Polo del Conocimiento

Pol. Con. (Edición núm. 69) Vol. 7, No 4
Abril 2022, pp. 1470-1501
ISSN: 2550 - 682X
DOI: 10.23857/pc.v7i4.3902

Gestión de requerimientos en proyectos de desarrollo de software bajo la metodología de Proceso Racional Unificado. Caso: Cooperativa de Ahorro y Crédito Jardín Azuayo

Requirements management in software development projects under the Unified Rational Process methodology. Case: Jardín Azuayo Savings and Credit Cooperative

Gerenciamento de requisitos em projetos de desenvolvimento de software sob a metodologia Unified Rational Process. Caso: Cooperativa de Poupança e Crédito Jardín Azuayo

Lorena Elizabeth Cajamarca-Cajamarca [I]
lorenacajamarca21@gmail.com
https://orcid.org/0000-0002-6971-818X

Miguel Ángel Zúñiga-Prieto [II]
miguel.zunigap@ucuenca.edu.ec
https://orcid.org/0000-0001-9369-1813

Correspondencia: lorenacajamarca21@gmail.com

Ciencias Económicas y Empresariales
Artículo de Investigación

*__Recibido:__ 25 de febrero de 2022 *__Aceptado:__ 30 de marzo de 2022 * __Publicado:__ 13 abril de 2022

I. Ingeniera de empresas, Mención talento humano y Marketing, Analista de gestión de servicios físicos y virtuales, COAC Jardín Azuayo, Universidad de Cuenca, Cuenca, Ecuador.
II. Doctor en Informática, Magister Scientiarium en Administración Tecnológica, Diplomado en Dirección de Proyectos, Diplomado en Pedagogías Innovadoras, Director del Departamento de Ciencias de la Computación, Docente e investigador en la Facultad de Ingeniería de la Universidad de Cuenca, Coordinador de proyectos de investigación locales y en asociación con universidades europeas, Gestor de proyectos de desarrollo y despliegue de software en el Ecuador y países de Latinoamérica, Universidad de Cuenca, Cuenca, Ecuador.

Resumen

Hoy en día se hacen necesarias propuestas metodológicas de ingeniería de requerimientos, que impongan un proceso bien disciplinado sobre el desarrollo de software. El presente artículo analiza la gestión de requerimientos en proyectos de desarrollo de software bajo la metodología de Proceso Racional Unificado (RUP). Inicia explicando la metodología RUP de acuerdo a diferentes postulados teóricos, para luego adaptarla y validarla en la Cooperativa de Ahorro y Crédito Jardín Azuayo; caso de estudio de la presente investigación. Se trata de una investigación descriptiva de campo, donde se utilizó una metodología estructurada guía, basada en el modelo de transferencia tecnológica de Gorschek (2006). Los resultados muestran que la metodología de gestión de requerimiento para proyectos de desarrollo de software de la Cooperativa define un proceso que incluye seis actividades, a saber, analizar el problema, entender las necesidades de los interesados, definir el sistema, administrar el alcance del sistema, refinar la definición del sistema, y gestionar los requisitos cambiantes. Se concluye que, esta metodología provee un conjunto de tareas iterativas, guías para la ejecución y formatos que definen y describen la estructura y contenidos de los artefactos a generar, producto de la ejecución de las actividades; permitiendo un desarrollo de software más eficiente, predecible, y adaptable a las necesidades tanto de los prestadores del servicio como de los usuarios finales.

Palabras Clave: gestión de requerimientos; Proceso Racional Unificado; proyectos de desarrollo de software; Cooperativa de Ahorro y Crédito Jardín Azuayo.

Abstract

Nowadays, methodological proposals for requirements engineering are necessary, which impose a well-disciplined process on software development. This article analyzes the management of requirements in software development projects under the Rational Unified Process (RUP) methodology. It begins by explaining the RUP methodology according to different theoretical postulates, to later adapt and validate it in the Jardín Azuayo Savings and Credit Cooperative; case study of this research. This is a descriptive field research, where a structured guide methodology was used, based on the technology transfer model of Gorschek (2006). The results show that the requirement management methodology for software development projects of the Cooperative defines a process that includes six activities, namely, analyze the problem, understand the needs of the interested parties, define the system, manage the scope of the system, refine the system

definition, and manage changing requirements. It is concluded that this methodology provides a set of iterative tasks, guides for the execution and formats that define and describe the structure and contents of the artifacts to be generated, product of the execution of the activities; allowing a more efficient, predictable software development, and adaptable to the needs of both service providers and end users.

Key words: requirements management; Unified Rational Process; software development projects; Jardín Azuayo Savings and Credit Cooperative.

Resumo

Atualmente, são necessárias propostas metodológicas de engenharia de requisitos, que imponham um processo bem disciplinado no desenvolvimento de software. Este artigo analisa o gerenciamento de requisitos em projetos de desenvolvimento de software sob a metodologia Rational Unified Process (RUP). Começa explicando a metodologia RUP de acordo com diferentes postulados teóricos, para depois adaptá-la e validá-la na Cooperativa de Poupança e Crédito Jardín Azuayo; estudo de caso desta pesquisa. Trata-se de uma pesquisa de campo descritiva, onde foi utilizada uma metodologia de guia estruturada, baseada no modelo de transferência de tecnologia de Gorschek (2006). Os resultados mostram que a metodologia de gerenciamento de requisitos para projetos de desenvolvimento de software da Cooperativa define um processo que inclui seis atividades, a saber, analisar o problema, entender as necessidades das partes interessadas, definir o sistema, gerenciar o escopo do sistema, refinar o definição do sistema e gerenciar os requisitos em mudança. Conclui-se que esta metodologia fornece um conjunto de tarefas iterativas, guias de execução e formatos que definem e descrevem a estrutura e o conteúdo dos artefatos a serem gerados, produto da execução das atividades; permitindo um desenvolvimento de software mais eficiente, previsível e adaptável às necessidades tanto dos provedores de serviços quanto dos usuários finais.

Palavras-chave: gestão de requisitos; Processo Racional Unificado; projetos de desenvolvimento de software; Cooperativa de Poupança e Crédito Jardim Azuayo.

Introducción

Con el paso del tiempo, el desarrollo de software ha venido presentado falencias, sobre todo en la especificación de requerimientos de software; lo que se ha vuelto un proceso crítico ya que muchos proyectos fallan debido a requerimientos incorrectos. Por ello, los ingenieros de software se han visto obligados a contar con un profundo conocimiento del ambiente organizacional como una condición previa al desarrollo. Por esta razón, la ingeniería del software es la que más esfuerzos han realizado puesto que los errores de comprensión cometidos en esta etapa y la deficiente participación de todos los interesados crea ineficiencias y duplicaciones, lo que resulta en problemas de comunicación, causando retrabajo, retrasos y altos costos en los proyectos (Monsalve, et al, 2011).

Las instituciones financieras para poder ser competitivas dentro de un mundo de avances tecnológicos, deben mejorar su oferta digitalizando sus servicios y disminuyendo tiempo de atención de sus clientes. Dichas instituciones demandan servicios virtuales para disminuir la presencialidad en las oficinas, pero para ello se requiere ser eficiente en la construcción y producción de los requerimientos. Estos requerimientos se transforman en proyectos de desarrollo de software, para lo cual es necesario contar con una metodología de gestión adecuada. Esto es especialmente importante para la Cooperativa de Ahorro y Crédito Jardín Azuayo, cuyo objetivo estratégico es disponer de servicios financieros oportunos, accesibles e innovadores que satisfagan las necesidades de los socios (COAC Jardín Azuayo, 2018).

Hoy en día existen metodologías que soportan la gestión de requerimientos, pero no se preocupan por los detalles de dicha gestión. Por ejemplo, las metodologías ágiles buscan conseguir el producto de software utilizando la comunicación directa entre las personas que intervienen en el proceso, pues se enfocan en el diseño y desarrollo de software; por otro lado, las metodologías tradicionales o pesadas hacen hincapié en las especificaciones y el diseño detallado de la aplicación y buscan conseguir el producto de software por medio del orden y la documentación (Molpeceres, 2002).

Las metodologías tradicionales, según Cadavid, et al (2013), inician el desarrollo de un proyecto centrándose en la especificación de requisitos tratando de asegurar resultados de alta calidad, y de cumplir el plan de proyecto. Los llamados métodos pesados o tradicionales, se caracterizan por su rigidez metodológica y la exhaustiva documentación; dentro de estos se encuentra el Proceso Racional Unificado – RUP, por sus siglas en inglés, Rational Unified Process, basado en Unified Modeling Language (UML) para la ingeniería de sistemas y de software; por sus características,

RUP está enfocado para grandes proyectos de software y equipos de trabajo de amplio alcance (Parra, 2011); el cual, para definir los requerimientos, se centra en las especificaciones basadas en casos de uso y una documentación sólida (Hanssen, et al, 2005).

Esto lleva a considerar estudios como los propuestos por Sopingi, et al (2021); Arias (2012) y Jaramillo (2016), en los que se aplica el framework RUP, en entidades financieras y en el sector académico, rescatándose beneficios en el desarrollo de software. Cabe acotar que dichos trabajos, no proporcionan un soporte adecuado para la gestión de requerimientos de proyectos de desarrollo de software de la Cooperativa pues proponen actividades a ejecutar, pero no proveen las guías prácticas de estudio, ni las plantillas que permitan al equipo de proyectos conocer de manera clara los entregables a producir, su estructura, ni contenido. Además, de acuerdo con Hanssen (2005), también es importante conocer que para aplicar RUP a la gestión de requerimientos, esta debe ser adaptada, reducida y especializada al contexto de uso, ya que ningún proyecto de desarrollo de software es igual.

Con esta justificación, se decide adaptar RUP para su aplicación en la gestión de requerimientos de los proyectos de tipo estratégicos y operativos de mejoras de las cooperativas de ahorro y crédito, ya que los requerimientos de este tipo de proyectos deben ser especificados al inicio del proyecto y luego ser divididos en especificaciones más pequeñas que deriven en la entrega del sistema en partes funcionales.

Por lo mencionado en los párrafos anteriores, este estudio adapta el marco de trabajo RUP para la gestión de requerimientos de proyectos de desarrollo de software de la Cooperativa Jardín Azuayo, con el objetivo de involucrar a las partes interesadas y a partir de estas definir las funcionalidades del sistema. En esta adaptación de la metodología se proveerá un conjunto de tareas, guías para la ejecución y formatos que definen y describen la estructura y contenidos de los artefactos a generar, producto de la ejecución de las actividades.

Metodología

La presente investigación se trata de un estudio de campo, que inicialmente utiliza fuentes documentales, como libros, artículos científicos, reglamentos y metodologías de crédito, y normas de la Superintendencia de Economía popular y solidaria, entre otros, para luego recolectar datos a partir de diversos participantes del desarrollo y manejo del software en cuestión. Se utilizó una metodología estructurada guía, basada en el modelo de transferencia tecnológica de Gorschek

(2006), donde se incluyen actividades de evaluación y observación tanto en el ámbito académico como en la industria. Este modelo es un proceso iterativo de ocho actividades, a saber, análisis del problema, formulación del problema, revisión del estado del arte, solución candidata, entrenamiento, validación inicial, validación realista y liberación de la solución.

Por otro lado, para la validación de la metodología de gestión de requerimientos para proyectos de desarrollo de software en la Cooperativa de Ahorro y Crédito Jardín Azuayo, se realizó un cuasi-experimento es decir, un estudio empírico dirigido a probar hipótesis causales descriptivas acerca de causas manipulables en el cual los sujetos no son asignados aleatoriamente (Shadish, Cook, & Campbell, 2002). Se decidió realizar el cuasi-experimento por la ausencia de metodologías ampliamente aceptadas que soporten la gestión de requerimientos de proyectos de software en instituciones financieras con el involucramiento de todas las partes interesadas. El cuasi-experimento fue diseñado siguiendo las guías propuestas por Wohlin, et al (2012).

Proceso Racional Unificado - RUP

El Proceso Racional Unificado, es una metodología de desarrollo de software propuesta por IBM que está basado en componentes e interfaces bien definidas, y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos (Maida y Pacienzia, 2015). El RUP representa el conjunto de metodologías adaptables al contexto y necesidades de cada organización. Permite a los desarrolladores escoger y desplegar únicamente los componentes del proceso que sean necesarios y el uso de iteraciones acrecienta la calidad del software producido, en gran medida (Anwar, 2014).

El proceso unificado de desarrollo de software es un cúmulo de actividades empleadas para transformar los requisitos de un determinado usuario en software. Posee tres características clave: es iterativo e incremental, está centrado en la arquitectura, y dirigido a casos de uso (Pereira, 2011). Como complemento, Ahmad et al, 2011 mencionan que el RUP suministra una buena documentación de sus actividades y posee un enfoque iterativo; ambas características dan consistencia al tratamiento y desarrollo de aplicaciones, no obstante, para lograr un proceso de certificación adecuado, es importante la evaluación de la calidad al final de cada iteración, a fin de aumentar la calidad del software y minimizar o eliminar los defectos.

Esta metodología involucra cuatro fases: inicio, elaboración, construcción, transición y nueve disciplinas repetibles en cada uno de las fases. Las seis primeras son: modelado del negocio,

requisitos, análisis y diseño, implementación, pruebas, y despliegue; conocidas como flujos de trabajo del proceso; y otras tres: gestión del cambio y configuraciones, gestión del proyecto, y entorno; conocidas como flujos de trabajo de soporte (Addetla, et al, 2014).

De manera más detallada, el ciclo de vida de un proyecto con la metodología RUP está compuesto por las siguientes fases: concepción, elaboración, construcción y transición. En la concepción se precisa el alcance del sistema, los costos, la arquitectura candidata y la organización del proyecto en general. En la fase de elaboración se realizan actividades de refinación y validación de la arquitectura y sus componentes. La fase de construcción gestiona los recursos, el control de los procesos y se termina el desarrollo de componentes. La fase de transición comprende la ejecución de planes de implementación, se completan los manuales de usuario y técnicos, se integra el sistema y se ajusta según la validación del usuario. Estas cuatro fases pretenden asegurar la calidad cuando el producto le llegue al usuario (Nextel Engineering, 2011).

Aplicación de la metodología de gestión de requerimientos para proyectos de desarrollo de software en la Cooperativa de Ahorro y Crédito Jardín Azuayo

Este apartado presenta un ejemplo de aplicabilidad de la metodología RUP como soporte en la gestión de requerimientos para proyectos de desarrollo de software. En la siguiente sección se describe cómo la metodología ha sido adaptada, tanto en la ejecución de las actividades propuestas, como en la generación de artefactos producidos durante la gestión de requerimientos. Para ello, se trabajó con las siguientes actividades: analizar el problema; entender las necesidades de los interesados; definir el sistema; administrar el alcance del sistema; refinar la definición del sistema; y gestionar los requisitos cambiantes.

La Cooperativa de Ahorro y Crédito Jardín Azuayo dispone de un sistema de precalificación, sin embargo, presenta ciertos inconvenientes, como: dependencia con el proveedor; el sistema está desarrollado en una arquitectura diferente y antigua que no permite escalabilidad; y es un sistema inestable lo que provoca cierta insatisfacción en los usuarios finales. Ante esta situación la cooperativa tiene la necesidad de crear un sistema de precalificación que, por un lado, solvente la problemática presentada y también que cumpla con la función de precalificar a los socios de acuerdo con un nivel de riesgo; a partir del cual se obtenga una lista de requisitos y una tabla de amortización referencial para presentar al socio. El sistema debe interactuar con tres sistemas como: el Buró de crédito, el sistema transaccional Jasit y Comunicaliza, así como también, con el Asesor Cooperativo.

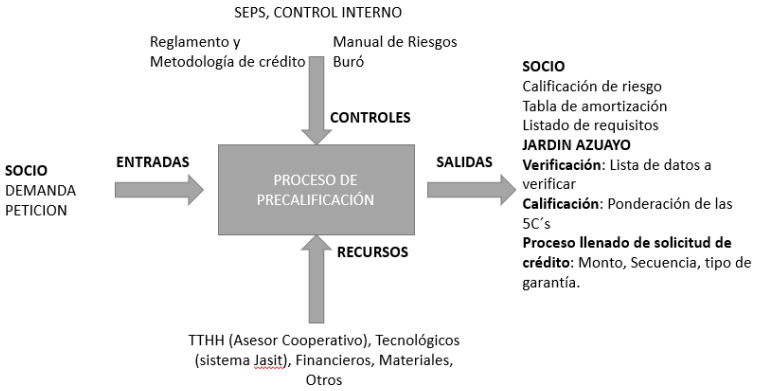
Actividades de la metodología de gestión de requerimientos:

1. Analizar el problema

El sistema de precalificación por ser nuevo, debe iniciar con el análisis del problema a fin de llegar a un acuerdo con las partes involucradas sobre la situación que se está resolviendo con el nuevo sistema. Las entradas utilizadas para esta actividad son los siguientes artefactos:

- P*roceso de servicio:* Este artefacto contiene únicamente la caracterización del proceso de precalificación en el que se especifican las entradas y salidas a obtener, así como los controles y los recursos que permitan cumplir el proceso. La *Ilustración 1* presenta la caracterización del proceso de la precalificación.

Ilustración 1. Caracterización del proceso de precalificación de crédito



Fuente: Elaboración propia (2022)

- *Solicitud de partes interesadas:* Este artefacto detalla las solicitudes de las partes interesadas con respecto al nuevo sistema de precalificación de crédito. A través de la revisión de diferentes documentos como: el reglamento de crédito, la metodología de crédito, las normas de la SEPS (Superintendencia de Economía popular y solidaria); el material del sistema de precalificación actual; y el material de una reunión virtual con las partes interesadas, se identificaron los requisitos presentados en la *Tabla 1*.

Tabla 1 Lista de requisitos

CÓDIGO	*LISTA DE REQUISITOS*
REQ1	El sistema de precalificación debe permitir generar un nuevo formulario de precalificación
REQ2	El sistema de precalificación debe permitir obtener una tabla de amortización y una lista de requisitos
REQ3	El sistema de precalificación debe permitir consultar, eliminar, editar la precalificación
REQ4	El sistema de precalificación debe permitir calcular un nivel de riesgo y calcular los modelos estadísticos
REQ5	El sistema de precalificación debe ser escalable, debe ser migrado a aplicación web, debe ser estable y disminuir las caídas del sistema y debe ser rápido
REQ6	El sistema de precalificación debe funcionar en las 73 oficinas con 3 asesores de crédito en promedio, con un crecimiento del 10% anual en usuarios finales
REQ7	El sistema de precalificación debe guardar todo cambio en la configuración para auditorias
REQ8	El sistema interactuara con los sistemas: Buró de crédito, Jasit y Comunicaliza.

Fuente: Elaboración propia (2022)

En la segunda actividad de análisis del problema, el analista del sistema elaboró la visión del sistema de precalificación del crédito, obteniendo como salida:

- La *Visión*, en el que se identificó el problema, las partes interesadas, impacto y beneficios, y se definieron las características principales del sistema. A continuación, en la *Tabla 2* se presenta un extracto del artefacto la visión de la precalificación de crédito.

Tabla 2. Extracto del artefacto La Visión

El problema	• Dependencia del Proveedor • Inestabilidad del sistema • No permite mayor escalabilidad • Tecnología antigua • Incumple la arquitectura de aplicación definida por la Cooperativa
Partes interesadas afectadas	• Usuarios finales (Asesor cooperativo), Analista de Gestión de la Calidad • Usuario administrador (Analista de Gestión de Requerimientos y Reclamos) • Usuario experto (Analista de Gestión de Servicios Físicos y Virtuales) • Especialista en producción • Director de proyecto • Patrocinador del proyecto • Arquitecto de software
Características principales	• El sistema de precalificación debe permitir crear un nuevo formulario de la precalificación • El sistema de precalificación debe permitir obtener una tabla de amortización y una la lista de requisitos para solicitar al socio • El sistema de precalificación debe permitir consultar, eliminar y editar la precalificación • Debe ser migrado a aplicación web, debe ser estable y debe disminuir las caídas del sistema, debe ser rápido, y ser escalable • debe funcionar en las 68 oficinas con 3 asesores de crédito en promedio, con un crecimiento del 10% anual en usuarios finales • El sistema de precalificación debe guardar todo cambio en la configuración para auditorias

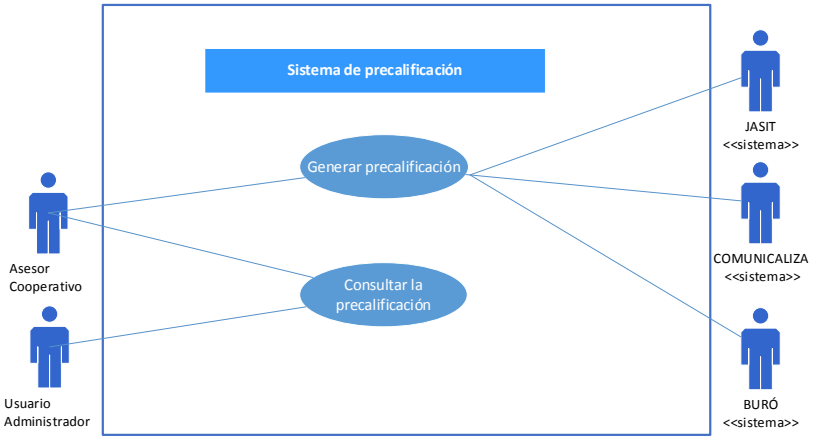
Fuente: Elaboración propia (2022)

Pol. Con. (Edición núm. 69) Vol. 7, No 4, Abril 2022, pp. 1470-1501, ISSN: 2550 - 682X

Por otra parte, en tercer lugar, el analista del sistema desarrolló la tarea *Buscar actores y Casos de uso* del sistema de precalificación para identificar los usuarios y sistemas que interactuarán con el mismo; obteniéndose como salidas:

- El modelo de caso de usos a un alto nivel, presentado en la *Ilustración 2*;
- Identificación de los actores presentados en la *Tabla 3*;
- Descripción preliminar de los casos de uso, se presenta un extracto del mismo en la *Tabla 4*;
- Especificaciones complementarias del sistema, presentadas en la *Tabla 5*.

Ilustración 2 Modelado de casos de uso



Fuente: Elaboración propia (2022)

Tabla 3. Actores del Sistema de Precalificación

ACTOR	*Descripción*
ASESOR COOPERATIVO	Es el usuario final quien utiliza el sistema para dar atención al socio en el servicio de crédito.
USUARIO ADMINISTRADOR	El Analista de Gestión de Requerimientos y Reclamos es el responsable de la administración y configuración del sistema.
SISTEMA JASIT	Es el sistema transaccional externo de crédito
SISTEMA BURÓ DE CRÉDITO	Es el sistema externo que permite consultar el historial crediticio del solicitante.
SISTEMA COMUNICALIZA	Es el sistema externo de consulta de datos personales de clientes

Fuente: Elaboración Propia (2022)

Tabla 4. Extracto de la Descripción de los Casos de Uso

CASOS DE USO	*Descripción*
GENERAR LA PRECALIFICACIÓN	Ingresar datos del solicitante para en base a la información generar una nueva precalificación, calcular el nivel de riesgo y por ende obtener el resultado final que es la lista de requisitos y la tabla de amortización
CONSULTAR PRECALIFICACIÓN	Este caso de uso permite consultar la precalificación siempre y cuando este dentro de los 30 días de validez. Al consultar el sistema recuperará todos los datos ingresados.

Tabla 5. Extracto de las especificaciones complementarias del sistema de precalificación

- El sistema de precalificación debe ser escalable
- El sistema de precalificación debe ser migrado a aplicación web
- El sistema de precalificación debe funcionar en las 68 oficinas con 3 asesores de crédito en promedio, con un crecimiento del 10% anual en usuarios finales
- El sistema de precalificación debe ser estable y debe disminuir las caídas del sistema
- El sistema de precalificación debe ser rápido y sin errores

Fuente: Elaboración propia (2022)

En la cuarta tarea el analista del sistema elaboró el *Plan de gestión de requisitos* en el artefacto con el nombre *Plan de gestión de requisitos y atributos,* con el objetivo de documentar los requisitos, tipos de requisitos y sus atributos y las directrices para la trazabilidad y la gestión de cambios de los requisitos del producto. Para un mejor entendimiento, a continuación, se presenta el contenido de este artefacto:

- En la *Tabla 6* se puede observar un extracto de los roles y responsabilidades de las partes interesadas que son fuentes de requisitos:

Tabla 6. Roles y Responsabilidades de las Fuentes de Requisitos

ROL	*Responsabilidad*
USUARIO EXPERTO	El Analista de Gestión de Servicios Físicos y Virtuales es la persona que tiene un amplio conocimiento del funcionamiento del sistema.
ARQUITECTO DE SOFTWARE	Responsable del análisis, diseño e implementación del sistema
USUARIO ADMINISTRADOR	El Analista de Gestión de Requerimientos y Reclamos es el responsable de la administración y configuración del sistema.
USUARIO FINAL	El Asesor Cooperativo quien utiliza el sistema para dar atención al socio en el servicio de crédito.

Fuente: Elaboración Propia (2022)

Pol. Con. (Edición núm. 69) Vol. 7, No 4, Abril 2022, pp. 1470-1501, ISSN: 2550 - 682X

- En la *Tabla 7* se pueden observar los atributos más importantes a ser asignados a los tipos requisitos identificados.

Tabla 7. Atributos de requisitos

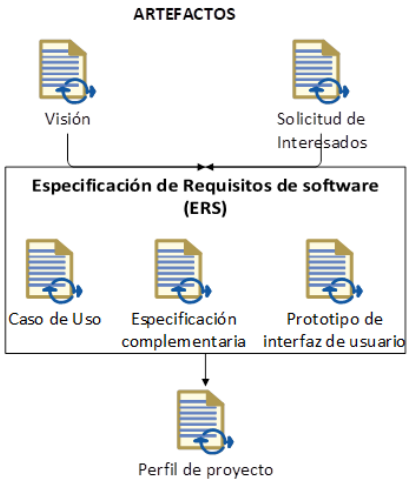
Los atributos para los tipos de requisitos: Solicitudes de los interesados, Visión, Casos de uso y Requisitos complementarios son los siguientes:

- **Estado:** Establecido después de que se haya analizado de acuerdo a los valores otorgados a los otros atributos y luego de que se haya negociado con el director de proyecto y arquitecto de software, pudiendo ser aprobado o rechazado.
- **Beneficio:** es el beneficio desde la perspectiva de las partes interesada, del dueño del producto. Se utiliza para administrar el alcance y determinar la prioridad de desarrollo. Pudiendo ser crítico, importante y útil.
- **Esfuerzo:** establecido por el Arquitecto para medir la complejidad y establecer expectativas de lo que se puede y no se puede lograr en un marco de tiempo determinado Niveles: Bajo = < 1 día, Medio = 1-20 días, Alto = >20 días.
- **Riesgo:** Establecido por el Arquitecto en función de la probabilidad de que la implementación del requisito experimente eventos no deseados Niveles: Bajo = <10%, Medio = 10-50%, Alto = >50%.
- **Estabilidad:** Establecido por el responsable del servicio y el Arquitecto en función de la probabilidad de que el requerimiento cambie o cambie la comprensión del requisito. Niveles: Baja: <10%, Media: 10-50%; Alta: >50%.
- **Impacto Arquitectónico:** Establecido por el Arquitecto de software. Niveles: Bajo: No afecta a la arquitectura existente; Medio: Requiere la ampliación de la arquitectura existente; Alto: La arquitectura existente se debe cambiar para acomodar el requisito.

Fuente: Elaboración propia (2022)

- En la *Ilustración 3* se estableció el mecanismo de seguimiento del avance en el cumplimiento de los requisitos del sistema de precalificación. El objetivo es hacer el seguimiento a las relaciones establecidas entre los artefactos, tal como se muestra en la figura.

Ilustración 3. Trazabilidad del sistema de precalificación



Fuente: Elaboración propia (2022)

Sin embargo, por la complejidad que representa realizar el seguimiento de manera manual a cada uno de los elementos identificados en el árbol de trazabilidad, esta sección de la tarea no se realizará

- Por último, la *Tabla 8* muestra los pasos a seguir para gestionar los cambios a los requisitos.

Tabla 8. Extracto del artefacto Plan de gestión de requisitos: Gestión de cambio de requisitos

Se podrá solicitar cambios siguiendo los siguiente pasos y condiciones:

- Solicitud de cambio: los miembros del equipo podrán solicitar cambios ya sea de manera directa o de ser el caso a través del formulario "Solicitudes de cambio".
- Clasificar la solicitud de cambio en una de las 3 categorías:
  - Defectos de implementación que no afectan a los requisitos
  - Modificaciones de los requisitos existentes de algún tipo
  - Solicitudes de mejora
- Asigne atributos al requisito cambiante

Elabore una matriz y liste los requisitos cambiantes y asigne atributos a cada requisito. Los atributos más importantes son: beneficio, esfuerzo, riesgo, estabilidad, impacto arquitectónico.

- Asigne valores a los atributos del requisito cambiante,

Clasifique cada atributo de cada requisito cada atributo en una escala de alto, medio o bajo. Recuerde que los valores asignados a los atributos de los requisitos permiten tomar decisiones,

- Analizar el impacto y valor del cambio: todo requerimiento pasará en primera instancia por el director de proyecto, quien en conjunto con el analista y Arquitecto de software analizarán el coste del cambio en función del impacto que tiene sobre el sistema global y si se puede implementar fácilmente o no. El director de proyecto es persona que puede aprobar o rechazar una solicitud de Cambio que le compete, sin embargo, si no le compete

Pol. Con. (Edición núm. 69) Vol. 7, No 4, Abril 2022, pp. 1470-1501, ISSN: 2550 - 682X

elevara al patrocinador para tomar una decisión.

El analista de sistemas presenta los cambios de requisitos priorizados a representantes de las partes interesadas y de los miembros del equipo del proyecto, y en función del impacto en los atributos de cada requisito pueden ser aprobados o rechazados.

- Aplicar la solicitud de cambio y verificar la trazabilidad: los cambios a los requisitos, debe realizar de arriba abajo, considerando la trazabilidad. Y verificar a que elementos de la trazabilidad afectó el cambio a los requisitos.
- Mantener el historial de cambios: para realizar un cambio es necesario que el analista de sistemas cree una línea base de los requisitos conocidos, y a partir de este se cree versiones.

Nota: Los cambios a los requisitos que se realicen después de la fase de levantamiento de requisitos, deberán seguir el procedimiento "Realizar control de cambios a proyectos" del proceso gestionar programas y proyectos de la Cooperativa"

Fuente: Elaboración propia (2022)

2. Entender las necesidades de los interesados

Para llevar a cabo esta actividad se procedió a realizar una sesión virtual con las partes interesadas compuesta por: 1 usuario administrador (Analista de Gestión de Requerimientos y Reclamos), 1 usuario experto (Analista de Gestión de Servicios Físicos y Virtuales), 1 usuario final (Asesor Cooperativo), 2 Analistas de Gestión de la Calidad, 1 Especialista en producción, y el director de proyecto y Arquitecto de software. Se procedió a realizar las siguientes tareas:

Se utilizó el artefacto de entrada *Proceso de la precalificación* con su respectiva caracterización del sistema y el artefacto de *especificación de requisitos de software*, donde se describió el modelo de caso de uso a un alto nivel; asimismo, se presentó la problemática y las características principales del sistema, para dar a conocer el contexto y alinear en el mismo enfoque a las partes interesadas, utilizando el artefacto *Visión.*

En cuanto a Buscar actores y casos de uso, se recogieron las nuevas solicitudes de mejora y se verificó si estas afectaban al *modelado de caso de uso*, lo cual no sucedió, por lo tanto, no se actualizó el artefacto de *especificación de requisitos de software*. Sin embargo, sí se validó y actualizaron las especificaciones complementarias del sistema con las partes interesadas, siendo necesario la actualización del artefacto *de* especificaciones complementarias.

Por último, se registraron las solicitudes con su respectivo origen y se priorizaron en el artefacto *Solicitud de las partes interesadas.* A continuación, en la *Tabla 9* se presenta un extracto de la lista de requisitos contenida en este artefacto.

Tabla 9. Extracto de los requisitos priorizados

Código	*Lista de requisitos*	*Origen*	*Priorización*
REQ1	El sistema de precalificación debe permitir generar un nuevo formulario de precalificación	Archivos del sistema actual	Alta
REQ2	El sistema de precalificación debe permitir obtener una tabla de amortización y una lista de requisitos.	Usuario experto	Alta
REQ3	El sistema de precalificación debe permitir consultar, eliminar y editar la precalificación	Archivos del sistema actual	Alta
REQ4	El sistema de precalificación debe permitir calcular un nivel de riesgo y calcular los modelos estadísticos	Usuario experto	Alta
REQ5	El sistema de precalificación debe ser escalable, debe ser migrado a aplicación web, debe ser estable y disminuir las caídas del sistema y deber ser rápido.	Departamento de producción – Tecnología y Usuario final	Alta
REQ6	El sistema de precalificación debe funcionar en las 68 oficinas con 3 asesores de crédito en promedio, con un crecimiento del 10% anual en usuarios finales	Director de proyecto	Alta
REQ7	El sistema de precalificación debe guardar todo cambio en la configuración para auditorias	Usuario administrador	Alta
REQ8	El sistema interactuara con los sistemas: buró de crédito, Jasit y Comunicaliza.	usuario administrador	Alta

Fuente: Elaboración propia (2022)

Luego de obtener las solicitudes de mejora, se actualizó el artefacto *Glosario,* agregando nuevas palabras y el artefacto *Visión* actualizando las principales características del sistema. Finalmente se ejecutó una tarea exclusiva de esta actividad que es *Administrar dependencias,* donde se realizó lo siguiente:

- *Asignar atributos:* En la *Tabla 10* se puede observar un extracto de los requisitos con sus respectivos valores de los atributos, en el que se establecieron los más importantes para cada requisito del sistema de la precalificación, tales como: el beneficio, el esfuerzo a implementar, el riesgo para el esfuerzo de desarrollo, la estabilidad y el impacto arquitectónico de cada requisito. Una vez determinado los atributos, se asignaron valores (alto, medio y bajo) para decidir qué requisito debe ser aprobado o eliminado. Al respecto todos los requisitos fueron aprobados excepto el requisito "Las condiciones establecidas en la precalificación deben cumplirse estrictamente en la solicitud de crédito". La gestión de este cambio se verá en la actividad *Gestionar los requisitos cambiantes.*

Pol. Con. (Edición núm. 69) Vol. 7, No 4, Abril 2022, pp. 1470-1501, ISSN: 2550 - 682X

Tabla 10. Extracto de los requisitos, respectivamente con sus atributos y valores asignados

		ATRIBUTOS						
Tipo de requisito	*Requisitos*	*Estado del requisito*	*Beneficio*	*Esfuerzo*	*Riesgo*	*Estabilidad*	*Impacto Arquitectónico*	*¿Cambio de requisitos?*
CARAC	El sistema de precalificación debe permitir obtener una tabla de amortización	Aprobado	Alto	Bajo	Bajo	Alto	Medio	No
CU	El sistema de precalificación debe permitir crear un nuevo formulario de precalificación	Aprobado	Alto	Bajo	Bajo	Alto	Medio	No

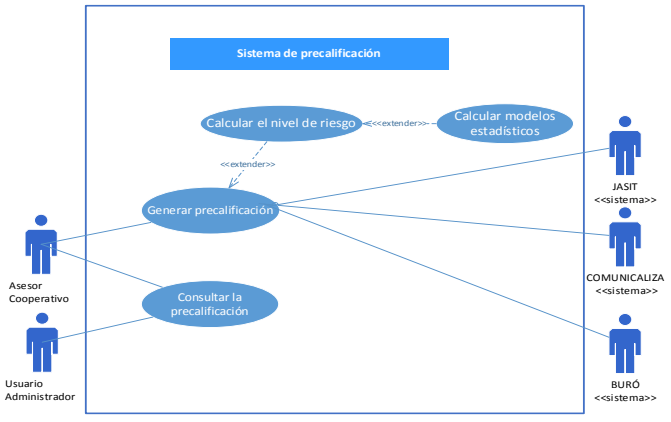
Fuente: Elaboración propia (2022)

- Cabe mencionar que RUP recomienda utilizar el artefacto modelo de diseño, sin embargo, para el proceso definido en esta tesis no se aplica, porque el diseño del sistema no se realiza con el modelo de clases.

3. Definir el sistema

Para realizar esta actividad el analista de sistemas en conjunto con el equipo de proyecto concentró los esfuerzos principalmente en la tarea *Buscar actores y casos de uso* en la que se actualizó el modelo caso de uso, presentándose el resultado en la *Ilustración 4*; se discutió con el equipo el flujo de los casos de uso sin tener todavía un resultado plasmado y se actualizaron las especificaciones complementarias, el resultado se verá en la *Tabla 15*. Además de lo anterior, Además, se actualizó el artefacto *Glosario* con nuevas palabras.

Ilustración 4. Modelado de casos de uso



Fuente: Elaboración propia (2022)

Por otra parte, en la tarea *Obtener los requerimientos de los interesados* se analizó el resultado de la solicitud de las partes interesadas lo cual permitió actualizar las especificaciones complementarias; y en la tarea A*dministrar dependencias* no se realizó ninguna acción ya que no fue necesario refinar los atributos de los requisitos.

- 4. Administrar el alcance del sistema

El alcance del proyecto está definido por el conjunto de requisitos que se le asignen, por lo tanto, los requisitos aprobados con sus respectivos atributos fueron verificados y priorizados en la actividad *Entender las necesidades de los interesados* en la *Tabla 9.* Se realizó lo siguiente:

En la tarea *Elaborar la visión* se actualizaron las características principales; y en *Administrar dependencias* se asignaron atributos y valores a los nuevos requisitos, ver *Tabla 12*.

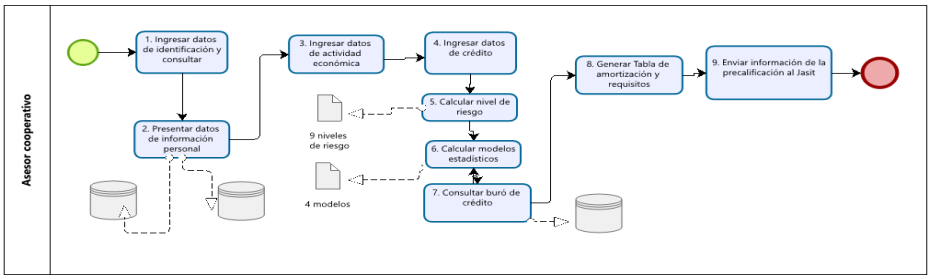
Tabla 12. Extracto de los nuevos requisitos con sus atributos y valores asignados

	Atributos					
Requisitos	*Estado del requisito*	*Beneficio*	*Esfuerzo*	*Riesgo*	*Estabilidad*	*Impacto Arquitectónico*
Cuando se ingrese otro documento de sustento de ingresos este deberá ser justificado	Aprobado	Alto	Bajo	Bajo	Alto	Bajo
Las condiciones establecidas en la precalificación deben cumplirse estrictamente en la solicitud de crédito	Rechazado	Medio	Alto	Bajo	Medio	Alto

Fuente: Elaboración propia (2022)

En esta actividad también se ejecutó la tarea *Priorizar los casos de uso*, donde el arquitecto de Software utilizó el artefacto *Visión y Especificación de requisitos de software;* así como el *Plan de gestión de requisitos,* considerando el impacto de los atributos de cada requisito para priorizar los casos de uso. Como resultado, se obtiene que los cuatro requisitos son priorizados para desarrollarse en una sola iteración. Una vez priorizados los casos de uso, el arquitecto junto con el analista, elaboraron diagramas de actividad para describir el flujo de los casos de uso (ver *Ilustración 5)*.

Ilustración 5. Flujo del sistema de precalificación



Fuente: Elaboración propia (2022)

5. Refinar la definición del sistema

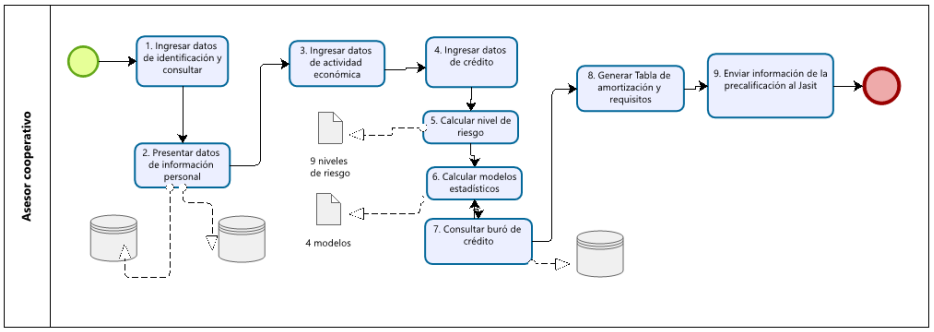
Luego de definir el alcance del proyecto con los requisitos aprobados, el equipo procedió con las siguientes tareas:

- *Detallar los casos de uso:* Con el apoyo del arquitecto de software se completó la definición de los casos de uso identificados y priorizados. Para lograrlo, se utilizaron los artefactos de entrada: *Visión, Solicitud de las partes interesadas, plan de gestión de requisitos y atributos, Especificación de requisitos de software y la Guía de soporte para el modelado de caso de uso e interfaz. Como resultado de esta actividad se obtuvo la salida: Especificación de requisitos de software actualizada. A continuación, en la Tabla 14 se presenta un extracto del detalle del* caso de uso "Calcular el nivel de riesgo". Y en la *Tabla 15* se presentan las *especificaciones complementarias* escritas en términos verificables.

Tabla 14. Descripción del Caso de uso Calcular el nivel de riesgo

Caso de uso Calcular el nivel de riesgo
Descripción: Este caso de uso permite calcular 9 tipologías de socios en función de este obtener una lista de requisitos y unta tabla de amortización. El actor principal de este caso de uso es el usuario (Asistente de Servicios/Asesor Cooperativo).

Diagrama de actividad



Flujo de eventos
El caso de uso inicia cuando el usuario ingresa los datos de identificación del solicitante
Flujo básico
1. El usuario presiona calcular riesgo
2. El sistema calcula el nivel del riesgo realizando lo siguiente:
   - Identifica las variables: 5 variables y coloca puntajes según la siguiente tabla.

Variables			
Variable 1	Resultado 1	Resultado 2	Resultado 3
Puntos	X	x	X

   - El sistema suma los puntajes obtenidos en cada variable y categoriza en riesgo alto, medio y bajo
   - El sistema almacena el **NIVEL DE RIESGO 1** con el símbolo correspondiente.
   - El sistema, luego toma el monto ingresado y categoriza al socio en 3 niveles, la cual es parametrizable en el módulo de Configuración.
   - El sistema agrega el símbolo del nivel de riesgo al campo NIVEL_RIESGO 1, quedando (A1, A2, A3, M1, M2, M3 B1, B2 Y B3) y almacena el puntaje.

Flujo alternativo
Editar la precalificación
1. El usuario selecciona editar la precalificación
2. El sistema habilita los campos que se pueden editar siempre y cuando no haya pasado a la secuencia de verificación.
3. El usuario verifica la edición.
4. El sistema guarda los campos editados

Eliminar la precalificación
1. El usuario puede seleccionar eliminar la precalificación, en cualquier momento del llenado de la precalificación.
2. El sistema solicita al usuario que verifique la eliminación.
3. El usuario verifica la eliminación
4. El sistema elimina la precalificación
5. El sistema cambia el estado de **Activo a Eliminado**, como resultado no se podrá recuperar la información generada.

Requerimientos especiales
La precalificación no necesita de requerimientos especiales
Precondiciones y postcondiciones
Precondiciones
1. El usuario debe ingresar con usuario, contraseña y oficina.
   Los usuarios administradores tienen acceso a la pantalla de Precalificación solo para consultar y los usuarios finales tienen acceso para modificar

Postcondiciones
1. Solo con el cálculo del nivel de riesgo se puede continuar con el caso de uso cálculo de modelos estadísticos.

Fuente: Elaboración propia (2022)

Tabla 15. Extracto de las especificaciones complementarias del sistema

- El sistema de precalificación debe ser migrado a una aplicación web.
- El sistema de precalificación debe estar disponible en un 99%
- El sistema de precalificación debe ser igual o inferior a 10 minutos
- El sistema debe estar disponible en horario de oficina, de lunes a viernes de 7h00 am a 19h00 y sábado y domingo de 7h00 a 14h00.
- Información almacenada por 15 años.

Fuente: Elaboración propia (2022)

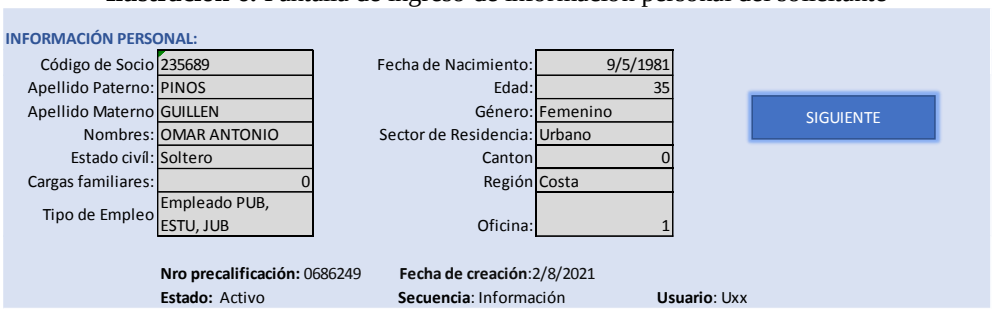
6. Prototipar la interfaz de usuario

Se procedió a crear el prototipo de interfaz de usuario considerando únicamente las ventanas principales de los casos de uso. Se utilizaron las directrices de interfaz de usuario definidas en la

Pol. Con. (Edición núm. 69) Vol. 7, No 4, Abril 2022, pp. 1470-1501, ISSN: 2550 - 682X

Guía de modelado de caso de uso e interfaz. En la *Ilustración 6* se puede observar un extracto del prototipo de la pantalla de información personal.

Ilustración 6. Pantalla de ingreso de información personal del solicitante



Fuente: Elaboración propia (2022)

7. Detallar los requisitos de software

En esta tarea se procedió a unir los artefactos: *Visión, Especificación de requisitos de software y el prototipo de interfaz de usuario* en un solo artefacto llamado *"Perfil de proyecto";* con el objetivo de sintetizar todo lo realizado y disponer de un solo documento, que es requerido por la Cooperativa.

Cabe mencionar que RUP recomienda que se establezcan dos roles: un especificador de requisitos para detallar los casos de uso y un diseñador de requisitos para delinear el interfaz de usuario, sin embargo, al ser este un proyecto pequeño, el Analista de sistemas será el especificador y diseñador de requisitos. Además, antes de prototipar se realiza la tarea Modelar la interfaz de usuario; pero en la metodología de la presente investigación no se realizará debido a que, luego de detallar los casos de uso, el Analista de sistemas junto con el equipo desarrolla la interfaz de las ventanas principales permitiendo al programador guiarse y desarrollar lo solicitado.

8. Gestionar los requisitos cambiantes

En este proyecto se gestionaron dos cambios a los requisitos y se realizó siguiendo los pasos establecidos en el *Plan de gestión de requisitos y atributos*, que contiene los lineamientos para gestionar los requisitos cambiantes. Los pasos se describen a continuación:

- *Solicitud de cambio:* En una reunión de seguimiento las partes interesadas solicitaron incorporar tres requisitos adicionales al sistema.
- *Clasificar la solicitud de cambio:* Los cambios se clasificaron como solicitudes de mejora.

- *Asignar atributos al requisito cambiante y asignar valores a los atributos del requisito cambiante:* Los atributos asignados y sus valores se pudieron observar en la *Tabla 12* de la actividad *Administrar el alcance del Sistema*, *Analizar el impacto y valor del cambio.* El analista y arquitecto de software asignaron los atributos y valores a los nuevos requisitos con el objetivo de hacer el respectivo seguimiento al cumplimiento de la trazabilidad. Una vez clasificadas, se presentó a un representante de las partes interesadas y a los miembros del equipo del proyecto; el director de proyecto decidió incorporar dos de los tres requisitos, por su alto beneficio, bajo esfuerzo y bajo impacto arquitectónico.
- *Aplicar la solicitud de cambio:* El requisito que fue aceptado e incorporado en los artefactos *Visión* y *Especificación de requisitos de software, en las secciones de casos de uso y especificaciones complementarias* correspondientes. El equipo evaluó el impacto de los requisitos, mismo que no alteró el alcance del sistema.
- *Mantener el historial de cambios:* se actualizó la versión de los artefactos que fueron afectados con la incorporación de los requisitos.

Una vez incorporado el requisito se realizó la tarea de *Revisión de requisitos*, rol que fue cumplido por el analista de gestión de la calidad, a través de una reunión virtual, quien comprobó que los requisitos identificados y priorizados cumplían con las necesidades de las partes interesadas. La reunión estuvo compuesta por el director y el patrocinador del proyecto, un usuario administrador, un usuario final y el arquitecto de software, obteniendo como resultado el artefacto *Registro de revisión;* el cual captura los resultados de la revisión de un artefacto del proyecto RUP recomendando utilizar como entrada el artefacto *Plan de iteración* que detalla las actividades y tareas secuenciadas, con los recursos asignados; sin embargo, para esta metodología no se considera, porque dicha tarea corresponde al Especialista de gestión de proyectos.

Validación empírica de la metodología de gestión de requerimientos para proyectos de desarrollo de software

Esta sección presenta un cuasi-experimento que tiene el objetivo de estudiar la metodología RUP como soporte para la gestión de requerimientos de proyectos de desarrollo de software para la Cooperativa Jardín Azuayo. El cuasi-experimento fue diseñado siguiendo las guías propuestas por Wohlin, et al (2012); las cuales establecen cuatro pasos: i) Planificación del experimento, ii) Preparación y ejecución del experimento, iii) Análisis de datos, y iv) Amenazas a la validez.

i) Planificación del experimento

El objetivo del cuasi-experimento según el paradigma Métrica de Pregunta – Objetivo (GQM por sus siglas en inglés) mencionado por Basili, et al (1994), es analizar la metodología de gestión de requerimientos, con el propósito de evaluar respecto a su facilidad de uso percibida, utilidad percibida e intención de uso, desde el punto de vista de un conjunto de profesionales con el cargo de Analistas de gestión de servicios físicos y virtuales.

En cuanto a la aplicación de la Metodología de gestión de requerimiento, se propone un escenario de complejidad media en un contexto industrial limitado, en este caso es el siguiente: La Cooperativa de ahorro y crédito Jardín Azuayo tiene como objetivo proveer servicios financieros para mejorar la calidad de vida de los socios, siendo el principal servicio financiero la colocación de créditos. Para completar el servicio de crédito es necesario crear un sistema de calificación que permita al Asesor Cooperativo: calificar el crédito con las 5 C´s: Carácter, Capacidad, Capital, Condiciones y Colateral. Los resultados obtenidos de la calificación son un insumo para tomar una decisión con respecto al crédito.

Este sistema tiene dos usuarios: uno final que es el Asesor Cooperativo quien accede al sistema para calificar, editar y consultar la calificación, y uno administrador, quien accede al sistema para consultar la información de la calificación. Para calificar el sistema, interactuará con otro actor que es el sistema JASIT con el objetivo de obtener información del socio.

La metodología esta soportada por seis actividades para la gestión de requerimientos, sin embargo, el enfoque se hará en dos de las seis actividades: Analizar el problema y Definir el sistema. Se consideran únicamente estas dos actividades porque son las más importantes de la metodología propuesta para una primera aplicación en la gestión de los requerimientos de proyectos de desarrollo de software. Estas dos actividades permiten identificar el problema, e identificar a las partes interesadas y sus necesidades y a partir de estas definir las funcionalidades principales del sistema, a través de un modelado de casos de uso, descripción del flujo de un caso de uso y definición de las especificaciones complementarias.

En lo que respecta a la selección de participantes, el experimento se realizará en un contexto profesional, seleccionando a cuatro personas, que representan el 80% de profesionales del departamento de Gestión de servicios de la Cooperativa Jardín Azuayo, con el cargo de “Analistas de gestión de servicios físicos y virtuales”, donde, para cumplir con su perfil deben realizar actividades dirigidas a las mejoras y/o mantenimiento de los sistemas de software.

Luego se seleccionaron las variables tanto independientes como dependientes. En este estudio, la variable independiente es una metodología que soporta las actividades para la gestión de requerimientos de proyectos de desarrollo de software; y las variables dependientes definidas como variables subjetivas de interés, se definen basados en los constructores de un modelo teórico para el análisis de la aceptación y adopción de tecnologías de información emergentes: el Modelo de Aceptación de la Tecnología (TAM, por sus siglas en inglés) propuesto por Davis (1989).

Sus principales constructores son: Facilidad de uso percibida (FUP), que es el grado en que los participantes perciben que requieren de un esfuerzo mínimo para aprender y usar un método en particular; Utilidad Percibida (UP), la cual define el grado en el que los participantes creen que el uso de un método en particular mejoraría su rendimiento dentro de un contexto organizacional y representa un juicio perceptivo de la efectividad del método; e Intención de uso (IU), referida a la medida en la que una persona tiene la intención de usar un método particular, representa un juicio perceptivo de la eficacia del método.

Estas tres variables subjetivas se midieron mediante una encuesta tipo Likert, que contiene 14 preguntas cerradas; de las cuales 5 preguntas corresponden a la variable FUP, 6 a la variable UP y 3 a la variable IU. Cada pregunta tiene dos estados opuestos que representan los puntajes máximos y mínimos (5 y 1), donde 3 es un valor neutro. El valor agregado de cada variable se calcula como la media aritmética de las respuestas a las preguntas asociadas a cada una de las variables subjetivas. Además, el orden de las preguntas se alteró aleatoriamente para evitar el sesgo de respuestas sistemáticas y algunas preguntas se reformularon convirtiéndolas en preguntas negativas para evitar respuestas monótonas (Hu, Chau, Sheng, & Tam, 1999).

La encuesta de evaluación de las variables subjetivas también incluyó 2 preguntas abiertas a fin de obtener retroalimentación de los participantes, estas son: ¿Tiene alguna sugerencia de cómo hacer que esta metodología sea más fácil de usar? y ¿Cuáles son las razones por las que tiene o no la intención de usar esta metodología en un futuro?

Adicional a lo anterior, se evaluó cuán entendible es la metodología, para lo cual, estudios empíricos recomiendan que para medir el grado de entendimiento se pueden utilizar las tareas de solución de problemas, como lo exponen Gemino (2005) y Bodart, Patel, Sim, y Weber (2001), por lo tanto, para responder a preguntas de solución de problemas el participante debe razonar acerca del dominio.

Se utilizaron dos variables basadas en el rendimiento: *Efectividad*: definición del problema, identificación de las partes interesadas, definición de características principales del sistema; descripción del flujo de eventos de un caso de uso y definición de especificaciones complementarias; y Eficiencia: calculada como el ratio entre el número de artefactos dividido por el tiempo total empleado en su especificación. Por último, se definieron tres hipótesis: 1) la metodología se percibe como difícil de usar; 2) la metodología no es percibida como útil; y 3) no existe la intención de utilizar la metodología en el futuro.

ii) Preparación y ejecución del experimento

El experimento fue planificado para ser realizado en dos sesiones. En la primera, se realizó una sesión de entrenamiento de 90 minutos, cuyo objetivo era presentar los temas sobre conceptos de Ingeniería de software, proceso de software, metodologías de desarrollo de software, ingeniería de requisitos; visión general de la metodología de gestión de requerimientos; y entrenamiento en las actividades de la metodología RUP: analizar el problema, entender las necesidades de los interesados, definir el sistema, administrar el alcance del sistema y gestionar los requisitos cambiantes. La segunda sesión de ejecución del experimento tuvo lugar unos días después con un tiempo esperado de duración de 120 minutos y consistió en dos tareas: analizar el problema y definir el sistema.

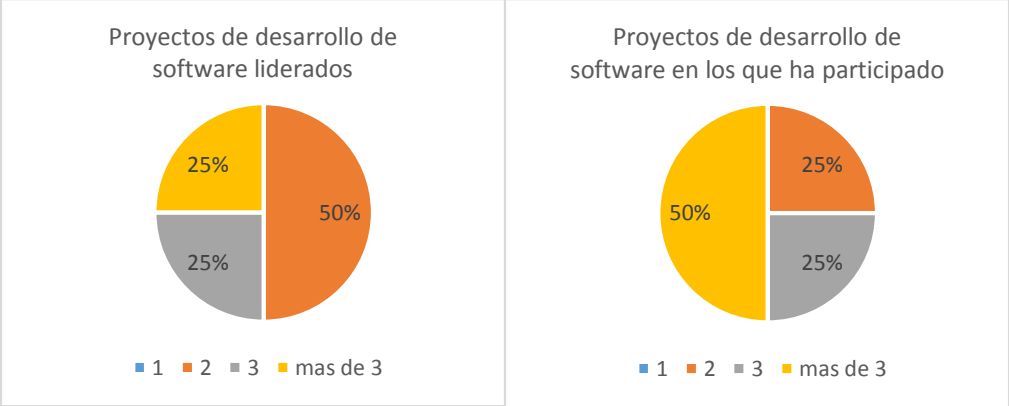
Múltiples documentos fueron diseñados como instrumentación para el experimento, a saber, boletines, anexos y guías. Asimismo, el material de entrenamiento consistió en un conjunto de diapositivas conteniendo la descripción de la metodología de gestión de requerimientos con un ejemplo de aplicación. El experimento se llevó a cabo con un grupo de 4 profesionales Analistas de gestión de servicios físicos y virtuales a través de una reunión virtual. Durante la sesión experimental, la capacitadora respondió a preguntas y clarificó dudas de los participantes.

iii) Análisis de datos

- *Análisis encuesta pre-ejercicio:*

Con respecto al conocimiento, ninguno de los participantes encuestados conocía ni tenía experiencia en metodologías de gestión de requerimientos. Sin embargo, todos los sujetos han participado en la gestión de requerimientos de proyectos de desarrollo de software de la organización, ya sea como miembros del equipo de proyectos o como líderes del proyecto; en al menos 2 proyectos de desarrollo de software. La *Ilustración 7* muestra la participación en los proyectos.

Ilustración 7. Participación de los participantes en proyectos de desarrollo de software



Fuente: Elaboración propia (2022)

- *Análisis Cualitativo*

Para realizar la segunda parte del análisis se solicitó a los participantes que llenen la encuesta "Percepción acerca de la Metodología de gestión de requerimientos de proyectos de desarrollo de software". Una vez obtenidas las respuestas de la encuesta, se agruparon según las variables de: factibilidad de uso percibida y utilidad percibida e intención de uso; posterior a ello, se calcularon los estadísticos descriptivos: media y desviación estándar; cuyos resultados fueron:

1. La variable FUP obtuvo una media de 3.85, por lo que se determina que la metodología es percibida como fácil de usar.
2. La métrica UP obtuvo una media de 4.75, por lo que se determina que la metodología es percibida como útil de usar.
3. La métrica IU obtuvo una media de 4.58, por lo que se determina que existe la intención de usar la metodología en el futuro.

Esto significa que, bajo las condiciones del experimento, la metodología de gestión de requerimientos para proyectos de desarrollo de software, es percibido como fácil de usar, útil y los participantes muestran una clara intención de emplearlo en el futuro. Por otra parte, el análisis de las respuestas a las preguntas abiertas reveló que la mayoría de los participantes aplicarían la metodología propuesta en un futuro por las siguientes razones: se considera a todas las partes interesadas y los requerimientos especificados son documentados.

Pol. Con. (Edición núm. 69) Vol. 7, No 4, Abril 2022, pp. 1470-1501, ISSN: 2550 - 682X

- *Análisis cuantitativo*

Para analizar las variables cuantitativas de rendimiento se calcula la media y desviación estándar, cuyos resultados muestran que la efectividad de los participantes fue de 93% y la eficiencia de 79%, deduciéndose que la aplicación de la metodología permitió gestionar los requerimientos correctamente.

iv) Amenazas a la validez

En esta sección, se especifican las principales amenazas que pueden poner en riesgo la validez del cuasi-experimento. Se consideran cuatro tipos de amenazas propuestas por Cooy & Campbell (1979):

- *Validez interna*

Los principales problemas a la validez interna fueron: experiencia de los participantes y cuán entendibles son los documentos. En la primera, se mitigó la amenaza mediante un cuestionario pre-ejercicio para analizar las posibles diferencias entre participantes con experiencia y participantes sin experiencia, lo que permitió rechazar el posible impacto que el factor experiencia puede tener en las variables bajo estudio; pues los 4 participantes han gestionado los requerimientos de al menos un proyecto de desarrollo software, y han formado parte del equipo de proyectos en al menos un proyecto. En la segunda, se mitigó la amenaza pues se aclararon las dudas y malentendidos que se presentaron en la sesión experimental.

- *Validez externa*

Entre las principales amenazas a la validez externa del experimento están el diseño de la evaluación, la selección de participantes y el tamaño y complejidad de las tareas experimentales, pudiendo afectar a la representatividad de los resultados. En la primera, se incluyeron las actividades más representativas que soportan a la gestión de requerimientos, es decir, las más relevantes y suficientes para lograr un entendimiento de la metodología. Además, el ejercicio se ajustó para que los participantes apliquen todas las reglas de la guía al menos una vez. En cuanto a la segunda, la selección de los participantes, el cuasi-experimento fue realizado con profesionales con un nivel medio en conocimiento para gestionar requerimientos. Por último, en lo que respecta al tamaño y complejidad de las tareas experimentales, se propusieron varias tareas con suficiente nivel de complejidad, debido a las restricciones de tiempo de las sesiones. También se proveyeron plantillas y guías para facilitar la ejecución de las tareas.

- *Validez de constructo*

El propósito de la validez de constructor es demostrar cómo las métricas investigadas cumplen el objetivo de los investigadores. Dado que la fiabilidad de la encuesta de evaluación es la principal amenaza, esta fue analizada mediante la aplicación del test Alfa de Cronbach (Maxwell, 2002). Los resultados indican que los valores de cada variable FUP (0,84), UP (0,78) e IU (0,74), fueron mayores al valor mínimo de aceptación $\alpha > 0.70$ según Maxwell (2002). Por lo tanto, se puede deducir que la encuesta de evaluación es confiable y válida de los constructos fundamentados en la percepción/intención dentro del marco de validación propuesto.

- *Validez de las conclusiones*

La principal amenaza es la validez de las pruebas estadísticas aplicadas, por lo tanto, para mitigar esta amenaza se aplicaron pruebas estadísticas descriptivas que permiten medir objetivamente los resultados.

Discusión de los resultados

La adaptación de la metodología RUP para la gestión de requerimientos en el desarrollo de proyectos de desarrollo de software para la cooperativa de ahorro y crédito se escogió debido a sus antecedentes de aplicabilidad en entidades financieras y en el sector académico, documentadas en los estudios realizados por Sopingi, et al (2021); Arias, (2012) y Jaramillo (2016), en los que se aplica el framework RUP con excelentes resultados. Al revisar los mencionados estudios, y, al mismo tiempo, conocer las diferencias entre metodologías ágiles y tradicionales como la RUP, planteadas por autores como Molpeceres (2002), se evidencian grandes beneficios en el desarrollo de software.

Entre dichos beneficios, detectados con la presente investigación, se encuentran que RUP es una metodología muy robusta y utilizada en pequeños, medianos y grandes proyectos de software; maneja una serie de entregas ejecutables, integra continuamente la arquitectura para producir nuevas versiones mejoradas; se adapta a las necesidades de la organización y del proyecto; usa UML como notación principal; aplica los casos de uso para definir las funcionalidades del sistema; usa documentación sólida para la descripción de requisitos que son definidos por el cliente considerando a todas las partes interesadas; establece un cronograma inicial sobre el cual se realiza el seguimiento, evaluación y elaboración de informes; y maneja una eficiente gestión de cambios

Pol. Con. (Edición núm. 69) Vol. 7, No 4, Abril 2022, pp. 1470-1501, ISSN: 2550 - 682X

de requerimientos pues antes de aceptar un cambio se analiza el impacto y también se deja constancia del mismo.

En otro orden de ideas, la adaptación de la metodología RUP por parte de los desarrolladores de software, se basa en dos razones fundamentales: es iterativo y proporciona una buena documentación de sus actividades, así lo expresa Ahmad et al (2011); lo cual concuerda con la realidad presente durante la aplicabilidad y validación de la metodología a la cooperativa objeto de estudio; a lo cual es preciso agregar otras razones: que es incremental, está centrado en la arquitectura y dirigido a casos de uso; tal y como lo manifiesta Pereira (2011).

Casi todos los procesos ejecutados con metodologías tradicionales, se centran en el cumplimiento de las especificaciones técnicas y de los procesos, dejando de lado la usabilidad del producto y excluyendo los beneficios significativos, tales como, mejora de la productividad, aumento de la eficiencia, minimización de errores, reducción de entrenamiento, mejoras en la aceptación, entre otros. Según Cadavid, et al (2013), las metodologías tradicionales, como RUP, se caracterizan por su rigidez metodológica y la exhaustiva documentación; inician el desarrollo de un proyecto centrándose en la especificación de requisitos tratando de asegurar resultados de alta calidad, y de cumplir el plan de proyecto. Esto se demostró con la aplicación de la metodología en el desarrollo del software de la Cooperativa objeto de estudio.

Por lo tanto, el éxito del desarrollo del software está directamente relacionado con la calidad de los requisitos, bien sea que se modifiquen, se supriman, se adicionen, e impacten directamente en el tiempo, costo, y la calidad del producto final; al respecto, Hanssen (2005), explica que la metodología RUP en la gestión de requerimientos, debe ser adaptada, reducida y especializada al contexto de uso, ya que ningún proyecto de desarrollo de software es igual al otro; esto fue lo que se hizo con la gestión de requerimientos en el desarrollo de software de la Cooperativa de Ahorro y Crédito Jardín Azuayo.

Conclusiones

La metodología de gestión de requerimiento para proyectos de desarrollo de software en la Cooperativa Jardín Azuayo define un proceso que incluye seis actividades. Durante la primera actividad, analizar el problema, el analista de sistemas comprende la definición y el alcance del problema que se está tratando de resolver con el nuevo sistema; en la segunda actividad, entender las necesidades de los interesados, el analista de sistema recopila la información de las partes

interesadas para comprender cuáles son sus necesidades, y estas a su vez servirán para definir las características de alto nivel del sistema; en la tercera actividad, definir el sistema, el analista junto con el equipo del proyecto se centran en identificar los actores y casos de uso de manera más completa y amplían los requisitos complementarios o no funcionales.

En la cuarta actividad, administrar el alcance del sistema, el analista de sistemas prioriza los requisitos que se incluirán en la iteración actual. En la quinta actividad, refinar la definición del sistema, el analista de sistemas detalla a profundidad lo que debe hacer el sistema en una especificación de requisitos de software; realiza un prototipo inicial de interfaz de usuario para exponer una arquitectura candidata frente a escenarios principales que demuestren la viabilidad del sistema. Finalmente, la actividad gestionar los requisitos cambiantes, se ejecuta de manera transversal mientras se desarrolla las cinco actividades, con el objetivo de gestionar correctamente los cambios a los requisitos.

Para validar la metodología propuesta se ejecutó un cuasi-experimento orientado a comprobar la facilidad de uso percibida, la utilidad de uso percibida y la intención de uso en el futuro de la metodología de gestión de requerimientos para proyectos de desarrollo de software de la Cooperativa Jardín Azuayo. El experimento consistió en un escenario real en el cual se gestionan los requerimientos de un proyecto de desarrollo de software, este consiste en desarrollar un sistema de Calificación, que permita al Asesor Cooperativo calificar el crédito con las 5 C´s: Carácter, Capacidad, Capital, Condiciones y Colateral.

Los resultados obtenidos de la calificación son un insumo para tomar una decisión con respecto al crédito. Cabe mencionar que este sistema ya existe, sin embargo, es necesario migrar a una plataforma web con un desarrollo propio. Los participantes elaboraron la Visión identificando el problema, las partes interesadas y definieron las características principales del sistema; posterior a ello, identificaron a los usuarios y sistemas, representados por actores, que interactuaron con el sistema; buscaron casos de uso; y definieron las especificaciones complementarias; finalmente describieron el flujo de eventos de uno de los casos de uso y especificaron en términos verificables los requisitos complementarios. Además, revisaron y evaluaron el plan de gestión de requisitos del sistema de Precalificación.

Los resultados permitieron concluir que, bajo las condiciones del experimento, la metodología de gestión de requerimientos es percibido como fácil de usar, útil, y los participantes muestran una clara intención de emplearlo en el futuro. Adicionalmente, se realizó un análisis cuantitativo de la

efectividad y eficiencia, que ha permitido constatar que, en general, la aplicación de la metodología permitió gestionar correctamente los requerimientos del proyecto de calificación de crédito.

Desde el punto de vista investigador, el experimento ha resultado ser un medio valioso para obtener retroalimentación de mejora de la metodología de gestión de requerimientos. Desde un punto de vista práctico, los resultados experimentales proporcionan buenos resultados en cuanto al rendimiento de la metodología propuesta, sin embargo, es necesario analizar si los mismos resultados se producen con más participantes que se encargan de liderar y por ende de gestionar los requerimientos de proyectos de desarrollo de software en la Cooperativa, con proyectos más grandes y más complejos.

De acuerdo a la revisión del estado del arte, éste es el primer estudio empírico que proporciona evidencia de la utilidad de una metodología de gestión de requerimientos para proyectos de desarrollo de software de una institución financiera. Por lo tanto, se concluye que, a diferencia de otros estudios en los que se ha aplicado la metodología de Proceso Racional Unificado (RUP), esta metodología para el desarrollo de todas las actividades provee un conjunto de tareas iterativas, guías para la ejecución y formatos que definen y describen la estructura y contenidos de los artefactos a generar, producto de la ejecución de las actividades; permitiendo un desarrollo de software, más eficiente, predecible, y adaptable a las necesidades tanto de los prestadores del servicio como de los usuarios finales.

Referencias

1. Addetla, S., Gorde, M., Ghadge, S., Kusal, S., & Bongale, A. (2014). Design of mobile event management and broadcast system using rational unified process. *International Journal of Computer Science and Mobile Computing*, *3*(4), 412-421.
2. Ahmad, B., Javed, M., Ahmad, S., Khan, I., Ahmad, M., & Saqib, S. (2011). Exploring documentation: a trivial dimension of RUP. *Computer Engineering and Intelligent Systems, 2*(6), 72-77.
3. Anwar, A. (2014). A review of RUP (Rational Unified Process). *International Journal of Software Engineering, 5*(2), 8-24.
4. Arias, V. (2012). *Sistema de validación y generación de mensajería de prueba para procesos batch en entidades financieras.* Tesis, Universidad Libre, Facultad de Ingeniería, Bogotá.

5. Basili, V., Caldiera, G., & Rombach, H. (1994). The Goal Question Metric Approach. En J. W. Sons, *Encyclopedia of Software Engineering* (págs. 1-10).
6. Bodart, F., Patel, A., Sim, M., & Weber, R. (2001). Should Optional Properties Be Used in Conceptual Modelling? A Theory and Three Empirical Tests. *Inf. Syst. Res.*(12), 384–405.
7. Cadavid, N., Fernandez, J., & Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva, 11*(2), 30-39.
8. Cooperativa de Ahorro y Crédito Jardín Azuayo. (2018). *Plan estrategico 2019- 2023.* Cooperativa de Ahorro y Crédito Jardín Azuayo. Obtenido de https://www.jardinazuayo.fin.ec/resources/files/undefined-Plan%20Estrategico%202019%20-%202023.pdf
9. Davis, F. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 319-340. doi:10.2307/249008
10. Gemino, A. (2005). Complexity and Clarity in Conceptual Modeling: Comparison of Mandatory and Optional Properties. Data & Knowledge Engineering. *55*, 301-326.
11. Gorschek, T. (2006). A Model for Technology Transfer in Practice. *Software IEEE* (23), 88–95.
12. Hanssen, G., Westerheim, H., & Bjørnson, F. (2005). Tailoring RUP to a Defined Project Type: A Case Study. *SINTEF ICT* (7465), 314–327.
13. Hu, P., Chau, P., Sheng, O., & Tam, K. (1999). Examining the Technology Acceptance Model Using Physician Acceptance of Telemedicine Technology. *Journal of Management Information Systems, 16*(2), 91-112.
14. Jaramillo, W. (2016). *Aplicación de la metodología RUP y el patrón de diseño MVC en la construcción de un sistema de gestión académica para la Unidad Educativa Ángel gestión académica para la Unidad Educativa Ángel.* Pontificia Universidad Católica del Ecuador, Quito. Obtenido de http://repositorio.puce.edu.ec/bitstream/handle/22000/11264/Documento%20Disertaci%c3%b3n%20Wendy%20Jaramillo.pdf?sequence=1&isAllowed=y
15. Maida, E., & Pacienzia, J. (2015). *Metodologías de desarrollo de software.* esis de Licenciatura en Sistemas y Computación, Universidad Católica Argentina, Facultad de Química e Ingeniería "Fray Rogelio Bacon", Argentina. Obtenido de https://repositorio.uca.edu.ar/handle/123456789/522

16. Maxwell, K. (2002). *Applied Statistics for Software Managers.* Prentice-Hall.
17. Molpeceres, A. (2002). *Procesos de desarrollo RUP, XP y FDD. AT Javahispano.* Obtenido de http://www.javahispano.org: Recuperado de: http://www.javahispano.org/contenidos
18. Nextel Engineering. (2011). *Control del ciclo de vida del desarrollo software.* Obtenido de http://www.nexteleng.es: http://www.nexteleng.es/ingenieria/ibmrational.asp
19. Parra, E. (septiembre-diciembre de 2011). Propuesta de metodología de desarrollo de software para objetos virtuales de aprendizaje -MESOVA. (u. U. Norte, Ed.) *Revista Virtual Universidad Católica del Norte F*(34), 113-137.
20. Pereira, A. (2011). Un perfil UML 2.0 para el modelado de planes del entrenamiento deportivo. *Revista Avanzada Científica, 14*(1), 1-13.
21. Shadish, W., Cook, T., & Campbell, D. (2002). *Experimental and Quasi-Experimental Designs for Generalized Causal Inference* . Boston, New York: Houghton Mifflin Company.
22. Sopingi, Meikati, E., & Wijiyanto. (2021). Application of the Rational Unified Process Method in Web Service Development Payment System Integration with Multibank Virtual Accounts. *Jurnal E-KOMTEK (Elektro-Komputer-Teknik), 5*(1), 75-88.
23. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., & Wesslén, A. (2012). Experimentation in Software Engineering. Springer Heidelberg. doi:doi:10.1016/S0065-2458(08)60338-1