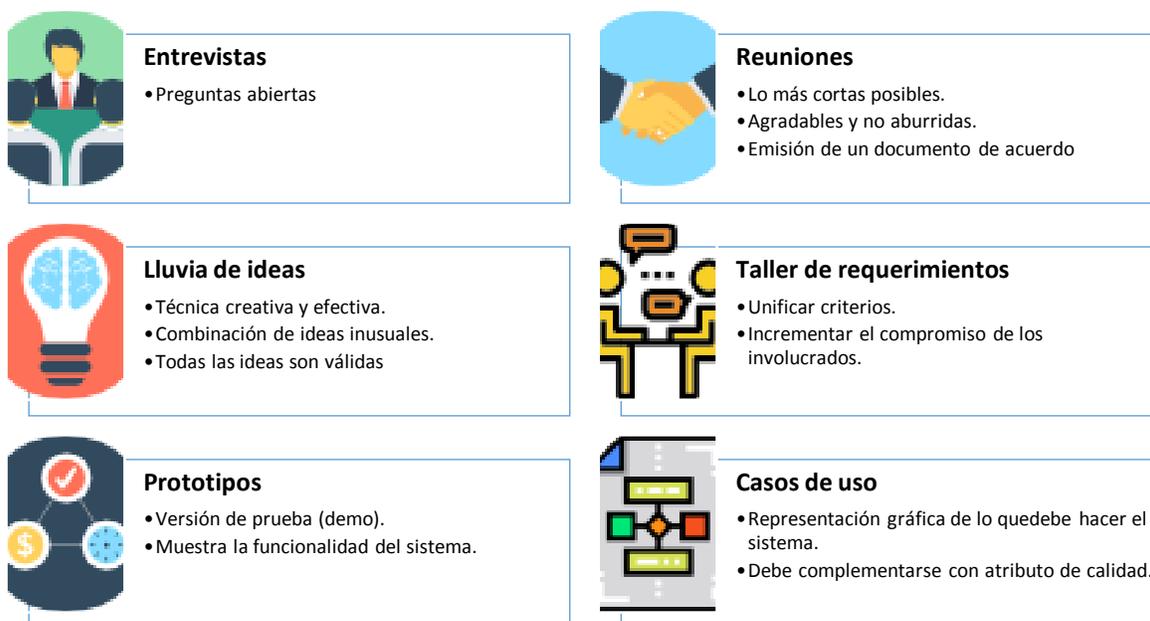


Figura 2. Técnicas de gestión de requerimientos (Ibañez, 2010).

La versión del SWEBOK (Software Engineering Body of Knowledge) detalla las siguientes técnicas que se asemejan a la figura anteriormente descrita (Bourque & Fairley, 2014, p. 38).

- Entrevistas. Entrevistar a los interesados es un medio tradicional de obtener los requisitos; es importante comprender las ventajas y limitaciones de las entrevistas y de qué manera se la debe realizar.
- Escenarios. Los escenarios proporcionan un medio valioso para proporcionar contexto a la obtención de los requisitos del usuario. Permiten que el ingeniero de

software proporcione un esquema para las preguntas sobre las tareas del usuario. El tipo de escenario común es la descripción del caso de uso.

- **Prototipos.** Esta técnica es una importante herramienta para aclarar requisitos ambiguos. Pueden actuar de manera similar a los escenarios, proporcionando a los usuarios un contexto en el que puedan comprender mejor la información que necesitan proporcionar. Existe una amplia gama de técnicas de creación de prototipos, desde maquetas de papel de diseños de pantallas hasta versiones beta de productos de software.
- **Reuniones.** El propósito de estas reuniones es lograr un efecto positivo mediante el cual un grupo de personas puede aportar con información sobre sus requisitos de software. Pueden hacer una lluvia de ideas y refinarlas acorde a la complejidad del requerimiento. Las reuniones tienen que ser manejadas cuidadosamente con la ayuda de un facilitador para prevenir cualquier inconveniente entre los participantes.
- **Observación.** Con el uso de esta técnica, los ingenieros de software aprenden sobre las tareas del usuario observando cómo realizan sus tareas interactuando entre sí y con herramientas de software u otros recursos. Estas técnicas tienen sus ventajas porque sirven para ilustrar varias tareas de usuario con los procesos de negocio de la empresa para su rápida comprensión.
- **Historias de usuarios.** Esta técnica se usa en metodologías de desarrollo Ágiles y se refiere a descripciones cortas de la funcionalidad requerida expresada en términos de cliente. El objetivo es evitar que un requerimiento se convierta en no válido antes de empezar con el desarrollo. Antes de implementar una historia de usuario, el cliente debe escribir un procedimiento de aceptación adecuado para determinar si se han cumplido los objetivos de la historia de usuario.
- **Otras técnicas.** Existe una serie de otras técnicas para apoyar la obtención de información de requerimientos y van desde el análisis de productos de competidores hasta la aplicación de técnicas de minería de datos a fuentes de conocimiento de dominio o bases de datos de solicitudes de clientes.

Si bien estas técnicas permiten tener un mejor acercamiento con el usuario final y tratar de comprender sus necesidades, es una tarea difícil para el ingeniero de software aplicar cualquiera de las técnicas antes mencionadas porque los usuarios pueden tener dificultades para describir sus requisitos, pueden dejar información importante no declarada o no cooperar en su totalidad.

#### **2.1.5. Requerimientos funcionales**

Para tener un concepto adecuado referente a los requerimientos funcionales, los siguientes autores lo definen como:

*“Describen la interacción entre el sistema y su ambiente independientemente de su implementación. El ambiente incluye al usuario y cualquier otro sistema externo que interactúa con el sistema”* (Quiroga, 2015, p. 3).

*“Los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular y cómo se debe comportar ante situaciones particulares”* (Laguna, 2014, p. 11).

En las definiciones de los autores coinciden en que el sistema a desarrollar debe contemplar todas las necesidades principales solicitadas por el cliente para un correcto funcionamiento.

#### **2.1.6. Ingeniería de requerimientos**

El concepto del diccionario sobre ingeniería menciona al conjunto de conocimientos científicos y tecnológicos para la innovación, invención, desarrollo y mejora de técnicas y herramientas para satisfacer las necesidades y resolver los problemas de las empresas y la sociedad. Entonces la ingeniería de requerimientos pone en práctica conocimiento científico a los procesos de recopilar, analizar y verificar las necesidades del cliente o usuario de manera correcta y completa para desarrollar un sistema.

Otros conceptos de ingeniería de requerimientos son de los autores Roger Pressman e Ian Sommerville que señalan a la ingeniería de requerimientos como:

*“La ingeniería de requerimientos es el proceso de entender y definir qué servicios son requeridos del sistema e identificar las restricciones en el funcionamiento y desarrollo del sistema.”* (Sommerville, 2011, p. 36).

*“La ingeniería de requisitos establece una base sólida para el diseño y la construcción. Sin él, el software resultante tiene una alta probabilidad de no satisfacer las necesidades del cliente”* (Pressman & Maxim, 2015, p. 133).

La ingeniería de requisitos abarca siete tareas distintas: inicio, elicitación, elaboración, negociación, especificación, validación y gestión. Es importante tener en cuenta que algunas de estas tareas ocurren en paralelo y todas están adaptadas a las necesidades del proyecto (Pressman & Maxim, 2015, p. 133). A continuación, se detalla el contenido de las tareas:

- a) Inicio. – En esta tarea se establece una comprensión básica del problema y se analiza la viabilidad del proyecto en donde, los Stakeholders como los gerentes comerciales, personas de mercadotecnia, gerentes de productos definen un caso comercial, su alcance y el beneficio que tendrá la empresa con su implementación.
- b) Elicitación. – Esta tarea permite realizar preguntas al usuario o cliente sobre las funcionalidades que va a tener el sistema, suena una tarea simple, pero es una tarea difícil comprender lo que el cliente necesita. Por lo tanto, en esta tarea se establecen objetivos de negocio claros y todos los involucrados en el proyecto deben compartir sus ideas honestamente.
- c) Elaboración. – Una vez obtenidos los requerimientos del cliente, se identifican varios aspectos de la función, el comportamiento y la información del software, además se establecen escenarios de cómo los usuarios y otros actores involucrados interactúan con el sistema.
- d) Negociación. – Como los clientes se acostumbran a solicitar mayor cantidad de requerimientos, es en esta tarea que se debe llegar a un acuerdo entre todos los involucrados del proyecto, una buena opción es la que menciona Pressman acerca de realizar una clasificación de los requerimientos, priorizarlos y analizar el costo de implementar cada uno de ellos, revisar el riesgo que podrían tener y de esta manera se podría eliminarlos, agruparlos para que cada parte logre un cierto nivel de satisfacción.
- e) Especificación. – Una especificación puede ser un documento escrito, un conjunto de modelos gráficos, un modelo matemático formal, una colección de escenarios de

uso, un prototipo o cualquier combinación de estos. Se sugiere realizar un documento estándar de acuerdo dependiendo del tamaño y la complejidad del software a ser desarrollado.

- f) Validación. – La validación de los requisitos examina la especificación para garantizar que todos los requisitos del software se hayan establecido sin ambigüedades; que se han detectado y corregido inconsistencias, omisiones y errores; y que los productos del trabajo se ajustan a los estándares establecidos para el proceso, el proyecto y el producto.
- g) Gestión. – La gestión de requisitos es un conjunto de actividades que ayudan al equipo del proyecto a identificar, controlar y rastrear los requisitos y los cambios en los requisitos en cualquier momento a medida que avanza el proyecto. Se debe realizar un seguimiento de los requisitos individuales y mantener vínculos entre los requisitos dependientes para que pueda evaluar el impacto de los cambios en los requisitos. Se debe establecer un proceso formal para hacer propuestas de cambio y vincularlas a los requisitos del sistema. El proceso formal de gestión de requisitos debe comenzar tan pronto como esté disponible una versión preliminar del documento de requisitos. Sin embargo, debe comenzar a planificar cómo administrar los requisitos cambiantes durante el proceso de obtención de requisitos (Sommerville, 2011, p. 112).

### **2.1.7. Importancia de la ingeniería de requerimientos**

La autora (Herrera, 2003, p. 3) en su documento de la Ingeniería de Requerimientos, menciona los principales beneficios de la ingeniería de requerimientos (Herrera, 2003: 3):

- Permite gestionar las necesidades del proyecto en forma estructurada.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados.
- Disminuye los costos y retrasos del proyecto.
- Mejora la calidad del software.
- Mejora la comunicación entre equipos.
- Evita rechazos de usuarios finales.

### 2.1.8. Estándar IEEE 830-1998

El estándar IEEE 830-1998 describe los enfoques recomendados para la especificación de requerimientos de software (ERS<sup>1</sup>). Describe el contenido y las cualidades de una buena ERS, por lo tanto, se utiliza como base para mejorar la trazabilidad de los requisitos en el mismo.

Este documento contiene un formato de lo que debe contener un buen documento de ERS, además, presenta una plantilla que se la puede adaptar según las necesidades del negocio.

A continuación, se muestra el esquema de trabajo del estándar IEEE-830:

<b>1</b>	<b>Introducción</b>
1.1	Propósito
1.2	Ámbito del Sistema
1.3	Definiciones, Acrónimos y Abreviaturas
1.4	Referencias
1.5	Visión general del documento
<b>2</b>	<b>Descripción General</b>
2.1	Perspectiva del Producto
2.2	Funciones del Producto
2.3	Características de los usuarios
2.4	Restricciones
2.5	Suposiciones y Dependencias
<b>3</b>	<b>Requisitos Específicos</b>
3.1	Interfaces Externas
3.2	Funciones
3.3	Requisitos de Rendimiento
3.4	Restricciones de Diseño
3.5	Atributos del Sistema
3.6	Otros Requisitos
<b>4</b>	<b>Apéndices</b>
<b>5</b>	<b>Índice</b>

Figura 3. Esquema de Especificación de Requerimientos (IEEE 830, 1998).

A continuación, se describe brevemente cada uno de los enunciados que se definen en el estándar mencionado.

---

<sup>1</sup> Especificación de Requerimientos de Software

## 1. Introducción

En esta sección se proporciona una introducción a todo el documento de Especificación de Requisitos de Software.

### 1.1. Propósito

Se define el propósito del documento de ERS y se especifica a quién va dirigido el documento.

### 1.2. Ámbito del Sistema

En esta subsección se registra el nombre al futuro sistema, se explica lo que el sistema va a realizar y las limitaciones, se describen los beneficios, objetivos y metas que se espera alcanzar.

### 1.3. Definiciones, Acrónimos y Abreviaturas.

Se definen todos los términos, acrónimos y abreviaturas utilizadas en el desarrollo del documento.

### 1.4. Referencias

Se debe presentar una lista completa de todos los documentos referenciados.

### 1.5. Visión General del Producto

En esta subsección se describe brevemente los contenidos y la organización del resto del documento.

## 2. Descripción General

En esta sección se describen todos los factores que afectan al producto y a sus requerimientos, no se los describen, solo es un contexto.

### 2.1. Perspectiva del Producto

Esta subsección debe relacionar el futuro del sistema con otros productos.

#### 2.1.1. Indicar si es un producto independiente o parte de un sistema mayor

#### 2.1.2. Interfaces de sistema

#### 2.1.3. Interfaces de usuario

##### 2.1.3.1. Características lógicas del interfaz

##### 2.1.3.2. Cuestiones de optimización del interfaz de usuario

#### 2.1.4. Interfaces hardware

#### 2.1.5. Interfaces software

2.1.5.1. Descripción del producto software utilizado

2.1.5.2. Propósito del interfaz

2.1.5.3. Definición del interfaz: contenido y formato

2.1.6. Interfaces de comunicaciones

2.1.7. Limitaciones de memoria

2.1.8. Operaciones

2.1.8.1. Modos de operación de los distintos grupos de usuarios

2.1.8.2. Periodos de operaciones interactivas y automáticas

2.1.8.3. Funciones respaldo del procesamiento de datos

2.1.8.4. Operaciones de backup y recuperación

2.1.9. Requerimientos para adaptarse a la ubicación

2.1.9.1. Indicar cualquier dato o secuencia de inicialización específico de cualquier lugar, modo de operación.

2.1.9.2. Características que deben ser modificadas para una instalación en particular.

2.2. Funciones del Producto

Esta subsección debe proporcionar un resumen de las funciones principales que el software debe llevar a cabo. Las funciones deben estar organizadas de manera que el cliente o cualquier otra persona lo comprendan perfectamente. Para ello se pueden utilizar métodos textuales o gráficos, siempre que dichos gráficos reflejen las relaciones entre funciones y no el diseño del sistema.

En la metodología estructurada se podrían utilizar el diagrama de flujo de datos y en una metodología orientada a objetos, se modelarían a través de los casos de uso.

2.3. Características de los usuarios

Se indica el tipo de usuario al que se dirige la aplicación, así como su experiencia técnica y nivel de conocimientos.

2.4. Restricciones

Se debe indicar cualquier tipo de limitación como pueden ser políticas de la empresa, limitaciones hardware, seguridad, protocolos de comunicación, interfaces con otras aplicaciones o estándares de la empresa en cuanto a interfaces. Serán las limitaciones que se imponen sobre los desarrolladores del producto.

### 2.5. Suposiciones y Dependencias

En este apartado aparecerá cualquier factor, que si cambia puede afectar a los requerimientos. No son restricciones de diseño, por ejemplo, asumir que un determinado sistema operativo estará disponible, presuponer una cierta organización de las unidades de la empresa. Si cambian ciertos detalles puede ser necesario revisar los requisitos.

## 3. Requisitos Específicos

Esta sección de la especificación de requisitos software contiene todos los requerimientos hasta un nivel de detalle suficiente para permitir a los diseñadores diseñar un sistema que satisfaga dichos requisitos, y que permita diseñar las pruebas que ratifiquen que el sistema cumple con las necesidades requeridas.

Los requisitos que se indiquen deben describir comportamientos externos del sistema, observables por el usuario, así como por parte de los operadores y otros sistemas. Los requerimientos deben ser no ambiguos, completos, consistentes, clasificados, verificables, modificables, explorables y fáciles de mantener.

### 3.1. Interfaces Externas

En esta subsección se definirán los requisitos que afecten a la interfaz de usuario e interfaz con otros sistemas (hardware y software), así como a interfaces de comunicaciones.

### 3.2. Funciones

En esta subsección se deben especificar todas aquellas acciones o funciones que debe llevar a cabo el sistema a desarrollar. Las acciones que se indican como “el sistema deberá” son las que se deben incluir en este apartado.

### 3.3. Requisitos de Rendimiento

En esta subsección se incluyen los requisitos relacionados con la carga que se espera que tenga que soportar el sistema (número de usuarios simultáneos, número de terminales). También se pueden incluir los requisitos que afecten a la información que se vaya a guardar en la base de datos (cantidad de registros en una base de datos, frecuencia de uso).

### 3.4. Restricciones de Diseño

Se incluyen todas las restricciones que afecten al diseño de la aplicación, como pueden ser estándares internos de la organización o limitaciones hardware.

### 3.5. Atributos del Sistema

Se detallarán atributos como la fiabilidad, mantenibilidad, seguridad, mecanismos de acceso restringido (contraseña), usuarios autorizados a realizar ciertas tareas críticas.

### 3.6. Otros requisitos

Aquellos requerimientos que no se hayan podido incluir en ninguna de las secciones anteriormente especificadas.

## 4. Apéndices

Se incluirá cualquier tipo de información relacionada con la ERS, pero que no forme parte de esta. Por ejemplo, se incluirían los resultados del análisis de costos, restricciones especiales acerca del lenguaje de programación.

## 5. Índice

Se proporciona un índice para poder tener un acceso rápido a la documentación presentada en la ERS.

### **2.1.9. Estándar ISO/IEC/IEEE 29148-2011**

Este estándar contiene disposiciones para los procesos y productos relacionados con la ingeniería de requisitos para sistemas y productos de software y servicios a lo largo del ciclo de vida. Define la construcción de un buen requisito, proporciona atributos y características de los

requisitos, y discute la aplicación iterativa y recursiva de los procesos de requisitos a lo largo del ciclo de vida.

Proporciona una guía adicional en la aplicación de la ingeniería de requisitos y procesos de gestión para las actividades relacionadas con los requisitos. Se definen los ítems de información aplicables a la ingeniería de requisitos y su contenido. El contenido de este estándar puede añadirse al conjunto existente de procesos de ciclo de vida relacionados con los requisitos definidos por ISO / IEC 12207 o ISO / IEC 15288, o puede utilizarse independientemente.

Esta Norma Internacional especifica los procesos requeridos que se van a implementar para la ingeniería de requisitos para sistemas y productos de software (incluyendo servicios) a lo largo de todo el ciclo de vida, da pautas para aplicar los requisitos y requisitos relacionados con procesos descritos en ISO / IEC 12207, especifica los elementos de información requeridos que se van a producir a través de la implementación de los procesos de requisitos, especifica el contenido requerido de los elementos de información requeridos y da pautas para el formato de los ítems de información requeridos y relacionados.

Esta Norma Internacional proporciona un tratamiento unificado de los procesos y productos involucrados en los requisitos de ingeniería a lo largo del ciclo de vida de sistemas y software; es el resultado de la armonización de las siguientes fuentes:

- ISO/IEC 12207:2008 (IEEE Std 12207-2008), Systems and software engineering — Software life cycle processes.
- ISO/IEC 15288:2008 (IEEE Std 15288-2008), Systems and software engineering — System life cycle processes.
- ISO/IEC/IEEE 15289:2011, Systems and software engineering — Content of life-cycle information products (documentation).
- ISO/IEC TR 19759, Software Engineering — Guide to the Software Engineering Body of Knowledge (SWEBOK).
- IEEE Std 830, IEEE Recommended Practice for Software Requirements Specifications.

- IEEE Std 1233, IEEE Guide for Developing System Requirements Specifications.
- IEEE Std 1362, IEEE Guide for Information Technology — System Definition — Concept of Operations (ConOps) Document.
- ISO/IEC TR 24748-1, Systems and software engineering — Life cycle management — Part 1: Guide for life cycle management.
- ISO/IEC/IEEE 24765, Systems and software engineering — Vocabulary.

El estándar propone elaborar tres documentos:

- ✓ Stakeholder requirements specification document (StRS). Documento de especificación de requisitos de las partes interesadas (ERI).
- ✓ System requirements specification document (SyRS). Documento de especificación de requisitos del sistema (ERSis).
- ✓ Software requirements specification document (SRS), Documento de especificación de requisitos de software (ERS).

La siguiente figura, muestra la estructura y los ítems del estándar ISO/IEC/IEEE 29148 para el documento de especificación de requisitos de software, el cual es nuestro objeto de estudio:

- 1. Propósito**
- 2. Alcance**
- 3. Perspectiva del producto**
  - 3.1 Interfaces del sistema
  - 3.2 Interfaces de usuario
  - 3.3 Interfaces de Hardware
  - 3.4 Interfaces de Software
  - 3.5 Interfaces de Comunicación
  - 3.6 Restricciones de Memoria
  - 3.7 Operaciones
  - 3.8 Requerimientos de adaptación del sitio
- 4. Funciones del producto**
- 5. Características de usuario**
- 6. Limitaciones**
- 7. Suposiciones y dependencias**
- 8. Aportes de los requerimientos**
- 9. Requerimientos específicos**
- 10. Interfaces externas**
- 11. Funciones**
- 12. Requerimientos de usabilidad**
- 13. Requerimientos de rendimiento**
- 14. Requerimientos de base de datos lógicas**
- 15. Diseño de restricciones**
- 16. Cumplimiento de estándares**
- 17. Atributos del sistema**
- 18. Verificación**
- 19. Información de soporte**

*Figura 4.* Esquema de Especificación de Requerimientos. (ISO/IEC/IEEE 29148, 2011)

A continuación, se detalla brevemente el contenido del esquema de la figura 2-4.

1. Propósito. - Delinear el propósito del software a ser desarrollado.
2. Alcance. - Describir el alcance del software bajo las siguientes consideraciones:
  - a) Identificar los productos de software que se van a producir por nombre por ejemplo Generador de Reportes.
  - b) Explicar qué funcionalidad tiene el producto.
  - c) Describir la aplicación del software incluyendo los beneficios relevantes, objetivos y metas.

### 3. Descripción del producto

Definir la relación del sistema con otros productos relacionados.

#### 3.1. Interfaces del sistema

Se enumera cada interfaz del sistema e identifica la funcionalidad del software para lograr que el requisito y la descripción de la interfaz coincidan con el sistema.

#### 3.2. Interfaces de usuario

Se debe especificar lo siguiente:

- a) Las características lógicas de cada interfaz entre el producto de software y sus usuarios. Esto incluye aquellas características de configuración, por ejemplo, formatos de pantalla requeridos, diseños de páginas o ventanas, contenido de cualquier informe o menú, disponibilidad de teclas de función programables.
- b) Todos los aspectos de la optimización de la interfaz con la persona que utiliza o proporciona otro tipo de apoyo al sistema. Esto simplemente puede comprender una lista de cosas que hacer y qué no hacer en cómo el sistema aparecerá al usuario. Un ejemplo puede ser un requisito para la opción de mensajes de error largos o cortos.

#### 3.3. Interfaces de hardware

Especificar las características lógicas de cada interfaz entre el producto de software y los elementos de hardware del sistema. Esto incluye características de configuración (número de puertos, conjuntos de instrucciones, dispositivos a utilizar).

#### 3.4. Interfaces de software

Especificar el uso de otros productos de software requeridos (por ejemplo, un sistema de gestión de datos, un sistema operativo, o un paquete matemático), o que interactúa con otros sistemas de aplicación (por ejemplo, el enlace entre un sistema de cuentas por cobrar y un sistema de contabilidad general).

Para cada producto de software requerido, se debe especificar:

- a) Un nombre.
- b) Mnemónica.
- c) Número de especificación.
- d) Número de versión.
- e) Fuente.

Para cada interfaz se debe especificar lo siguiente:

- a) Describir el propósito de cada interface relacionado con este producto de software.
- b) Definir a cada interfaz en términos de contenido y formato de mensaje. No es necesario detallar ninguna interfaz bien documentada, pero se requiere una referencia al documento que define la interfaz.

### 3.5. Interfaces de comunicaciones

Especificar las distintas interfaces a las comunicaciones, como los protocolos de red local.

### 3.6. Limitaciones de memoria

Especificar las características y límites aplicables a la memoria primaria y secundaria.

### 3.7. Operaciones

Especificar las operaciones normales y especiales requeridas por el usuario, como:

- a) Los diversos modos de operaciones en la organización del usuario (por ejemplo, operaciones iniciadas por el usuario).
- b) Períodos de operaciones interactivas y períodos de operaciones desatendidas.
- c) Funciones de apoyo al procesamiento de datos.
- d) Operaciones de copia de seguridad y recuperación.

### 3.8. Requisitos de adaptación del sitio

- a) Definición de los requisitos para cualquier secuencia de datos o inicialización específica de un sitio, una misión o un modo operativo (por ejemplo, valores de cuadrícula, límites de seguridad).
- b) Especificación del sitio o de las características relacionadas con la misión que se deben modificar para adaptar el software a una instalación particular.

#### 4. Funciones del producto

Se debe proporcionar un resumen de las principales funciones que el software realizará. Por ejemplo, una ERS para un programa de contabilidad puede usar esta parte para tratar el mantenimiento de cuentas de clientes, la declaración de clientes y la preparación de facturas sin mencionar la gran cantidad de detalle que cada una de esas funciones requiere.

Se debe tener en cuenta lo siguiente:

- a) Las funciones del producto deben organizarse de manera que la lista de funciones sea comprensible para el usuario o para cualquier persona que lea el documento por primera vez.
- b) Los métodos textuales o gráficos pueden usarse para mostrar las diferentes funciones y sus relaciones. El diagrama no pretende mostrar un diseño de un producto, sino que simplemente muestra las relaciones lógicas entre variables.

#### 5. Características del usuario

Describir las características generales de los grupos de usuarios del producto, incluidas las características que pueden influir en la usabilidad, como el nivel educativo, la experiencia, las discapacidades y los conocimientos técnicos. Esta descripción no debe especificar requisitos específicos, sino que debe indicar las razones por las cuales ciertos requisitos específicos se especifican.

#### 6. Limitaciones

Proporcionar una descripción general de cualquier otro elemento que limite las opciones del proveedor, incluyendo:

- a) Políticas reguladoras.
- b) Limitaciones de hardware.
- c) Interfaces con otras aplicaciones.
- d) Operación en paralelo.
- e) Funciones de auditoría.
- f) Funciones de control.
- g) Requisitos lingüísticos de orden superior.
- h) Protocolos de enlace de señales.
- i) Requisitos de calidad (por ejemplo, fiabilidad).
- j) Criticidad de la solicitud.
- k) Consideraciones de seguridad.
- l) Consideraciones físicas / mentales.

#### 7. Suposiciones y dependencias

Enumerar cada uno de los factores que afectan los requisitos establecidos en la ERS. Estos factores no son restricciones de diseño en el software, pero cualquier cambio en estos factores puede afectar los requisitos en la ERS. Por ejemplo, se puede suponer que un sistema operativo específico estará disponible en el hardware designado para el software

#### 8. Distribución de los requisitos

Asignar los requisitos de software a los elementos de software. Para requisitos que requieran la implementación sobre múltiples elementos de software, o cuando la asignación a un elemento de software es inicialmente indefinida, esto debe ser dicho. Se utilizará una tabla de referencia cruzada por función y elemento de software para resumir los prorrateos.

#### 9. Requisitos específicos

Especificar todos los requisitos de software a un nivel de detalle suficiente para permitir:

- a) A los diseñadores diseñar un sistema para satisfacer esos requisitos.
- b) Los responsables de realizar los test puedan probar que el sistema satisface dichos requisitos.
- c) Describir cada entrada en el sistema, cada salida (respuesta) del sistema y todas las funciones realizadas por el sistema en respuesta a una entrada o en apoyo de una salida.

Los requisitos específicos deben:

- a) Ser declarados de conformidad con todas las características descritas en esta norma Internacional.
- b) Con referencia cruzada a documentos anteriores que se relacionan.
- c) Únicamente identificables.

#### 10. Interfaces externos

Se debe definir todas las entradas y salidas del sistema. La descripción debe complementar las descripciones de la interfaz y no debe repetir la información allí señalada.

Cada interfaz definida debe incluir el siguiente contenido:

- a) Nombre del artículo.
- b) Descripción del propósito.
- c) Fuente de entrada o destino de la producción.
- d) Rango, exactitud y / o tolerancia válidos.
- e) Unidades de medida.
- f) Tiempo.
- g) Relaciones con otros insumos / productos.
- h) Formatos / organización de la pantalla.
- i) Formatos / organización de la ventana.

- j) Formatos de datos.
- k) Formatos de comandos.
- l) Mensaje final.

## 11. Funciones

Se debe definir las acciones fundamentales que deben llevarse a cabo en el software para aceptar y procesar las entradas y procesar y generar las salidas, incluyendo:

- a) Comprobaciones de validez de los insumos.
- b) Secuencia exacta de operaciones.
- c) Respuestas a situaciones anormales, incluyendo:
- d) Desbordamiento.
- e) Instalaciones de comunicación.
- f) Manejo y recuperación de errores.
- g) Efecto de los parámetros.
- h) Relación entre los productos y los insumos:
- i) Secuencias de entrada / salida.
- j) Fórmulas para la conversión de entrada a salida.

Puede ser apropiado dividir los requisitos funcionales en subfunciones o subprocesos. Esto no implica que el diseño del software también debe ser particionado de esa manera.

## 12. Requerimientos de usabilidad

Definir los requisitos de usabilidad (calidad en el uso). Los requisitos y objetivos de usabilidad para el sistema incluyen la efectividad mensurable, la eficiencia y los criterios de satisfacción en contextos específicos de uso.

## 13. Requisitos de rendimiento

Especifique los requisitos numéricos estáticos y dinámicos colocados en el software o en la interacción humana con el software como un todo.

Los requisitos numéricos estáticos pueden incluir lo siguiente:

- a) El número de terminales a ser soportados.
- b) El número de usuarios simultáneos a ser apoyados.
- c) Cantidad y tipo de información a manejar.

#### 14. Requerimientos de la base de datos lógica

Especifique los requisitos lógicos para cualquier información que se va a colocar en una base de datos, incluyendo:

- a) Tipos de información utilizados por diversas funciones.
- b) Frecuencia de uso.
- c) Acceso a las capacidades.
- d) Entidades de datos y sus relaciones.
- e) Limitaciones de integridad.
- f) Requisitos de retención de datos.

#### 15. Restricciones de diseño

Especifique restricciones en el diseño del sistema impuestas por normas externas, requisitos reglamentarios o limitaciones del proyecto.

#### 16. Cumplimiento de los estándares

Especifique los requisitos derivados de normas o regulaciones existentes, incluyendo:

- a) Formato del informe.
- b) Nomenclatura de datos.
- c) Procedimientos contables.
- d) Seguimiento de auditoría.

Se podría especificar el requisito de que el software rastree la actividad de procesamiento. Dichas trazas son necesarias para que algunas aplicaciones cumplan con las normas mínimas reglamentarias o financieras.

#### 17. Atributos del sistema

Especifique los atributos requeridos del producto de software. La siguiente es una lista parcial de ejemplos:

- a) **Fiabilidad:** especificar los factores necesarios para establecer la fiabilidad requerida del sistema al momento de la entrega.
- b) **Disponibilidad:** especificar los factores necesarios para garantizar un nivel de disponibilidad definido para todo el sistema, como punto de control, recuperación y reinicio.
- c) **Seguridad:** especificar los requisitos para proteger el software de acceso accidental o malicioso, modificación de uso, destrucción o divulgación.
  - 1. Utilizar ciertas técnicas criptográficas.
  - 2. Mantener el conjunto de datos de registro o historial específicos.
  - 3. Asignar ciertas funciones a diferentes módulos.
  - 4. Restringir las comunicaciones entre algunas áreas del programa.
  - 5. Comprobar la integridad de los datos para las variables críticas.
  - 6. Asegurar la privacidad de los datos.
- d) **Mantenimiento:** especificar los atributos del software que se relacionan con la facilidad de mantenimiento del propio software.
- e) **Portabilidad:** especificar los atributos del software que se relacionan con la facilidad de portar el software a otras máquinas host y / o sistemas operativos, incluyendo:
  - 1. Porcentaje de elementos con código dependiente del host.
  - 2. Porcentaje de código dependiente del host.
  - 3. Uso de un lenguaje portátil probado.
  - 4. Uso de un subconjunto particular de compilador o lenguaje.
  - 5. Uso de un sistema operativo particular.

## 18. Verificación

Proporcionar los métodos de verificación planificados para calificar el software.

## 19. Información de apoyo

La ERS debe contener información de apoyo adicional incluyendo:

- a) Muestra de formatos de entrada / salida, descripciones de estudios de análisis de costos o resultados de encuestas de usuarios.
- b) Información de apoyo o de antecedentes que pueden ayudar a los lectores de la ERS.
- c) Una descripción de los problemas a resolver por el software.
- d) Instrucciones especiales de embalaje para el código y los medios de comunicación para cumplir con la seguridad, exportación, carga inicial u otros requisitos.

### **2.1.10. Software de gestión**

El software de administración es una frase general utilizada para describir una categoría de software de computadora diseñada para ayudar a simplificar la complejidad de grandes proyectos y tareas, así como para facilitar la colaboración del equipo y la presentación de informes de proyectos. La mayoría de las soluciones de software de gestión también pueden manejar la gestión de recursos y empleados, coordinación de horarios, asignación de tareas, presupuesto, análisis de tiempo y riesgos, y más (Beal, 2016).

El software de gestión es un término amplio que también se puede aplicar al software de gestión financiera, software de gestión de red, software de gestión de relaciones con el cliente, software de gestión de activos o software de gestión de inventario (Beal, 2016).

### **2.1.11. Metodología SCRUM**

Scrum es un marco de trabajo en el que equipos de desarrollo pequeños pueden crear productos o desarrollar proyectos de una forma iterativa e incremental (Deemer, Benefield, Larman, & Vodde, 2017, p. 3).

El desarrollo se estructura en ciclos de trabajo llamados Sprints (también conocidos como iteraciones). Estas iteraciones no deben durar más de cuatro semanas cada una (siendo dos semanas la duración habitual) y tienen lugar una tras otra sin pausa entre ellas.

# SCRUM

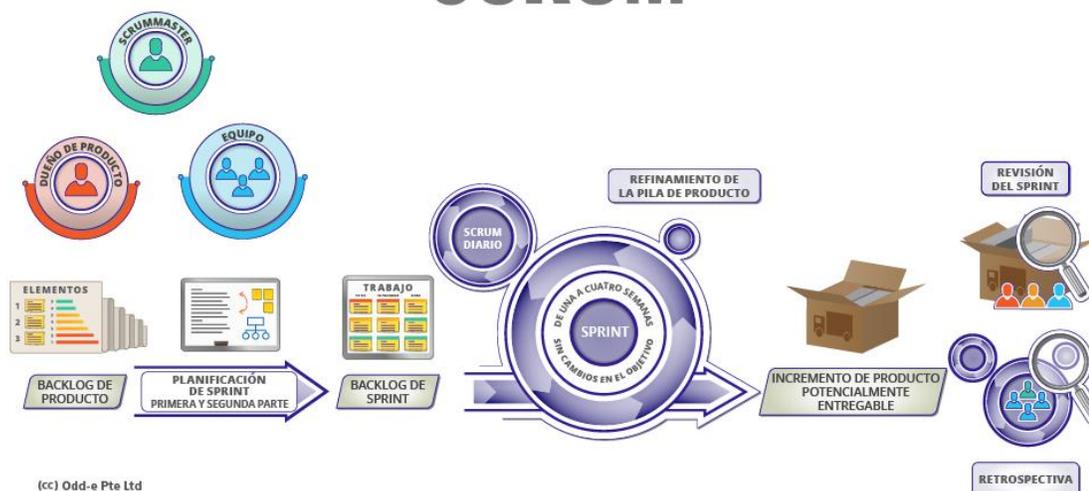


Figura 5 Visión General de SCRUM

Fuente: (The Scrum Primer, 2017)

El Backlog del producto existe y evoluciona durante toda la vida del producto; sirve de guía para la culminación de este. En cualquier momento, el Backlog del producto es la visión única y definitiva de “todo lo que podría ser realizado en algún momento por el equipo, en orden de prioridad”. Sólo existe un Backlog de producto para cada proyecto, lo cual significa que el dueño del producto debe tomar decisiones de prioridad sobre todas las tareas disponibles, en representación de los intereses de todos los Stakeholders.

El resultado de cada sprint se denomina un incremento del entregable del producto. Antes de comenzar el primer sprint, el dueño del producto, el equipo de desarrollo y el Scrum Máster deben revisar todo lo necesario para que un elemento del Backlog se pueda finalizar en el tiempo estipulado y no afecte al entregable del proyecto.

## 2.2. Marco Legal

Según el artículo 6 del Acuerdo No. 119 de la constitución del Ecuador se menciona el uso del software libre en las empresas públicas, esto no quiere decir que las empresas privadas no

pueden hacer uso de software libre; lo que busca este artículo es el fomento al software libre y su aplicación y utilización en las áreas de tecnologías de la información.

En la Cooperativa Financiera Atuntaqui no existe una normativa que especifique el uso de un cierto tipo de herramientas, pero en la institución se manejan con los dos tipos de herramientas, es decir, se desarrollan proyectos utilizando los IDE de desarrollo de Microsoft y se desarrollan proyectos usando software libre como PHP, MySQL y Apache como servidor Web.

## CAPÍTULO III. METODOLOGÍA

### 3.1. Descripción del área de estudio

La investigación se realizará en las oficinas del Departamento de Desarrollo de la Cooperativa Financiera Atuntaqui.

### 3.2. Tipo de investigación

El enfoque de la tesis es cualitativo por observación y registros visuales de la forma como se recopila requerimientos

### 3.3. Diseño de la investigación

#### 3.3.1. Modalidad de investigación

La investigación será bibliográfica porque utiliza fuentes de consulta como libros, artículos científicos indexados y publicados.

La investigación tendrá la modalidad de campo porque busca obtener la información de la variable dependiente fase de requerimiento y variable independiente software de gestión en el lugar mismo en que ocurre.

#### 3.3.2 Tipos o Niveles de Investigación

##### **Investigación Exploratoria**

Se realiza en el Departamento de Desarrollo de la Cooperativa Financiera Atuntaqui por la falta de aplicación de estándares.

##### **Investigación Descriptiva**

Por medio de la recolección, análisis y encuestas se llegará a identificar la relación entre la variable independiente y dependiente.

##### **Investigación Explicativa**

Dentro de la investigación se intentará comprobar la siguiente afirmación:

¿El software de gestión incide en la fase de análisis de requerimientos funcionales para el desarrollo de software?

### 3.4. Población y Muestra

La investigación se realizará a las jefaturas de las diferentes áreas de la Cooperativa Financiera Atuntaqui, ya que ellos intervienen junto con las áreas de Tecnologías en la toma de requerimientos.

**Tabla 4.** *Población y muestra de investigación.*

<b>Población</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Jefe de Sistemas	1	10%
Jefe de Operaciones	1	10%
Jefe Financiero	1	10%
Jefe de Seguridades	1	10%
Jefe de Negocios	1	10%
Jefe de Talento Humano	1	10%
Jefe de Riesgos	1	10%
Personal de Desarrollo	3	30%
<b>Total</b>	<b>10</b>	<b>100%</b>

**Fuente:** Autor

Como la población en donde se va a desarrollar el proyecto de Investigación no pasa de 100 personas, se trabajará con la totalidad del universo sin que sea necesario sacar muestras representativas.

### 3.5. Métodos

Deductivo:

“La deducción es un proceso que parte de un principio general ya conocido para inferir de él, consecuencias particulares” (Gutiérrez, Curso de Métodos de Investigación, 2006).

Este método permite observar el proceso actual que se lleva a cabo en la toma de requerimientos, indicar y proponer soluciones adecuadas y óptimas para el Departamento de Desarrollo de la Cooperativa Financiera Atuntaqui.

### **3.6. Estrategias Técnicas**

Se utilizarán las siguientes técnicas:

- Entrevistas: Se aplicará a las jefaturas.
- Encuesta: para poder realizar sondeos y medir opiniones del proceso que se lleva en la toma de requerimientos.
- Observación: se realizará visitas a las reuniones de la toma de requerimientos con el personal involucrado.

### **3.7. Instrumentos**

Los instrumentos que se emplearán serán:

- Preguntas en cuestionarios.
- Fichas de observación.
- Filmadora.
- Celular como equipo de comunicación.
- Cámara fotográfica.

## CAPÍTULO IV. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

### 4.1 Comparativa entre el estándar IEEE 830 y el estándar ISO/IEC/IEEE 29148

Según el análisis realizado al artículo del autor Henning Femmer titulado “Reviewing Natural Language Requirements with Requirements Smells – A Research Proposal –” (2013) realiza su estudio con el estándar IEEE 830-1998 y el estándar ISO/IEC/IEEE 29148-2011, debido a los siguientes criterios:

*“most relevant is the IEEE 830-1998 (IEEE Recommended Practice for Software Requirements Specifications)”* (Femmer, 2013, p. 3).

*“ISO-29148 is the most current and complete official standard on requirements engineering”* (Femmer, 2013, p. 4).

El autor en el artículo mencionado toma en consideración los dos estándares debido a las características de usabilidad y conforme al estudio comparativo realizado por los autores Florian Schneider y Brian Berenbach en el artículo titulado “A Literature Survey on International Standards for Systems Requirements Engineering” (2013), en el cual hace mención a las ventajas del estándar ISO 29148-2011 como la siguiente definición:

*“ISO/IEC/IEEE 29148:2011 is actually the standard that every requirements engineer should be familiar with. It is not only a collection of the most basic principles of requirements engineering, it also hints at how high-quality requirements can be achieved”* (Schneider y Berenbach, 2013:803)

Con este antecedente se ha decidido realizar la comparativa de estos dos estándares para el estudio propuesto en vista de los resultados que obtuvieron los autores mencionados en sus investigaciones realizadas.

Como se realizó el detalle del contenido de cada uno de los estándares, a continuación, se muestra la comparativa realizada de las ventajas que provee cada uno de ellos para la realización de la especificación de requerimientos de software para desarrollar un producto. Estas

características se obtuvieron después de haber realizado el análisis e interpretación del estándar y por lo tanto los valores obtenidos son cualitativos.

A continuación, se detalla la tabla con las comparaciones realizadas de características cualitativas que ofrece cada estándar.

**Tabla 5.** *Comparación de características entre IEEE-830 y ISO/IEC/IEEE 29148*

<b>Característica</b>	<b>IEEE 830-1998</b>	<b>ISO/IEC/IEEE 29148-2011</b>
<b>Ayuda a los clientes de software a describir con precisión lo que se desea.</b>	Si	Si
<b>Ayuda a los desarrolladores a entender exactamente lo que el cliente quiere.</b>	Si	Si
<b>Desarrolla una plantilla para la especificación de requisitos de software.</b>	Si	Si
<b>Establece las bases para un acuerdo entre los clientes y los programadores.</b>	Si	Si
<b>Reduce el esfuerzo de desarrollo.</b>	Si	Si
<b>Proporciona una base para estimaciones realistas de costos</b>	Si	Si

<b>Proporciona una base para la validación y verificación de requisitos.</b>	Si	Si
<b>Facilita la transferencia del producto de software a nuevos usuarios o nuevas máquinas.</b>	Si	Si
<b>Armoniza IEEE 830, SWEBOK y otros 7 estándares</b>	No	Si
<b>Mayor énfasis en las características de los buenos requisitos, las actividades y procesos de ER, las operaciones y los diferentes elementos de información.</b>	No	Si
<b>Cumple con ISO / IEC 15288 e ISO / IEC 12207</b>	No	Si

**Fuente:** Autor

La tabla anterior refleja las características principales de los dos estándares estudiados; como se puede observar algunas cualidades la comparte y otras la difieren. Entre las cualidades semejantes tenemos la de brindar un apoyo al desarrollador para realizar una óptima gestión de requerimientos, proveen a que los clientes junto con los especialistas puedan desglosar las necesidades correctamente, de esta manera se puede obtener un costo real y el tiempo prudente para la finalización de un proyecto.

De acuerdo con la revisión realizada a los dos estándares, el estándar IEEE-830 ofrece un desarrollo rápido de requerimientos mientras que el ISO/IEC/IEEE-29148 ofrece un amplio contenido para gestionar requerimientos enfocándose en una profunda calidad de gestión y con mayor inversión de esfuerzo y tiempo. Estos datos fueron obtenidos de acuerdo con la encuesta

número dos en el anexo E enfocada al uso de ambos estándares en la Cooperativa Financiera Atuntaqui.

#### **4.2 Interpretación de resultados**

Con los datos obtenidos de la segunda encuesta, se define la opinión sobre el estándar a utilizar en la fase de requerimientos en el desarrollo de software de la Cooperativa Financiera Atuntaqui, de acuerdo con los resultados obtenidos se puede destacar los siguiente:

##### a) Conclusiones

- Los usuarios (jefes departamentales de la institución), al comparar los dos estándares en diferentes proyectos, se inclinaron sobre el estándar IEEE-830.
- Los usuarios notaron que en los proyectos de la Institución el estándar ISO/IEC/IEEE-29148 era muy complejo por la gran cantidad de aspectos que se debe considerar para gestionar los requerimientos.
- Una jefatura se inclinó por utilizar el estándar ISO/IEC/IEEE-29148 pero en proyectos nuevos y de gran magnitud.
- El estándar IEEE-830 redujo el tiempo de gestionar los requerimientos y de igual manera los recursos empleados.
- El estándar IEEE-830 se adapta mejor a proyectos nuevos y al mantenimiento de software que ya se encuentran en producción.
- Las jefaturas manifestaron que el uso del estándar IEEE-830 aporta un gran atributo de calidad al software desarrollado en la Institución.

##### b) Recomendaciones

- Se debe utilizar el estándar IEEE-830 para los proyectos nuevos y de mantenimiento del software en la Cooperativa Financiera Atuntaqui.
- Se debe realizar los cambios necesarios (si los hubiere) del formato de gestión de requerimientos para que se adapte de mejor manera al giro del negocio de la Institución.
- Se debe capacitar al personal de desarrollo en temas de gestión de procesos.
- Se debe establecer y formalizar el documento de gestión de requerimientos en base al estándar IEEE-830.

c) Resultados

Con la comparación realizada a las características cualitativas de los estándares estudiados y con los resultados de la segunda encuesta realizada a los jefes departamentales de la Institución podemos decir que:

- 1) Del total de encuestados, el 100% hace énfasis en la utilización del estándar IEEE-830 para los proyectos nuevos y de mantenimiento en la Institución.
- 2) Del total de encuestados, el 100% menciona que el uso y aplicación dentro de la institución del estándar IEEE-830 aporta mejores resultados a la gestión de requerimientos.
- 3) Del total de encuestados, el 100% menciona que utilizando el estándar IEEE-830 reduce el tiempo para gestionar los requerimientos y de igual manera los recursos materiales y de personal.

## **CAPÍTULO V. PROPUESTA**

### **5.1. Antecedentes**

La Cooperativa Financiera Atuntaqui dispone de una infraestructura de servidores web propietarios como el Internet Information Services y gratuitos como Apache; de igual manera en la institución se desarrollan aplicaciones con software propietario como Visual Studio y software gratuito como PHP.

Como la infraestructura de la institución ofrece disponibilidad para cualquier tipo de aplicación (Web y Escritorio) y de acuerdo con el conversatorio realizado con el jefe de sistemas y el administrador de desarrollo, se optó que el prototipo para gestionar los requerimientos sea de tipo web y desarrollado con las herramientas gratuitas (PHP, MySQL) que ya hace uso la institución.

#### **5.1.1. Análisis del proceso actual de desarrollo**

La siguiente figura muestra el proceso actual que realizan las jefaturas para solicitar el desarrollo o la modificación de un sistema.

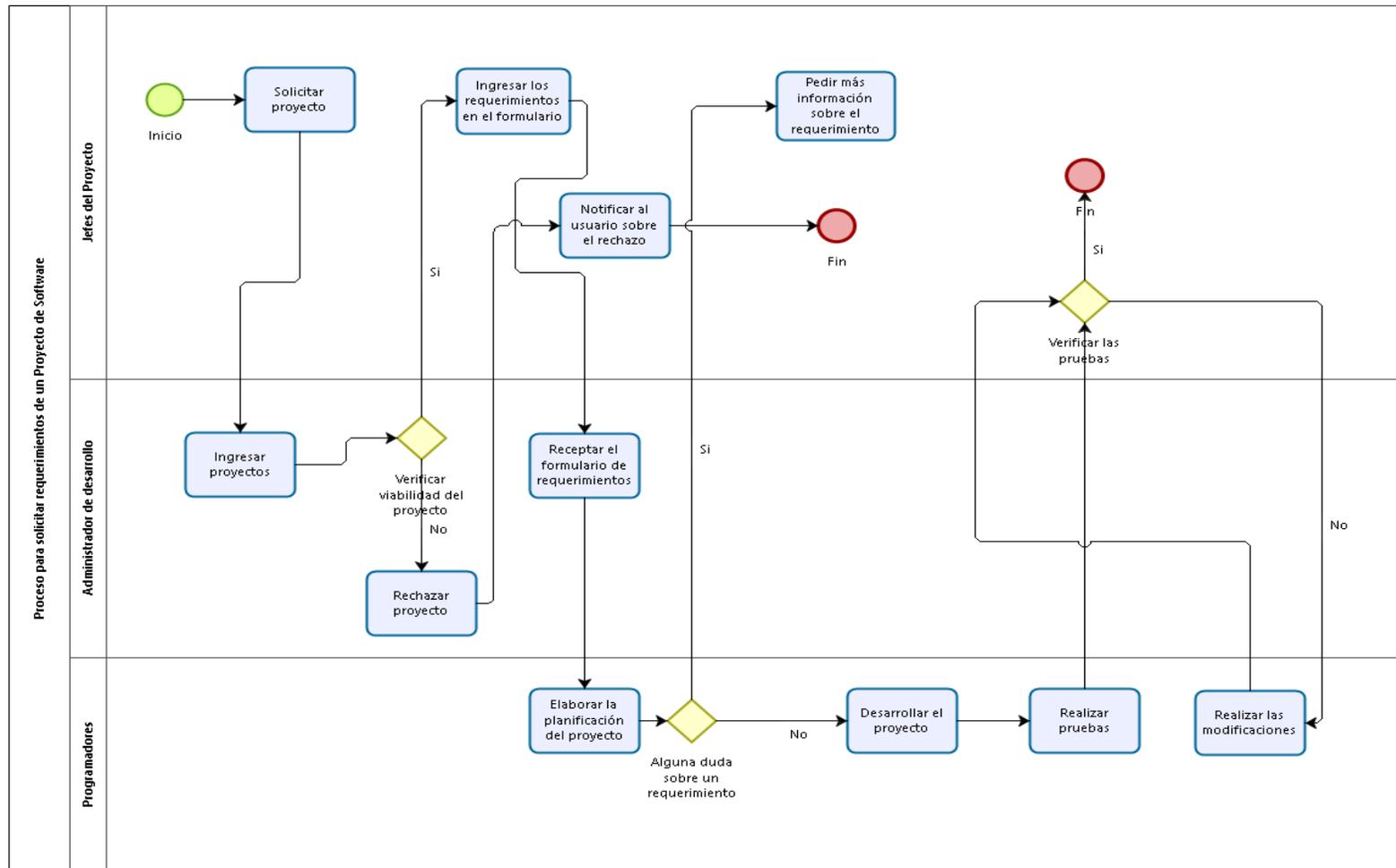


Figura 6 Proceso actual para solicitar requerimientos sobre un proyecto de software  
Fuente Autor

## 5.2. Propuesta de mejora al proceso de gestión de requerimientos

De acuerdo con el estudio realizado y los resultados presentados en la encuesta, la propuesta tiene dos partes; la primera es la de tener un conversatorio con el área de sistemas y los jefes departamentales para realizar lo siguiente:

- 1) Proponer la formación de un equipo de trabajo con la jefatura de sistemas de la institución (jefe de sistemas, administrador de desarrollo) para gestionar los requerimientos, en donde de manera conjunta con los usuarios (jefes departamentales) se pueda realizar un levantamiento correcto de las necesidades sobre cierta temática; ya sea para proyectos nuevos como para actualización de aplicativos.
- 2) Seguir el esquema del estándar IEEE 830, tomando las buenas prácticas para la gestión de requerimientos y que cada uno de ellos contemple lo estipulado en la norma sugerida. Se puede empezar a aplicar la norma en los proyectos nuevos y de manera progresiva con la actualización de aplicativos con el fin de ir adaptando el documento a las necesidades del área, sin olvidar el cumplimiento del cómo y del qué debe tener cada requerimiento.
- 3) Utilizar el siguiente formato de trabajo (Véase la sección de Anexos: Ejemplo del documento de especificación de requerimientos) de acuerdo con el estándar IEEE 830, en donde debe contemplar:
  - a. Propósito.
  - b. Alcance.
  - c. Personal involucrado.
  - d. Definiciones, acrónimos y abreviaturas.
  - e. Referencias.
  - f. Resumen.
  - g. Perspectiva del producto.
    1. Funcionalidad del producto.
    2. Características de usuario.
    3. Restricciones.
    4. Suposiciones y dependencias.

- h. Requerimientos específicos.
  - 1. Interfaces de usuario.
  - 2. Interfaces de hardware.
  - 3. Interfaces de software.
  - 4. Interfaces de comunicaciones.
- i. Requerimientos funcionales.
- j. Requerimientos de usabilidad.
- k. Requisitos de rendimiento.
- l. Atributos del sistema.

La otra parte de la propuesta consiste en un prototipo de software para gestionar la documentación de requerimientos en donde se va a tener lo siguiente

- a) Ingreso de datos del proyecto, el responsable y el personal de apoyo.
- b) Subir el documento de requerimientos en formato PDF.
- c) Ingreso de requerimientos con el avance y estado.
- d) Visualización del historial del documento subido.

El prototipo consta de un ingreso con usuario y contraseña; se puede subir el documento de especificación de requerimientos basados en el estándar IEEE-830 y de igual manera se puede ingresar los requerimientos en donde se puede observar el porcentaje de avance de este y el estado actual del requerimiento.

De esta manera se puede conocer el avance de los requerimientos por cada proyecto, además permite tener una comprensión rápida de la estructura y funcionalidad del proyecto.



### 5.3. Desarrollo del prototipo utilizando SCRUM

En función de la encuesta realizada y de los resultados obtenidos en el capítulo anterior se puede justificar el desarrollo de un prototipo de software para la gestión de requerimientos funcionales para la Cooperativa Financiera Atuntaqui.

Para el desarrollo del prototipo se utilizó la metodología SCRUM, de igual manera se utilizó PHP como lenguaje de desarrollo, MySQL como base de datos, Apache como servidor Web y jQuery para dinamizar las páginas y mejorar la interacción del usuario con el sistema.

### 5.4. Definición de Requerimientos

Para determinar los requerimientos del prototipo, se realizaron conversatorios con la jefatura de sistemas de la institución. Se detallaron algunas historias de usuario que fueron considerables las más importantes para el estudio propuesto; a futuro se puede incluir algunas mejoras, dependiendo de la evolución del sistema. En el anexo G se encuentra el documento ERS del prototipo, el cual se ha desglosado para crear las historias de usuario y el Product Backlog.

#### 5.4.1. Historias de usuario

Para el prototipo se consideraron las siguientes historias de usuario con sus tareas definidas:

a) Pantalla de Login

Como usuario Administrador quiero ingresar al sistema de gestión de requerimientos para poder gestionar los proyectos, requerimientos y analizar los reportes.

*Criterios de aceptación:*

- Dado que me encuentro en la página de login, cuando ingrese el usuario y clave correctamente entonces puedo acceder a la aplicación.
- Dado que me encuentro en la página de login, cuando ingrese el usuario y clave errónea entonces puedo visualizar los mensajes de advertencia.
- Dado que me encuentro en la página de login, cuando no ingrese el usuario y clave puedo visualizar un mensaje de campos obligatorios.
- Dado que me encuentro en la página de login, cuando ingrese el usuario y clave correctamente entonces puedo acceder al módulo de administrador.

*Tareas:*

- Implementar una pantalla de Login con un usuario y clave.
- Validar las credenciales para el usuario administrador con un usuario y una clave predeterminada inicial.
- Mostrar mensajes de advertencia si las credenciales son incorrectas.
- Diseñar plan de pruebas.
- Ejecutar plan de pruebas.

Como usuario Programador quiero ingresar al sistema para poder visualizar los requerimientos e ingresar el avance correspondiente.

*Criterios de aceptación:*

- Dado que me encuentro en la página de login, cuando ingrese el usuario y clave correctamente entonces puedo acceder a la aplicación.
- Dado que me encuentro en la página de login, cuando ingrese el usuario y clave errónea entonces puedo visualizar los mensajes de advertencia.
- Dado que me encuentro en la página de login, cuando no ingrese el usuario y clave puedo visualizar un mensaje de campos obligatorios.
- Dado que me encuentro en la página de login, cuando ingrese el usuario y clave correctamente entonces puedo acceder al módulo de programador.

*Tareas:*

- Implementar una pantalla de login con un usuario y clave.
- Validar las credenciales una vez que el usuario administrador cree los usuarios con rol de programadores.
- Mostrar mensajes de advertencia si las credenciales son incorrectas.
- Diseñar plan de pruebas.
- Ejecutar plan de pruebas.

## b) Gestión de Empleados

Como usuario administrador quiero gestionar los empleados con sus roles para que los usuarios programadores puedan ingresar al sistema.

*Criterios de aceptación:*

- Dado que me encuentre en la página de administrar empleados, cuando ingrese los valores correctamente entonces puedo visualizar el usuario creado.
- Dado que me encuentre en la página de administrar empleados, cuando modifique un registro entonces puedo ver los cambios realizados.
- Dado que me encuentre en la página de administrar empleados, cuando ingrese una cédula no válida entonces se visualiza el mensaje de advertencia correspondiente.
- Dado que me encuentre en la página de administrar empleados, cuando trate de guardar información en blanco entonces se visualiza el mensaje de advertencia correspondiente.

*Tareas:*

- Implementar un formulario de ingreso, modificación y eliminación con los datos del proyecto.
- Implementar el backend para almacenar los datos.
- Validar los datos de entrada del formulario.
- Diseñar plan de pruebas.
- Ejecutar plan de pruebas.

### c) Gestión de Proyectos

Como usuario administrador quiero gestionar los proyectos internos de la institución para poder almacenar el documento de especificación de requerimientos.

*Criterios de aceptación:*

- Dado que me encuentre en la página de administrar proyectos, cuando ingrese los valores correctamente entonces puedo visualizar el proyecto creado.
- Dado que me encuentre en la página de administrar proyectos, cuando modifique un registro entonces puedo ver los cambios realizados.
- Dado que me encuentre en la página de administrar proyectos, cuando no seleccione al menos una persona de apoyo entonces se visualiza el mensaje de advertencia correspondiente.

- Dado que me encuentro en la página de administrar proyectos, cuando trate de guardar información en blanco entonces se visualiza el mensaje de advertencia correspondiente.

*Tareas:*

- Implementar un formulario de ingreso con los datos del proyecto.
- Implementar el backend para almacenar los datos.
- Validar los datos de entrada del formulario.
- Diseñar plan de pruebas.
- Ejecutar plan de pruebas.

d) Gestión de Requerimientos

Como usuario administrador quiero gestionar los requerimientos de cada proyecto para poder almacenar los datos y visualizar el avance de trabajo de cada requisito.

*Criterios de aceptación:*

- Dado que me encuentro en la página de administrar requisitos, cuando seleccione un proyecto e ingrese los valores correctamente entonces puedo visualizar el requerimiento creado.
- Dado que me encuentro en la página de administrar requisitos, cuando seleccione un proyecto y modifique un registro entonces puedo ver los cambios realizados.
- Dado que me encuentro en la página de administrar requisitos, cuando no seleccione un proyecto entonces no se puede ingresar requerimientos.
- Dado que me encuentro en la página de administrar requisitos, cuando trate de guardar información en blanco entonces se visualiza el mensaje de advertencia correspondiente.

*Tareas:*

- Implementar un formulario de ingreso con los datos del proyecto.
- Implementar el backend para almacenar los datos.

- Validar los datos de entrada del formulario.
- Diseñar plan de pruebas.
- Ejecutar plan de pruebas.

e) Gestión de Estructuras del Proyecto

Como usuario administrador quiero ingresar la información de la estructura del proyecto conforme al estándar IEEE 830 para almacenar la información y poder imprimir el documento de ERS.

*Criterios de aceptación:*

- Dado que me encuentre en la página de administrar estructura de proyectos, cuando ingrese los valores correctamente entonces puedo visualizar el proyecto creado.
- Dado que me encuentre en la página de administrar estructura de proyectos, cuando modifique un registro entonces puedo ver los cambios realizados.
- Dado que me encuentre en la página de administrar proyectos, cuando no seleccione un proyecto entonces no se puede ingresar la información.
- Dado que me encuentre en la página de administrar estructura de proyectos, cuando ingrese los valores correctamente se creará una versión del documento automáticamente.

*Tareas:*

- Implementar un formulario de ingreso con los datos del proyecto.
- Implementar el backend para almacenar los datos.
- Validar los datos de entrada del formulario.
- Diseñar plan de pruebas.
- Ejecutar plan de pruebas.

#### 5.4.2. Product Backlog

En la siguiente figura se muestra el Backlog con los requerimientos del prototipo, se utilizó una herramienta gratuita para tener una mejor percepción visual del trabajo.

Votes	User Stories	Status	Points
<input type="checkbox"/>	▲ 0 #1 Pantalla de Login	New	9
Project Scope [Doomline]			
<input type="checkbox"/>	▲ 0 #2 Gestión de Empleados	New	13
<input type="checkbox"/>	▲ 0 #3 Gestión de Proyectos	New	13
<input type="checkbox"/>	▲ 0 #4 Gestión de Requerimientos	New	13
<input type="checkbox"/>	▲ 0 #5 Gestión de Estructuras del Proyecto	New	13

Figura 8 Product Backlog del prototipo.

### 5.4.3. Sprint Planning

Al revisar las historias de usuario, se pudo agrupar en tres sprints; cada uno con un plazo de dos semanas; adicional a esto se realizó un sprint para el diseño de la base de datos. A continuación, se detalla cada uno de los sprints:

**Tabla 6.** *Sprint número uno.*

Nro.	Actividad	Tiempo (Horas)
1	Diseño de la Base de Datos	6
2	Selección del template del proyecto Web	2
3	Pruebas	4

Fuente: Autor

**Tabla 7.** *Sprint número dos.*

Nro.	Actividad	Tiempo (Horas)
1	Pantalla de Login (Autenticación y Autorización)	4
2	Gestión de empleados	6
3	Gestión de Proyectos	6
4	Pruebas	8

Fuente: Autor

**Tabla 8.** *Sprint número tres*

Nro.	Actividad	Tiempo (Horas)
1	Gestión de requerimientos	6
2	Gestión de estructuras de proyectos	6
3	Pruebas	8

Fuente: Autor

#### 5.4.4. Implementación y pruebas

Para el desarrollo del prototipo de nombre Arjé se utilizó las siguientes herramientas:

- Lenguaje de desarrollo PHP
- Base de Datos MySQL 5.6.24
- Servidor Apache
- jQuery 1.11.3

##### 5.4.4.1. Configuración de la conexión a la Base de Datos

El archivo de configuración para que PHP se conecte con la base de datos será la siguiente:

```
<?php
    class Conexion{
        protected $dbhost = "localhost";
        protected $dbuser = "root";
        protected $dbpass = "";
        protected $dbname = "bugs_sgdr";
        protected $con;
        public function Conexion(){
            $this->con = new mysqli($this->dbhost,$this->dbuser,$this->dbpass,$this->dbname) or die('cannot connect to the server');
            return $this->con;
        }
    }
?>
```

En este archivo debemos especificar la dirección del Servidor Web de la institución, el usuario de la base de datos, la contraseña y el nombre de la base de datos.

El llamado a esta clase se la debe realizar en los archivos que tengan interacción con la base de datos.

#### 5.4.4.2. Configuración del Servidor Web

Esta configuración es proporcionada por la Institución de manera privada por lo que se no se detalla ninguna información por pedido de la Cooperativa Financiera Atuntaqui.

#### 5.4.4.3. Configuración de jQuery

jQuery es una librería de Javascript que permite a las páginas de PHP ser dinámicas, mejorar su rendimiento y funcionalidad.

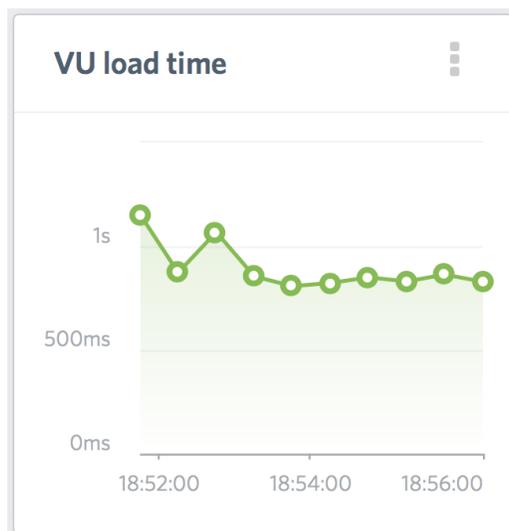
Para que jQuery se pueda utilizar en las páginas de PHP, se debe agregar la referencia al archivo mencionado de la siguiente manera:

```
<script type="text/javascript" src="js/jquery-1.11.3.min.js"></script>
```

En este caso el archivo de javascript que contiene la librería de jQuery se encuentra en la carpeta js/, esta referencia debe estar en todas las páginas que requieran el uso de este recurso.

#### 5.4.4.4. Pruebas

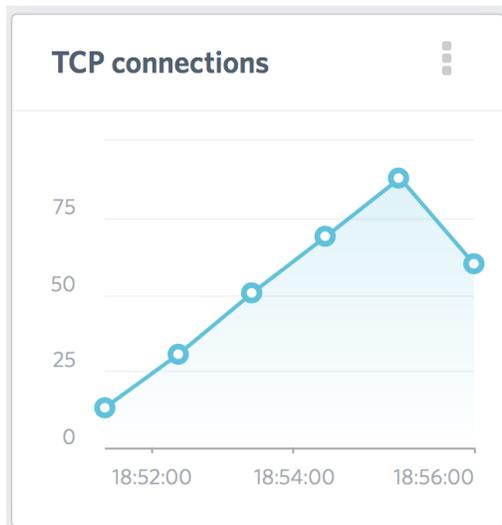
Se realizaron pruebas de carga y estrés de la aplicación utilizando la herramienta Load Impact para conocer el rendimiento y respuesta que tiene la aplicación cuando acceden varios usuarios.



*Figura 9* Tiempos de carga de la aplicación.

**Fuente:** Autor

La siguiente figura muestra el número de conexiones TCP realizadas en la prueba a la aplicación.



*Figura 10* Vista de conexiones TCP realizadas al sistema.

**Fuente:** Autor

En la siguiente figura muestra el resultado de petición por cada segundo.

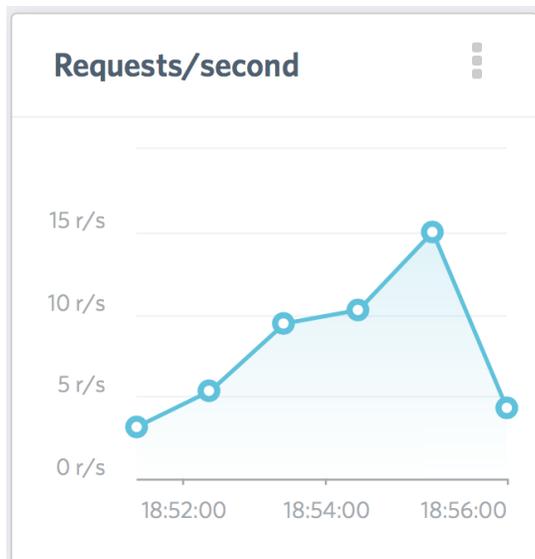


Figura 11 Vista de solicitudes por segundo realizadas al sistema.

Fuente: Autor

La siguiente figura muestra el total de solicitudes realizadas al sistema.



Figura 12 Vista del total de solicitudes realizadas al sistema.

Fuente: Autor

Otro tipo de pruebas se realizaron con escenarios para poder verificar el funcionamiento del aplicativo con la base de datos.

A continuación, se detalla los escenarios propuestos y los resultados obtenidos:

**Tabla 9.** *Escenarios de prueba uno.*

<b>Tipo de escenario</b>	<b>Formación</b>	<b>Número de Intentos</b>	<b>Resultados erróneos</b>	<b>Resultados satisfactorios</b>
<b>Validación de credenciales</b>	Verificar que al ingresar las credenciales correctas se pueda ingresar a la aplicación.	10	0	10
	Verificar que al ingresar credenciales incorrectas se muestren los mensajes correspondientes	10	0	10

**Fuente:** Autor

**Tabla 10.** *Escenarios de prueba dos.*

<b>Tipo de escenario</b>	<b>Formación</b>	<b>Número de Intentos</b>	<b>Resultados erróneos</b>	<b>Resultados satisfactorios</b>
<b>Validación de datos de parametrización</b>	Verificar que al ingresar, modificar y eliminar los datos sean correctos.	10	0	10

**Fuente:** Autor

Otra prueba importante es la de verificar el avance de los sprint, los cuales se ven reflejados en los gráficos de esfuerzo o burndown, la siguiente figura muestra el esfuerzo realizado para el primer sprint:

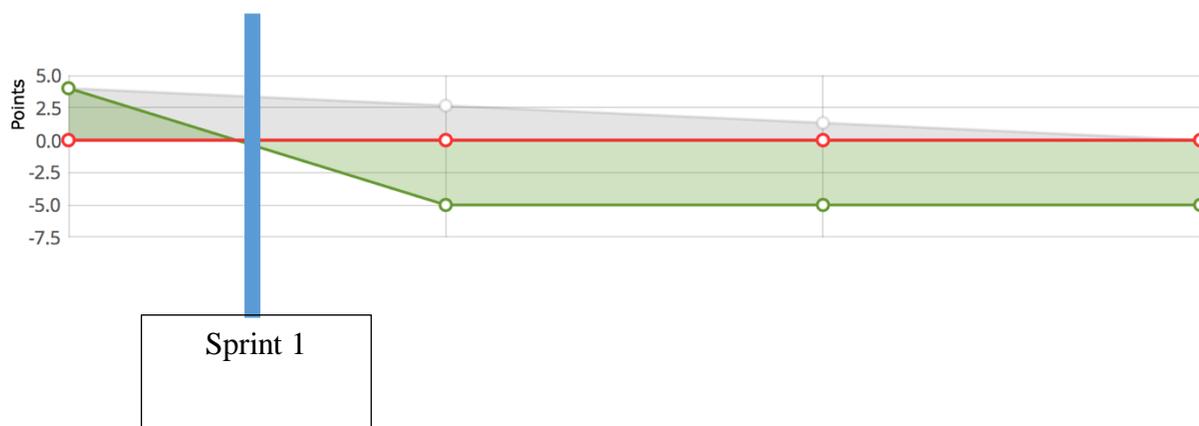


Figura 13 Gráfico Burndown

Tabla 11. Escenarios de prueba tres.

Tipo de escenario	Formación	Número de Intentos	Resultados erróneos	Resultados satisfactorios
Validación de datos del proyecto	Verificar que al ingresar, modificar y eliminar los datos sean correctos.	10	0	10

Fuente: Autor

Tabla 12. Escenarios de prueba cuatro.

Tipo de escenario	Formación	Número de Intentos	Resultados erróneos	Resultados satisfactorios
Validación de datos de las etapas del proyecto	Verificar que al ingresar, modificar y eliminar los datos sean correctos.	10	0	10

Fuente: Autor

Tabla 13. Escenarios de prueba cinco.

Tipo de escenario	Formación	Número de Intentos	Resultados erróneos	Resultados satisfactorios
-------------------	-----------	--------------------	---------------------	---------------------------

<b>Validación de carga y almacenamiento de los archivos en PDF</b>	Verificar que, al subir el archivo, los datos sean correctos.	10	0	10
--	---	----	---	----

Fuente: Autor

### 5.5. Medición del impacto del software de gestión

Para obtener un valor sobre el impacto que tiene la implementación del software de gestión de requerimientos funcionales, se consideraron las fórmulas de los autores (Uwizeyemungu & Raymond, 2012) en el artículo de tema “Impact of an ERP system’s capabilities upon the realization of its business value: a resource-based perspective”, en el cual valora la implementación de un ERP (Enterprise Resourcing Planning) en diferentes empresas y de acuerdo a las valoraciones realizadas por el departamento de tecnologías de cada una de ellas, se obtiene el valor del impacto asociado al ERP.

Si bien es cierto el prototipo no se asemeja a un ERP, tampoco se puede aplicar cálculos relacionados con los impactos de proyectos sociales o proyectos de negocio. Por esta razón se ha adoptado el cálculo de implementar un ERP, el cual puede brindar un resultado cercano a la realidad del prototipo desarrollado y de abarcar el objetivo propuesto.

En el artículo de los autores (Uwizeyemungu & Raymond, 2012) muestra la siguiente fórmula:

$$impactoEfectos = \sum_{i=1,n} (a_i * b_i)$$

En donde la variable  $a$  corresponde a la importancia del indicador de rendimiento del ERP, con la siguiente escala de valoración: [1] nada importante, [2] poco importante, [3] medianamente importante, [4] importante y [5] muy importante. Estos valores son resultado de aplicar el cuestionario al área de tecnologías sobre la importancia del prototipo desarrollado en la institución.

La variable  $b$ , corresponde al grado de variación del indicador de rendimiento inducido por el efecto del ERP, es decir que estos valores corresponden a la operación actual de los procesos de

la institución, utilizando el prototipo para gestionar requerimientos funcionales; la escala de valoración es: [0] ninguno, [1] débil, [2] medio y [3] fuerte.

La segunda fórmula corresponde a la variación del indicador de rendimiento basado en un impacto positivo del prototipo, es decir que en la escala del 0 al 3, el valor de  $B = 3$ ; entonces se tendría:

$$impactoVariaciones = \sum_{i=1,n} (a_i * B_i)$$

Una vez obtenidos los valores de las sumatorias, se divide para obtener el resultado del impacto global del prototipo desarrollado y aplicado a la institución, el resultado de la fórmula sería:

$$impactoGlobal = 5 * \frac{impactoEfectos}{impactoVariaciones}$$

Para tener un resultado de mayor comprensión se transforma el valor a una escala de cinco puntos; de esta manera se tiene una mejor apreciación del impacto obtenido del prototipo aplicado a la institución.

A continuación, se muestra los valores obtenidos y los resultados de las fórmulas aplicadas.

**Tabla 14.** *Medición del impacto del protoipo*

<b>Fórmula</b>	<b>Prototipo de Gestión de requerimientos funcionales</b>
(1) $\sum_{i=1,n} (a_i * b_i)$	143
(2) $\sum_{i=1,n} (a_i * B_i)$	222
(1) / (2)	0.64414414
<b>Resultado</b>	3.22
<b>normalizado a escala de 5 puntos</b>	

**Fuente:** Autor

El valor del impacto resultante en la escala de cinco puntos es de **3.22**, correspondiente a medianamente importante.

**Tabla 15.** *Cálculo de valores para medir el impacto del prototipo*

Efectos	Indicador de rendimiento (PI)					
	PI Estudiado	i	a	b	a*b	a*B
<b>Efectos</b>	Mejora de los procesos internos	1	4	2	8	12
<b>Automatización</b>	Mejoras laborales	2	1	2	2	3
	Rotación de personal (Jefaturas)	3	2	1	2	6
	Tiempo medio de recolección de requisitos de los clientes	4	3	2	6	9
	Retrasos en el envío de requerimientos	5	2	1	2	6
	Mecanismos de elicitación	6	2	2	4	6
	Priorizar requerimientos	7	3	2	6	9
	% de cambios solicitados por los clientes	8	4	2	8	12
	Interacción entre clientes	9	2	1	2	6
	Integración con base de datos interna	10	5	2	10	15
	Integración con otras aplicaciones internas	11	1	2	2	3
	Auditoría de avance de proyectos	12	4	2	8	12
<b>Subtotal efectos automatización</b>					<b>60</b>	<b>99</b>
<b>Efectos informativos</b>	Entregas tardías del software a producción	13	3	2	6	9
	Retrasos en cronograma de entregables	14	4	2	8	12
	Tiempo de ciclo de vida del software en producción	15	4	2	8	12
	% de software modificados por clientes	16	3	2	6	9
	% de cambios al software en producción	17	3	2	6	9

	Aporte de los clientes	18	1	2	2	3
	% Uso de documentación estandarizada	19	2	2	4	6
	% de requerimientos modificados por clientes	20	3	2	6	9
	Razones de productividad	21	4	3	12	12
	Reportes	22	4	2	8	12
<b>Subtotal de efectos informativos</b>					<b>66</b>	<b>93</b>
<b>Efectos de transformación</b>	Planificación de proyectos	23	3	2	6	9
	Número de nuevos productos enviados a producción	24	3	1	3	9
	Cronograma de entregables	25	4	2	8	12
<b>Subtotal para efectos de transformación</b>					<b>17</b>	<b>30</b>
<b>TOTAL</b>					<b>143</b>	<b>222</b>

Fuente: Autor

## **5.6. Resultado de la propuesta**

Una vez desarrollado el prototipo de gestión de requerimientos, se realizaron las pruebas correspondientes para comprobar el funcionamiento y beneficio que tiene el prototipo al proceso de desarrollo de software.

Se realizó una última encuesta con dos preguntas para verificar el grado de satisfacción del prototipo y conocer el impacto que el sistema tiene en la gestión de requerimientos funcionales en la Cooperativa Financiera Atuntaqui.

Los resultados de las dos preguntas se encuentran en el anexo F; por lo tanto, se puede mencionar que el impacto que produce implementar un software de gestión para la fase de análisis de requerimientos en la Cooperativa Financiera Atuntaqui es favorable y beneficioso con un 64.4% de aceptación por parte de las jefaturas de la Institución.

## CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES

### 6.1 Conclusiones

1. La gestión de requerimientos en la institución no es la adecuada y es necesario aplicar mejoras al proceso, además se debe concientizar de los resultados obtenidos en las encuestas iniciales para desarrollar el software de mejor manera.
2. El prototipo desarrollado ayuda a tener de manera organizada los documentos de requerimientos de cada proyecto, proporciona soporte para los desarrolladores acerca de las necesidades del usuario, permite la comprensión de la funcionalidad de un sistema y ayuda al programador para realizar un correcto mantenimiento.
3. Con la aplicación de estándares a los sistemas desarrollados dentro de la institución y con la conformación de un equipo de expertos en gestionar los requerimientos, el cronograma de entregables ha mejorado y de igual manera la satisfacción del usuario final.
4. El valor del impacto es un resultado preliminar el cual con el tiempo puede variar positivamente o negativamente según el uso del prototipo.

## 6.2 Recomendaciones

1. Se debe conformar un equipo de trabajo dentro de la institución con la ayuda del departamento de sistemas para que junto con el usuario se gestionen los requerimientos de manera clara, precisa y confiable para el desarrollo del producto.
2. En el equipo de trabajo se debe incluir al o a los programadores que van a desarrollar el sistema para que conozcan las necesidades del usuario, de igual manera el proceso sobre una determinada actividad y permitir dar opiniones y cambios a los requerimientos si los hubiere.
3. Se debe documentar de manera clara cada requerimiento, indicando paso a paso el procedimiento de la actividad, mostrar las restricciones correspondientes y establecer los parámetros de evaluación para los casos de error o satisfactorio en las pruebas a cada requerimiento.
4. Se debe capacitar a las áreas de tecnologías en temas sobre administración de proyectos, calidad de software y temas de gestión para que se pueda concientizar en la importancia que tienen estos conocimientos para desarrollar software de calidad.
5. Se debe mejorar la asignación de roles en el equipo de desarrollo de tal manera que el conocimiento se distribuya y mejore el proceso actual de desarrollo de software.

## REFERENCIAS BIBLIOGRÁFICAS

1. Arias, M. (2006). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *Revista InterSedes*, 10-11.
2. Beal, V. (2016). *Webopedia*. Obtenido de management software: [https://www.webopedia.com/TERM/M/management\\_software.html](https://www.webopedia.com/TERM/M/management_software.html)
3. Bourque, P., & Fairley, R. (2014). *SWEBOK v3.0*. New Jersey: IEEE Computer Society.
4. CMMI. (2016). *CMMI Institute*. Obtenido de Published Appraisal Results: <https://sas.cmmiinstitute.com/pars/pars.aspx>
5. CMMI Institute. (Junio de 2016). *CMMI Institute*. Obtenido de <http://partners.cmmiinstitute.com/wp-content/uploads/2016/09/Maturity-Profile-Ending-Jun-30-2016.pdf>
6. Dávila, A., Melendez, K., & Flores, L. (2006). Determinación de los Requerimientos de Calidad del Producto Software Basados en Normas Internacionales. *IEEE LATIN AMERICA TRANSACTIONS*, 100-101.
7. Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2017). *Una introducción básica a la teoría y práctica de Scrum*. Scrum Primer.
8. Ecuador, C. d. (2008). *Registro Oficial*. Quito.
9. Femmer, H. (2013). Reviewing Natural Language Requirements with Requirements Smells – A Research Proposal. *Research Gate*, 1-8.
10. Gómez Fuentes, M. d. (2011). *Análisis de Requerimientos*. México D.F.: Universidad Autónoma Metropolitana.
11. Herrera, L. J. (2003). *Monografías*. Obtenido de Monografías: <http://www.monografias.com/trabajos6/resof/resof.shtml>

12. Hinojosa, C., Raura, G., Fonseca, E., & Dieste, O. (2015). La Gestión del Conocimiento Aplicada en la Ingeniería de Requisitos: Un Caso de Estudio en Ecuador. *18th Workshop*, 2-4.
13. Ibañez, J. (2010). *Gestión de proyectos*. Obtenido de Project Management: <http://blog.masterinprojectmanagement.net/gestion-de-requerimientos-vi-establecer-un-plan-de-la-gestion-de-requisitos/>
14. IEEE 830. (1998). *IEEE Recommended Practice for Software Requirements Specifications*. New York: IEEE.
15. ISO/IEC/IEEE 29148. (2011). *Systems and software engineering — Life cycle processes — Requirements engineering*. New York: IEEE.
16. Laguna, M. Á. (2014). *Requisitos*. Obtenido de Ingeniería del software I: <https://www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf>
17. Laredo, A. (2011). Indicadores de calidad en el desarrollo de Software. *Revista de Investigación de Sistemas e Informática*, 20-23.
18. López Echeverry, A. M., Cabrera, C., & Valencia Ayala, L. E. (2008). INTRODUCCIÓN A LA CALIDAD DE SOFTWARE . *Scientia et Technica*.
19. López Echeverry, A. M., Cabrera, C., & Valencia Ayala, L. E. (2008). Introducción a la calidad del software. *Scientia et Technica*, 328.
20. NASA. (2014). *NASA Software Engineering Requirements*. NASA Procedural Requirements.
21. Pfleeger, S. L. (2002). *Ingeniería de Software, Teoría y Práctica*. Pearson Education.
22. Pressman, R., & Maxim, B. (2015). *Software Engineering*. New York: McGraw Hill.
23. Quiroga, J. P. (2015). *Requerimientos Funcionales y No Funcionales*. Obtenido de Requerimientos: <http://www.electrohuila.com.co/Portals/0/UpDocuments/0b530417-2986-450e-bd92-34928a11e2f5.pdf>
24. Rosique, M. F., Jiménez, M., & Sánchez, P. (2010). Evaluación de herramientas de gestión de requisitos. *II Jornadas de Introducción a la Investigación de la UPCT*, 47-48.

25. Sanguino, M. (2014). Formulación de un marco de referencia para la personalización de requerimientos en una fábrica de desarrollo de software bancario para mejorar la atención de las entidades financieras.
26. Simbaña Saransig, J., Simbaña Quinsasamin, G., Hinojosa Raza, C., & Ron Egas, M. (2015). Prácticas de Ingeniería de Requisitos en las Empresas de Desarrollo de Software, en la Ciudad de Quito - Ecuador. *GEEKS*, 5.
27. Sommerville, I. (2011). *SOFTWARE ENGINEERING*. Boston: Pearson Education, Inc.
28. Uwizeyemungu, S., & Raymond, L. (2012). Impact of an ERP system's capabilities upon the realisation of its business value: a resource-based perspective. *Springer Science+Business Media*, 73-87.
29. Vo, H. (2009). *Software Engineering*. Obtenido de Software Engineering: <http://cnx.org/content/col110790/1.1/>
30. Wiegers, K., & Beatty, J. (2013). *Software Requirements, Third Edition*. Washington: Microsoft Press.
31. Yeh, D. R. (1990). Forward to System and Software Requirements Engineering. *IEEE Computer Society Press Tutorial*. M. Dorfman, and R.H Thayer.

## Anexo A: Glosario de Términos Técnicos

**Análisis de requisitos:** fase de un proyecto software donde se efectúa un conjunto de actividades con el propósito de comprender el problema planteado con todo detalle y se enuncia el resultado de dicho proceso de comprensión en forma de un planteamiento técnico del problema que se denomina especificación técnica.

**Caso de uso:** herramienta que modela los servicios que ofrece el sistema a través de un diálogo entre un actor y el sistema. Acciones del usuario y reacciones del sistema. “Un caso de uso es una secuencia de transacciones proporcionadas por el sistema que proporcionan un resultado mensurable de valores a un actor particular”

**DFD:** Diagrama de Flujo de Datos. Es un diagrama en forma de red que representa el flujo de datos y las transformaciones que se aplican sobre ellos al moverse desde la entrada hasta la salida del sistema. Se utiliza para modelizar las funciones del sistema y los datos que fluyen entre ellas a distintos niveles de abstracción. El sistema, por tanto, se modela mediante un conjunto de DFS nivelados en el que los niveles superiores definen las funciones del sistema de forma general u los niveles inferiores definen estas funciones en niveles mejor detallados.

**Especificación de requisitos de Software (ERS):** proceso de redacción o registro de los requisitos. Se pueden utilizar tanto el lenguaje natural, como modelos gráficos.

**Ingeniería del Software:** es el establecimiento y uso de principios de ingeniería robustos, orientados a obtener software económico que sea fiable y funcione de manera eficiente sobre máquinas reales, mediante la aplicación de los siguientes elementos y actividades: los métodos, la planificación y estimación de proyectos, el análisis de los requisitos del sistema y del software, el diseño de estructuras de datos, la arquitectura de programas y procedimientos algorítmicos, la codificación, las pruebas, la instalación y el mantenimiento, las herramientas y los procedimientos.

**Ingeniería de Requerimientos (IR):** El proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un sistema.

**Proyecto software:** conjunto de actividades coordinadas cronológicamente para alcanzar un subconjunto de objetivos a partir de la definición de un subconjunto de necesidades.

**Requisitos:** Descripción de las necesidades o deseos de un producto.

**Software:** es el conjunto de programas, procedimientos y documentación asociada a la operación de un sistema informático.

**UML:** Unified Modeling Language. Lenguaje de programación gráfico para el modelado de proyectos software orientados a objetos.

**Validación de los requisitos:** Proceso de confirmación, por parte de los usuarios o del cliente, de que los requisitos especificados son válidos, consistentes y completos.

**Verificación de los requisitos:** Proceso de comprobación de que los requisitos realmente cubren las necesidades del cliente.

## Anexo B: Otras versiones del IEEE 830

Dicha versión del estándar IEEE 830 propone tres posibles modelos para la sección 3 de una Especificación de Requisitos de Software (Chalmeta, 2000).

### Modelo 1

- 3. Requerimientos específicos
  - 3.1. Requerimientos funcionales
    - 3.1.1. Requerimiento funcional 1.
      - 3.1.1.1. Introducción
      - 3.1.1.2. Entradas
      - 3.1.1.3. Proceso
      - 3.1.1.4. Salidas
    - 3.1.2. Requerimiento funcional 2.
    - .....
    - 3.1.n. Requerimiento funcional n
  - 3.2. Requerimientos de interfaces externos.
    - 3.2.1. Interfaces de usuario
    - 3.2.2. Interfaces hardware
    - 3.2.3. Interfaces software
    - 3.2.4. Interfaces de comunicaciones
  - 3.3. Requerimientos de eficiencia
  - 3.4. Restricciones de diseño
    - 3.4.1. Estándares cumplidos
    - 3.4.2. Limitaciones Hardware
  - .....
  - 3.5. Atributos
    - 3.5.1. Seguridad
    - 3.5.2. Mantenimiento
  - .....
  - 3.6. Otros requerimientos
    - 3.6.1. Bases de Datos
    - 3.6.2. Operaciones
    - 3.6.3. Requerimientos de adaptación a situaciones
  - .....

## Modelo 2

- 3. Requerimientos específicos
  - 3.1. Requerimientos funcionales
    - 3.1.1. Requerimiento funcional 1.
      - 3.1.1.1. Especificación.
        - 3.1.1.1.1. Introducción
        - 3.1.1.1.2. Entradas
        - 3.1.1.1.3. Proceso
        - 3.1.1.1.4. Salidas
      - 3.1.1.2. Requerimientos de interfaces externos
        - 3.1.1.2.1. Interfaces de usuario
        - 3.1.1.2.2. Interfaces hardware
        - 3.1.1.2.3. Interfaces software
        - 3.1.1.2.4. Interfaces de comunicaciones
    - 3.1.2. Requerimiento funcional 2.  
.....
    - 3.1.n .Requerimiento funcional n.
  - 3.2. Requerimientos de eficiencia.
  - 3.3. Restricciones de diseño.
    - 3.3.1. Estándares cumplidos
    - 3.3.2. Limitaciones hardware
    - .....
  - 3.4. Atributos.
    - 3.4.1. Seguridad.
    - 3.4.2. Mantenimiento.
    - .....
  - 3.5. Otros requerimientos.
    - 3.5.1. Bases de Datos.
    - 3.5.2. Operaciones.
    - 3.5.3. Requerimientos de adaptación a situaciones.
    - .....

## Modelo 3

- 3. Requerimientos específicos.
  - 3.1. Requerimientos funcionales.
    - 3.1.1. Requerimiento funcional 1
      - 3.1.1.1. Introducción
      - 3.1.1.2. Entradas
      - 3.1.1.3. Proceso
      - 3.1.1.4. Salidas
      - 3.1.1.5. Requerimientos de eficiencia
      - 3.1.1.6. Restricciones de diseño
        - 3.1.1.6.1. Estándares cumplidos
        - 3.1.1.6.2. Limitaciones hardware
      - 3.1.1.7. Atributos-
        - 3.1.1.7.1. Seguridad
        - 3.1.1.7.2. Mantenimiento
      - .....
      - 3.1.1.8. Otros requerimientos
        - 3.1.1.8.1. Bases de Datos
        - 3.1.1.8.2. Operaciones
        - 3.1.1.8.3. Requerimientos de adaptación a situaciones
      - .....
    - 3.1.2. Requerimiento funcional 2.
      - .....
    - 3.1.n. Requerimiento funcional n.
      - .....
  - 3.2. Requerimientos de interfaces externos.
    - 3.2.1. Interfaces de usuario
      - 3.2.1.1. Requerimientos de eficiencia.
      - 3.2.1.2. Restricciones de diseño.
        - 3.2.1.2.1. Estándares cumplidos
        - 3.2.1.2.2. Limitaciones hardware
      - 3.2.1.3. Atributos.
        - 3.2.1.3.1. Seguridad
        - 3.2.1.3.2. Mantenimiento
      - .....
      - 3.2.1.4. Otros requerimientos
        - 3.2.1.4.1. Bases de datos
        - 3.2.1.4.2. Operaciones
        - 3.2.1.4.3. Requerimientos de adaptación a situaciones
      - .....
    - 3.2.2. Interfaces hardware.
    - 3.2.3. Interfaces software.
    - 3.2.4. Interfaces de comunicaciones.

## Anexo C: Ejemplo del documento de especificación de requerimientos

### Documento de especificación de Requerimientos de Software

**Para el proyecto** <Proyecto>

**Versión:** <Nro. De versión>

**Realizado por** <Autor>

<Organización>

<Fecha de creación>

### INDICE

#### Historial de revisión

Autor	Fecha	Razones del cambio	Versión

**Acta de Constitución del Proyecto:** <Nombre del proyecto>

#### Introducción

La introducción de la Especificación de requisitos de software (ERS) debe proporcionar una vista general de la ERS. Debe incluir el objetivo, el alcance, las definiciones y acrónimos, las referencias, y la vista general de la ERS.

#### Propósito

- Propósito del documento.
- Audiencia a la que va dirigido.

#### Alcance

- Identificación del producto(s) a desarrollar mediante un nombre

- Consistencia con definiciones similares de documentos de mayor nivel (ej. Descripción del sistema) que puedan existir

### Personal involucrado

Nombre	
Rol	
Categoría profesional	
Responsabilidades	
Información de contacto	
Aprobación	

Relación de personas involucradas en el desarrollo del sistema, con información de contacto. Esta información es útil para que el gestor del proyecto pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requisitos y validaciones de seguimiento.

### Definiciones, acrónimos y abreviaturas

Definición de todos los términos, abreviaturas y acrónimos necesarios para interpretar apropiadamente este documento. En ella se pueden indicar referencias a uno o más apéndices, o a otros documentos.

### Referencias

Referencia	Título	Ruta	Fecha	Autor
[Ref.]	[Título]	[Ruta]	[Fecha]	[Autor]

Relación completa de todos los documentos relacionados en la especificación de requisitos de software, identificando de cada documento el título, referencia (si procede), fecha y organización que lo proporciona.

## Resumen

- Descripción del contenido del resto del documento
- Explicación de la organización del documento

## Descripción general

### Perspectiva del producto

Indicar si es un producto independiente o parte de un sistema mayor. En el caso de tratarse de un producto que forma parte de un sistema mayor, un diagrama que sitúe el producto dentro del sistema e identifique sus conexiones para facilitar la comprensión.

### Funcionalidad del producto

Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en información de detalle.

En ocasiones la información de esta sección puede tomarse de un documento de especificación del sistema de mayor nivel (ej. Requisitos del sistema).

Las funcionalidades deben estar organizadas de manera que el cliente o cualquier interlocutor puedan entenderlo perfectamente. Para ello se pueden utilizar métodos textuales o gráficos.

### Características de los usuarios

Tipo de usuario	
Formación	
Habilidades	
Actividades	

Descripción de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica.

## Restricciones

Descripción de aquellas limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware o de sistema operativo.

## Suposiciones y dependencias

Descripción de aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo, una asunción puede ser que determinado sistema operativo está disponible para el hardware requerido. De hecho, si el sistema operativo no estuviera disponible, la ERS debería modificarse.

## Evolución previsible del sistema

Identificación de futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.

## Requisitos específicos

Esta es la sección más extensa e importante del documento.

Debe contener una lista detallada y completa de los requisitos que debe cumplir el sistema a desarrollar. El nivel de detalle de los requisitos debe ser el suficiente para que el equipo de desarrollo pueda diseñar un sistema que satisfaga los requisitos y los encargados de las pruebas puedan determinar si éstos se satisfacen.

Los requisitos se dispondrán en forma de listas numeradas para su identificación, seguimiento, trazabilidad y validación (ej. RF-1, RF-1.1, RF-2).

Para cada requisito debe completarse la siguiente tabla:

<b>Número de requisito</b>	
<b>Nombre de requisito</b>	
<b>Tipo</b>	<input type="checkbox"/> Requisito <input type="checkbox"/> Restricción
<b>Descripción</b>	
<b>Prioridad del requisito</b>	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
<b>Precondición</b>	

<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Post condición</b>		
<b>Restricciones</b>	<b>Paso</b>	<b>Acción</b>

La distribución de los párrafos que forman este punto puede diferir del propuesto en esta plantilla, si las características del sistema aconsejan otra distribución para ofrecer mayor claridad en la exposición.

### **Requisitos comunes de los interfaces**

Descripción detallada de todas las entradas y salidas del sistema.

### **Interfaces de usuario**

Describir los requisitos del interfaz de usuario para el producto. Esto puede estar en la forma de descripciones del texto o pantallas del interfaz. Por ejemplo, posiblemente el cliente ha especificado el estilo y los colores del producto. Describa exacto cómo el producto aparecerá a su usuario previsto.

### **Interfaces de hardware**

Especificar las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema. Se incluirán características de configuración.

### **Interfaces de software**

Indicar si hay que integrar el producto con otros productos de software. Para cada producto de software debe especificarse lo siguiente:

- Descripción del producto software utilizado
- Propósito del interfaz
- Definición del interfaz: contenido y formato

## **Interfaces de comunicación**

Describir los requisitos de interfaces de comunicación si hay comunicaciones con otros sistemas y cuáles son los protocolos de comunicación.

## **Requisitos funcionales**

Definición de acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados. En ellas se incluye:

- Comprobación de validez de las entradas
- Secuencia exacta de operaciones
- Respuesta a situaciones anormales (desbordamientos, comunicaciones, recuperación de errores)
- Parámetros
- Generación de salidas
- Relaciones entre entradas y salidas (secuencias de entradas y salidas, fórmulas para la conversión de información)
- Especificación de los requisitos lógicos para la información que será almacenada en base de datos (tipo de información, requerido)

Los requisitos funcionales pueden ser divididos en sub-secciones.

Requisito funcional 1

Requisito funcional 2

Requisito funcional 3

Requisito funcional n

## **Requisitos no funcionales**

### **Requisitos de rendimiento**

Especificación de los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema.

Todos estos requisitos deben ser medibles. Por ejemplo, indicando “el 95% de las transacciones deben realizarse en menos de 1 segundo”, en lugar de “los usuarios no deben esperar a que se complete la transacción”.

### **Seguridad**

Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requisitos pueden especificar:

- Empleo de técnicas criptográficas.
- Registro de ficheros con “logs” de actividad.
- Asignación de determinadas funcionalidades a determinados módulos.
- Restricciones de comunicación entre determinados módulos.
- Comprobaciones de integridad de información crítica.

### **Fiabilidad**

Especificación de los factores de fiabilidad necesaria del sistema. Esto se expresa generalmente como el tiempo entre los incidentes permisibles, o el total de incidentes permisible.

### **Disponibilidad**

Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en % de tiempo en los que el software tiene que mostrar disponibilidad.

### **Mantenibilidad**

Identificación del tipo de mantenimiento necesario del sistema. Especificación de quien debe realizar las tareas de mantenimiento, por ejemplo, usuarios, o un desarrollador. Especificación de cuándo debe realizarse las tareas de mantenimiento. Por ejemplo, generación de estadísticas de acceso semanal y mensual.

### **Portabilidad**

Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos. Pueden incluirse:

- Porcentaje de componentes dependientes del servidor.
- Porcentaje de código dependiente del servidor.
- Uso de un determinado lenguaje por su portabilidad.
- Uso de un determinado compilador o plataforma de desarrollo.
- Uso de un determinado sistema operativo.

### **Otros requisitos**

Cualquier otro requisito que no encaje en ninguna de las secciones anteriores. Por ejemplo:

- Requisitos culturales y políticos
- Requisitos Legales

### **Apéndices**

Pueden contener todo tipo de información relevante para la ERS pero que, propiamente, no forme parte de la ERS.

**Anexo D: Encuesta I**

1. ¿Qué persona realiza los requerimientos de un nuevo sistema?
  - a. Usted
  - b. Usted y un equipo de sistemas ajeno a la institución
  - c. Usted y un equipo de sistemas de la institución
  
2. ¿Recibe ayuda de un equipo de especialistas para realizar los requerimientos?
  - a. Si
  - b. No
  
3. ¿Utiliza algún formato para la elaboración de requerimientos?
  - a. Si
  - b. No
  
4. ¿El formato es fácil de comprender y de utilizar?
  - a. Si
  - b. No
  
5. ¿Qué tiempo le lleva aproximadamente elaborar los requerimientos?
  - a. 1 día
  - b. 3 a 5 días
  - c. Mayor a 5 días
  
6. ¿Solicita cambios a los requerimientos iniciales, cuando el sistema está en fase de pruebas?
  - a. Si
  - b. No
  
7. ¿Con qué frecuencia solicita los cambios a los requerimientos iniciales?
  - a. 1 a 5 cambios
  - b. 6 a 10 cambios
  - c. Mayores a 10
  
8. ¿Cuándo el sistema se encuentra en fase de pruebas, usted observa que el funcionamiento es el esperado de acuerdo con lo que usted solicitó?
  - a. Si
  - b. No
  
9. ¿Qué sugerencia puede dar al momento que va a realizar los requerimientos del sistema?

**Anexo E: Encuesta II**

1. ¿Qué persona realiza los requerimientos de un nuevo sistema?
  - a. Usted
  - b. Usted y un equipo de sistemas ajeno a la institución
  - c. Usted y un equipo de sistemas de la institución
  
2. ¿Recibe ayuda de un equipo de especialistas para realizar los requerimientos?
  - a. Si
  - b. No
  
3. ¿Utiliza algún formato para la elaboración de requerimientos?
  - a. Si
  - b. No
  
4. ¿Cuál de los dos estándares a su parecer es mejor para utilizar?
  - a. IEEE-830
  - b. ISO/IEC/IEEE-29148
  
5. ¿Cuál de los estándares reduce el tiempo de elaboración del documento de requerimientos?
  - a. IEEE-830
  - b. ISO/IEC/IEEE-29148
  
6. ¿Cuál de los dos estándares se adapta mejor al área que dirige?
  - a. IEEE-830
  - b. ISO/IEC/IEEE-29148
  
7. ¿Qué porcentaje de aceptación tiene del estándar IEEE-830?
  - a. 0-30%
  - b. 31-60%
  - c. 61-100%
  
8. ¿Qué porcentaje de aceptación tiene del estándar ISO/IEC/IEEE-29148?
  - a. 0-30%
  - b. 31-60%
  - c. 61-100%
  
9. ¿Cuál de los dos estándares desearía utilizar a futuro con la gestión de requerimientos?
  - a. IEEE-830
  - b. ISO/IEC/IEEE-29148

**Anexo F: Encuesta III**

1. ¿Está de acuerdo con el uso del software para gestión de requerimientos?
  - a. Si
  - b. No
  
2. ¿Qué porcentaje de aceptación tiene sobre el software de gestión de requerimientos?
  - a. 0-30%
  - b. 31-60%
  - c. 61-100%

## Anexo G: Documento de Especificación de Requerimientos del Prototipo

### Documento de especificación de Requerimientos de Software

Para el proyecto <Arjé Gestión de Requerimientos>

Versión: <1.0>

Realizado por <Eric Guzmán>

<12/Marzo/2017>

Historial de revisión

Autor	Fecha	Razones del cambio	Versión
Eric Guzmán	12/03/2017	Documento inicial	1.0

### Introducción

En función de la encuesta realizada y de los resultados obtenidos en el capítulo anterior se puede justificar el desarrollo de un prototipo de software para la gestión de requerimientos funcionales para el desarrollo de aplicaciones en la Cooperativa Financiera Atuntaqui.

Para el desarrollo del prototipo se utilizó la metodología SCRUM, de igual manera se utilizó PHP como lenguaje de desarrollo, MySQL como base de datos, Apache como servidor Web y jQuery para dinamizar las páginas y mejorar la interacción del usuario con el sistema.

El prototipo consta de un ingreso con usuario y contraseña; se puede subir el documento de especificación de requerimientos basados en el estándar IEEE-830 y de igual manera se puede ingresar los requerimientos en donde se puede observar el porcentaje de avance de este y el estado actual del requerimiento.

De esta manera se puede conocer el avance de los requerimientos por cada proyecto, además permite tener una comprensión rápida de la estructura y funcionalidad del proyecto.

## Propósito

El prototipo de nombre Arjé permite la gestión de documentos de requerimientos funcionales de un proyecto de software. Ayuda al programador a tener distribuido de manera ordenada el documento de gestión de requerimientos y los casos de uso con el historial de cambios los cuales se puede visualizar directamente en la aplicación.

## Alcance

- Permite la creación de proyectos.
- Solo permite la subida de archivos en formato PDF.
- Permite visualizar que etapa no tiene actualizado un documento si alguno de una etapa anterior ha sido modificado.
- Permite el ingreso mediante la autenticación de un usuario y contraseña.
- Las pistas de auditoría permiten conocer el uso del sistema por parte de los usuarios.
- Funciona localmente en la intranet.
- Es un sistema autónomo e independiente.

## Personal involucrado

Nombre	Eric Guzmán
Rol	Desarrollador, Tester
Categoría profesional	Ingeniero en Sistemas
Responsabilidades	Desarrollo del producto, Usuario de pruebas, Gestor de requerimientos
Información de contacto	eguzman@atuntaqui.fin.ec
Aprobación	Si

## Definiciones, acrónimos y abreviaturas

- Arjé: nombre del prototipo de gestión de requerimientos.

- ERS: Documento de Especificación de Requerimientos.

## **Referencias**

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

## **Resumen**

Los siguientes ítems ofrecen una descripción general del proyecto a desarrollar, permite conocer la estructura del sistema y de todos los componentes que el estándar IEEE-830 ofrece para gestionar los requerimientos iniciales del prototipo con nombre Arjé.

Se detalla de igual manera los requerimientos funcionales y las posibles interfaces de usuario del producto a ser desarrollado. Los siguientes ítems contienen especificaciones técnicas para los desarrolladores y especificaciones en lenguaje natural para el cliente.

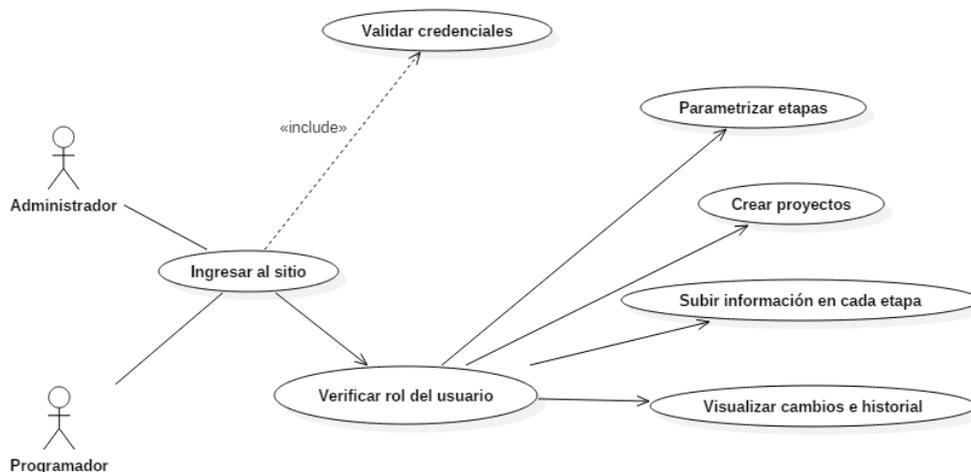
## **Descripción general**

### **Perspectiva del producto**

El software Arjé ha sido diseñado para su desempeño y funcionalidad en la Web, con un lenguaje liviano como PHP y una base de datos como MySQL.

### **Funcionalidad del producto**

En el siguiente gráfico de casos de uso, se detalla la funcionalidad del sistema, los actores involucrados y las actividades que realiza cada actor de acuerdo con el rol asignado.



### Características de los usuarios

Tipo de usuario	Administrador
Formación	Técnica
Habilidades	Gestión, administración
Actividades	Parametrizar las opciones de cada etapa. Ingresar proyectos, ingresar documentación, visualizar cambios e historiales de versión de documentos.

Tipo de usuario	Programador
Formación	Técnica
Habilidades	Uso frecuente
Actividades	Ingresar proyectos, ingresar documentación, visualizar cambios e historiales de versión de documentos.

### Restricciones

- La limitación actual es a los documentos a subir que son en formato PDF.

### Suposiciones y dependencias

No existe ninguna información para esta sección ya que el funcionamiento del sistema es de manera independiente.

### Evolución previsible del sistema

No se contemplan.

### Requisitos específicos

<b>Número de requisito</b>	RF-1	
<b>Nombre de requisito</b>	Autenticación y autorización	
<b>Tipo</b>	<input type="checkbox"/> Requisito <input type="checkbox"/> Restricción	
<b>Descripción</b>	Realizar la autenticación y autorización de acuerdo con el usuario que ingresa en la caja de texto; validar la contraseña y mostrar el mensaje “Usuario o contraseña incorrecta” en caso de credenciales erróneas.	
<b>Prioridad del requisito</b>	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional	
<b>Precondición</b>	Ingresar al sitio	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
	1	Ingresar el usuario y contraseña en las cajas de texto.
	2	Verificar el rol del usuario para habilitar las opciones de administrador.
	3	Verificar la clave almacenada en formato md5 en la base de datos.
<b>Post condición</b>	El sistema muestra la interfaz principal del sistema.	
<b>Restricciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el usuario o contraseña es erróneo mostrar el siguiente mensaje “Usuario o contraseña incorrecta”.
<b>Número de requisito</b>	RF-2	

<b>Nombre de requisito</b>	Parametrización de etapas	
<b>Tipo</b>	<input type="checkbox"/> Requisito <input type="checkbox"/> Restricción	
<b>Descripción</b>	Ingresar la información de las etapas que posteriormente almacenarán la información con los documentos en PDF.	
<b>Prioridad del requisito</b>	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional	
<b>Precondición</b>	Ingresar al sitio como Administrador.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
	1	Escoger la opción de parametrización.
	2	Ingresar la información.
	3	Se puede modificar la información.
	4	Se puede eliminar la información.
<b>Post condición</b>	El sistema muestra la interfaz para la parametrización.	
<b>Restricciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Se debe ingresar en orden ascendente las etapas

<b>Número de requisito</b>	RF-3	
<b>Nombre de requisito</b>	Ingreso de información en cada etapa	
<b>Tipo</b>	<input type="checkbox"/> Requisito <input type="checkbox"/> Restricción	
<b>Descripción</b>	Al escoger la etapa en el panel izquierdo de navegación, permite la opción de cargar los documentos en formato PDF	
<b>Prioridad del requisito</b>	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional	
<b>Precondición</b>	Ingresar al sitio como Administrado o programador.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>

	1	Escoger la etapa en el menú del panel izquierdo.
	2	Ingresar la descripción del cambio realizado. Colocar “Archivo inicial” para la primera carga. Especificar los cambios cuando haya modificaciones.
	3	Seleccionamos el documento a subir en formato PDF
	4	Al presionar el botón Enviar se debe almacenar la información y mostrar en la parte de historial de cambios.
	5	Mostrar un símbolo de advertencia en la etapa que ha sido modificada y que no tiene el documento actualizado.
<b>Post condición</b>	El sistema muestra la interfaz para cargar los archivos.	
<b>Restricciones</b>	<b>Paso</b>	<b>Acción</b>
	4	Mostrar un mensaje al momento de subir el archivo “Datos enviados correctamente”, si hubo un error “Hubo un inconveniente al enviar la información”.
<b>Número de requisito</b>	RF-4	
<b>Nombre de requisito</b>	Visualización de documentos e historial de cambios	
<b>Tipo</b>	<input type="checkbox"/> Requisito <input type="checkbox"/> Restricción	
<b>Descripción</b>	Al escoger la etapa en el panel izquierdo de navegación, permite la opción de visualizar el historial de cambios con el usuario, fecha y detalle del cambio. Al hacer click en el vínculo de cada cambio se muestra en otra ventana el documento PDF y con la descripción del cambio a detalle.	
<b>Prioridad del requisito</b>	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional	
<b>Precondición</b>	Ingresar al sitio como Administrado o programador.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
	1	Escoger la etapa en el menú del panel izquierdo.
	2	Visualizar en el panel de cambios las modificaciones realizadas.

	3	Al hacer click en el vínculo de cada cambio se muestra el archivo en formato PDF para visualizar los cambios realizados.
<b>Post condición</b>	El sistema muestra la interfaz para visualizar los cambios a los documentos.	
<b>Restricciones</b>	<b>Paso</b>	<b>Acción</b>

### Requisitos comunes de los interfaces

- Diseño sencillo
- Colores suaves a la vista

### Interfaces de usuario

La interfaz utiliza un diseño agradable y de fácil manejo para el usuario en un entorno web y con colores suaves para la vista.

Interfaz inicial: el proyecto muestra una interfaz amigable con la opción de ingresar un usuario y la contraseña para su acceso.

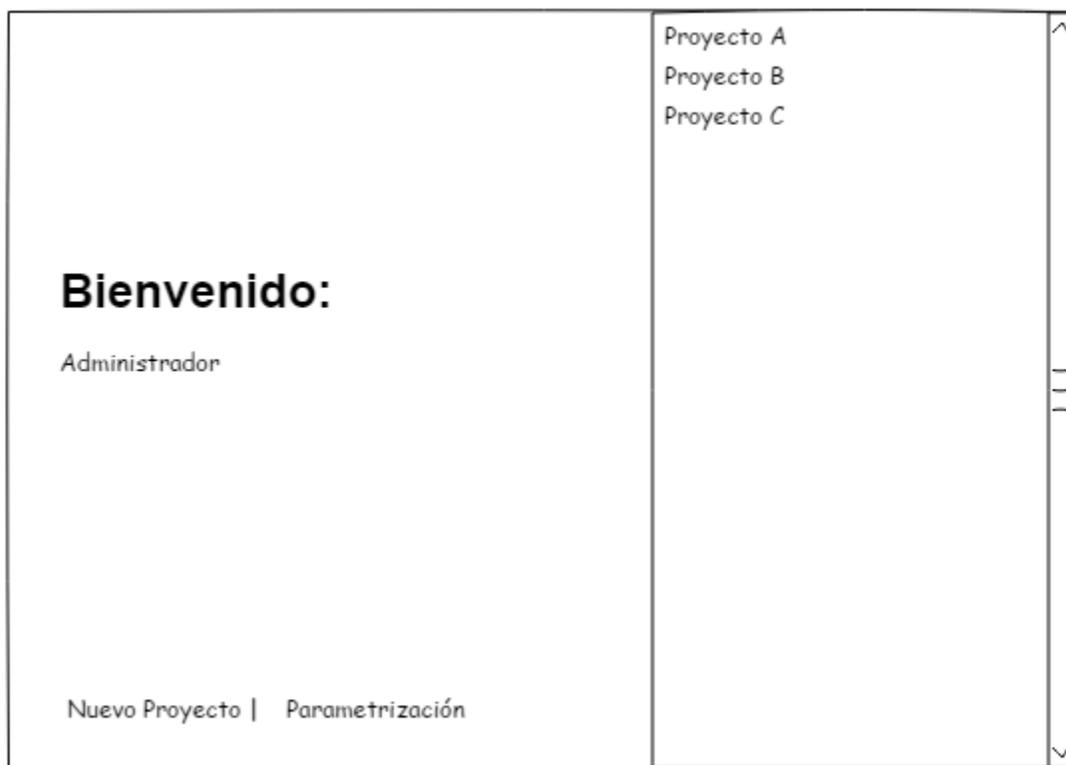
## Arjé

Gestion de Requerimientos

Usuario:

Contraseña:

Interfaz principal: en esta interfaz se muestran las opciones de acuerdo con el rol del usuario ingresado. Algunas opciones de parametrización solo están habilitadas para el administrador del sitio. En la parte derecha se visualiza el listado de proyectos creados.



Interfaz de parametrización (Administrador): aquí se puede parametrizar las opciones a cada etapa para el ingreso de información.

Administrador

### Parametrización de Etapas

	Etapas	Acciones	
1	Documento de Especificación de Requerimientos	Modificar	Eliminar
2	Casos de Uso	Modificar	Eliminar

Regresar

Interfaz de ingreso de proyectos: se muestra las cajas de texto para el ingreso de la información de los proyectos.

Administrador

### Bienvenido:

Nombre del proyecto:

Responsable

Detalle

Apoyo

Steve Jobs  
 Bill Gates  
 John Smith

Regresar

Interfaz de ingreso de documentación: en esta interfaz se puede cargar los documentos por cada opción de etapa previamente parametrizado.

Descripción de proyecto A

**Proyecto A**

C:/ERS.pdf

Etapas	
1	Documento de Especificación de Requerimientos 
2	Casos de Uso

Cambios realizados

Historial de cambios

Id	Archivo	Usuario	Fecha
1	ERS 1	eguzman	02/02/2017
2	ERS 2	tandrade	03/25/2017
3	ERS 3	jsmith	03/27/2017

Interfaz para visualización de cambios: en esta interfaz se puede visualizar el contenido del documento PDF previamente cargado y observar los cambios, qué persona las realizó y la fecha.

Se muestra la información del archivo guardado PDF

543 x 250

Cambios realizados

## Interfaces de hardware

Se debe contar con un servidor Web con soporte para PHP y con conectividad 24/7. Además, se debe tener en consideración lo siguiente por parte del equipo que va a alojar el servidor Web:

- Adaptadores de red.
- Procesador i5 o superior.
- Memoria de 4GB o superior.
- Disco Duro de 256GB o superior

### **Interfaces de software**

- Servidor Web con soporte para PHP.
- PHP.
- MySQL.
- jQuery.

### **Interfaces de comunicación**

El puerto para la comunicación con la base de datos debe estar libre y sin conflicto con otros puertos. Además de la conectividad con los protocolos de internet preestablecidos y el puerto del servidor web por ejemplo 80, 8080.

### **Requisitos no funcionales**

- El sistema debe presentar una interfaz amigable y fácil de manejar, con nombres claros en las etiquetas que faciliten la comprensión del sitio al usuario.
- Lo colores y el diseño de la interfaz debe ser agradable a la vista del usuario, empleando colores cálidos y suaves para que no produzca cansancio en su uso.

### **Requisitos de rendimiento**

Como es una aplicación independiente y que no maneja escenarios críticos en producción que afecten a la productividad de la empresa, los valores de rendimiento se miden por la velocidad de

la Intranet, la velocidad de respuesta del servidor web y de la base de datos, que por defecto son normales y aceptables para un óptimo desempeño de la aplicación.

### Seguridad

- Encriptación de la clave con md5.
- Autenticación y autorización mediante usuario y contraseña.
- Pistas de auditoría.

### Fiabilidad

Proceso	Cumple el funcionamiento
Validación de clave y autenticación de usuario	✓
Encriptación de clave	✓
Mensajes de advertencia	✓
Parametrización de etapas	✓
Ingreso de proyectos	✓
Ingreso y carga de archivos PDF	✓
Validación de archivos PDF	✓
Visualización de PDF e historial de archivos	✓

### Disponibilidad

El sistema está alojado en un servidor de producción en la institución con servicio 24/7.

### Mantenibilidad

El sistema es fácil de mantener ya que los archivos principales que interactúan con la base de datos están en archivos de clase con acrónimos de las funciones fáciles de comprender y mantener en los casos que se requiera cambios adicionales.

**Portabilidad**

El sistema es fácil de transportar ya que, al ser una aplicación Web, se puede trasladar el código fuente hacia un servidor que soporte el lenguaje PHP y se conecte a una base de datos MySQL.

**Otros requisitos**

No aplica.