Recibido** 24/09/2020* ***ReCIBE, Año 9 No. 1, Mayo 2020
***Aceptado** 12/10/2020*

ESENCIALIZACIÓN DE LA PRÁCTICA GESTIÓN DE REQUISITOS DE RUP

ESSENTIALIZATION OF THE RUP REQUIREMENTS MANAGEMENT PRACTICE

Jhonny E. Chapal Vallejo
jhonnychapal@gmail.com

John S. Guerrero Riascos
snydeerjs10@gmail.com

Alexander A. Barón Salazar
abaron_98@udenar.edu.co

Facultad de Ingeniería,Universidad de Nariño,Pasto, Nariño, Colombia.

RESUMEN

La construcción y aplicación de buenas prácticas en la gestión de requisitos de software es importante para obtener productos de alta calidad. En la industria de software se aplican procesos de mejoramiento continuo de prácticas para educción, análisis, especificación y validación de requisitos. Se evidencia que, en el contexto industrial, se hace compleja la aplicación de prácticas que proponen métodos de software por la dificultad para identificar los componentes de la práctica. Es el caso de la práctica Gestión de Requisitos de RUP. La aplicación de esta práctica es compleja porque no se identifican claramente los componentes de la práctica y los mecanismos de aplicación, seguimiento y evaluación. En este artículo se aplica el Modelo para la Definición de Prácticas en Ingeniería de Software a la práctica Gestión de Requisitos de RUP. Este proceso que busca identificar y definir los elementos esenciales para facilitar la aplicación, seguimiento y evaluación de prácticas se denomina esencialización. De esta manera a partir de una práctica existente se obtiene una práctica bien formada y nombrada que facilita su aplicación, seguimiento y evaluación en contextos reales. Una práctica esencializada facilita a los practicantes entender, aplicar y evaluar la práctica ya que define una estructura adecuada con elementos bien definidos. También aporta en el seguimiento a la salud y el progreso del esfuerzo de Ingeniería de Software mediante el uso del núcleo del estándar Essence.

PALABRAS CLAVE

Gestión de Requisitos, Esencialización de prácticas, Núcleo de Essence.

ABSTRACT

In the software requirements management, the construction and application of good practices are important to obtain high quality products. In the software industry, continuous improvement processes of practices are applied for the preparation, analysis, specification and validation of requirements. In the industrial context, it is evident that the application of practices that propose software methods becomes complex due to the difficulty in identifying the components of the practice. This is the case of the RUP Requirements Management practice. The application of this practice is complex because the components of the practice and the mechanisms of application, the monitoring and evaluation are not clearly identified. In this article, the Model for the Definition of Practices in Software Engineering is applied to the RUP Requirements Management practice. This process called essentialization seeks to identify and define the essential elements to facilitate the application, monitoring and evaluation of practices. In this way, from an existing practice, a well-formed and named practice is obtained that facilitates its application, monitoring and evaluation in real contexts. An essentialized practice makes it easier for practitioners to understand, apply and evaluate the practice as it defines an adequate structure with well-defined elements. It also contributes to monitoring the health and progress of the Software Engineering effort by using the kernel of the Essence standard.

KEYWORDS

Essence kernel, Essentialization of practices, Requirements Management.

1. INTRODUCCIÓN

La construcción y aplicación de buenas prácticas en la gestión de requisitos de software es importante para obtener productos de alta calidad (Londoño & Tabares, 2008). Otros autores llegan a una conclusión similar al afirmar que, en la Ingeniería de Requisitos es determinante lograr productos de software correctos, fiables y mantenibles. Por lo tanto, es necesario tener buenas prácticas para separar y especificar correctamente los requisitos, controlar su evolución y soportar los cambios (Tabares, et al., 2007).

En el contexto industrial de la Ingeniería de Software, se hace compleja la aplicación de prácticas debido a que, es difícil identificar y definir los componentes de la práctica en las formas de representación que diferentes autores proponen. En particular, en las propuestas que describen la práctica Gestión de Requisitos de Rational Unified Process (PGR-RUP), no se identifican claramente los componentes y los mecanismos de aplicación, seguimiento y evaluación. Por ejemplo, en trabajos como (Rational Software Company, 1998), (Metzner & Niño, 2016), (Vera, et al., 2019) y (Jacobson, et al., 2000), se describe la práctica de diversas maneras. En estas propuestas es difícil identificar y definir componentes de la práctica como: nombre de la práctica, criterios de entrada, criterios de finalización, actividades, tareas y productos de trabajo.

En (Barón, 2019) se propone un modelo para definir prácticas bien formadas y nombradas. Una práctica es bien nombrada si su nombre incluye: un verbo nominalizado, un adjetivo y un sustantivo. Así mismo, una práctica es bien formada si el conjunto de actividades cumple con reglas de coherencia, consistencia y suficiencia. El proceso para identificar y definir los elementos esenciales de la práctica se denomina esencialización.

En este artículo se presenta el proceso de esencialización de la PGR-RUP aplicando el modelo de Barón propuesto en (Barón, 2019). La PGR-RUP esencializada, facilita a los practicantes entender, aplicar y evaluar la práctica ya que se define una estructura adecuada con los componentes bien definidos. La PGR-RUP esencializada, facilita también, el seguimiento a la salud y el progreso del esfuerzo de Ingeniería de Software mediante el uso del núcleo del estándar Essence.

Este artículo está estructurado de la siguiente manera: en el capítulo II se presenta el contexto de esencialización de PGR-RUP. Más adelante, en el capítulo III, se describe el estudio sobre las formas de representación y descripción de la PGR-RUP. Posteriormente, en el capítulo IV, se presenta la conversión de la PGR-RUP en una práctica bien formada y nombrada. A continuación, en el capítulo V, se presentan las conclusiones y el trabajo futuro. Finalmente, se relacionan las referencias de la investigación.

2. CONTEXTO DE ESENCIALIZACIÓN DE PGR-RUP

A continuación, se presentan los fundamentos teóricos que son necesarios para el desarrollo de esta investigación.

2.1 ESSENCE, EL MARCO DE TRABAJO PARA DEFINIR PRÁCTICAS (OMG, 2018)

La comunidad Software Engineering Method and Theory (SEMAT, por su sigla en inglés), consolida un marco de pensamiento denominado Essence que involucra reestructurar la Ingeniería de Software y mejorar sus métodos.

Essence incluye un núcleo que proporciona un conjunto de elementos que prevalecen en cada esfuerzo de Ingeniería de Software. El núcleo de Essence incluye: (i) las áreas de interés, que permiten organizar los elementos del núcleo en cliente, solución y esfuerzo (ver Figura 1); (ii) los alfas, son las cosas con las que siempre se trabaja, permiten dimensionar el esfuerzo de Ingeniería de Software que el equipo realiza (ver Figura 2); (iii) los espacios de actividad, son las cosas que siempre se hacen, permiten definir el componente dinámico del núcleo (ver Figura 3); (iv) las competencias, son las habilidades necesarias que el equipo debe desarrollar para abordar el esfuerzo de Ingeniería de Software (ver Figura 4).

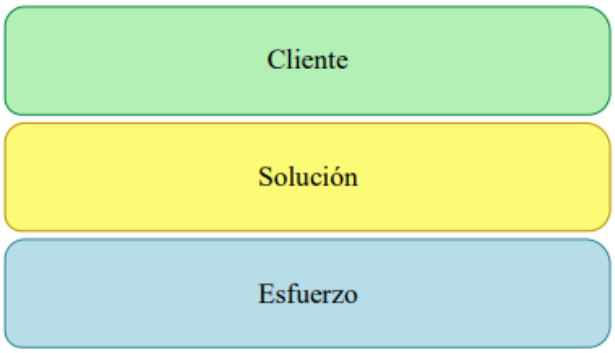

Figura 1. Áreas de interés del núcleo (OMG, 2018)

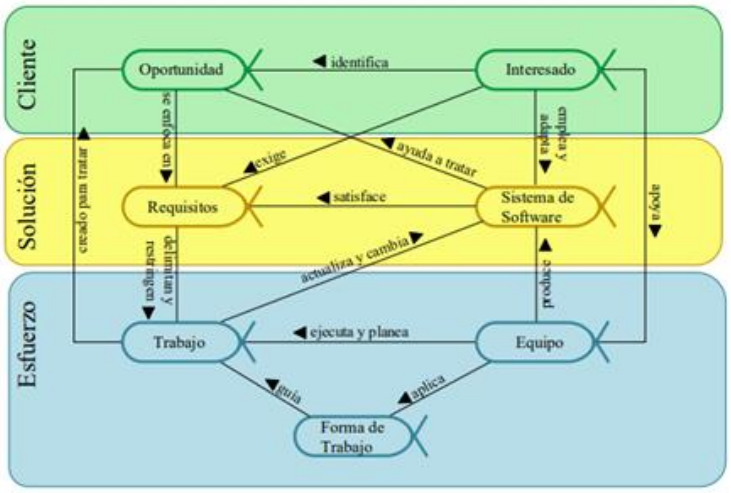

Figura 2. Alfas del núcleo (OMG, 2018)

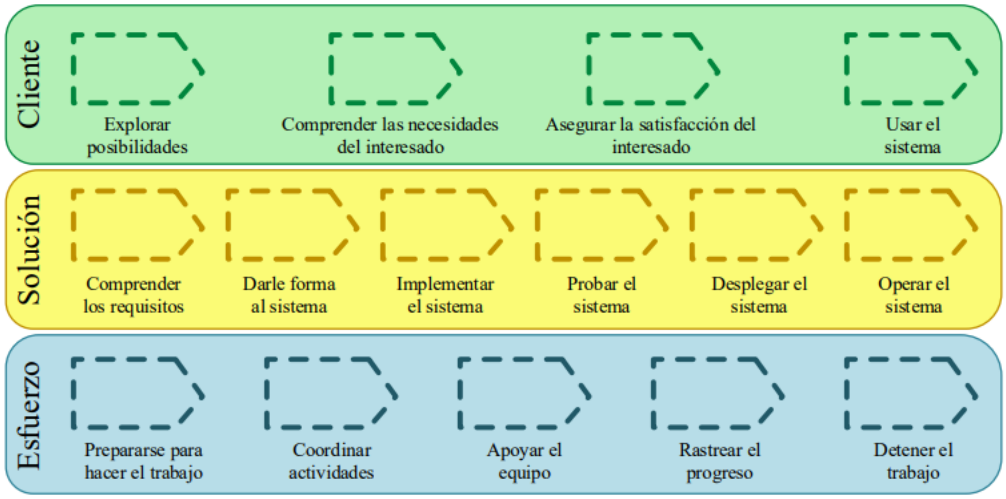

Figura 3. Espacios de actividad del núcleo (OMG, 2018)

Figura 4. Competencias del núcleo (OMG, 2018)

2.2 RUP Y LA GESTIÓN DE REQUISITOS, EL OBJETO DE ESTUDIO

El Proceso Unificado de Rational (RUP, por su sigla en inglés), es un proceso de desarrollo de sistemas que se fundamenta en principios de Ingeniería de Software. En RUP se define un proceso iterativo e incremental, centrado en la arquitectura y orientado por casos de uso. Además, RUP proporciona un enfoque disciplinado para asignar y administrar tareas y responsabilidades en una organización de desarrollo de software (Somerville, 2011). RUP se muestra de manera gráfica en la Figura 5.

2.2.1 Ciclo de vida

El ciclo de vida de RUP está estructurado en dos dimensiones, dinámica y estática: (i) estructura dinámica (horizontal), representa la dimensión del tiempo del proceso. Muestra el proceso expresado en ciclos, fases, iteraciones e hitos; (ii) estructura estática (vertical); describe cómo los elementos del proceso (actividades, disciplinas, artefactos y roles) se agrupan lógicamente.

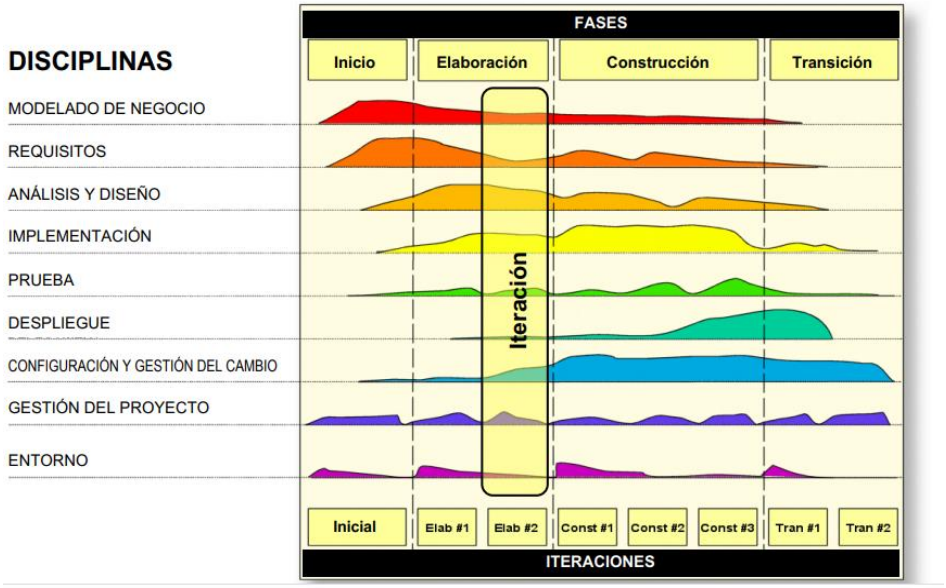

Figura 5. Ciclo de vida de RUP (Rational Software Company, 1998)

2.2.2 Mejores Prácticas de RUP (Rational Software Company, 1998)

- **Desarrollo Iterativo de Software:** Permite definir el problema, diseñar la solución, construir el software y probar el producto al final en iteraciones que liberan incrementos.
- **Arquitectura Basada en Componentes:** RUP se enfoca en el desarrollo temprano y la referencia de una arquitectura ejecutable robusta, antes de comprometer recursos para el desarrollo a gran escala.
- **Modelado Visual de Software:** RUP muestra cómo modelar visualmente el software para capturar la estructura y el comportamiento de arquitecturas y componentes.
- **Verificación de la Calidad del Software:** RUP incorpora actividades para planificar, diseñar, implementar, ejecutar y evaluar pruebas orientadas al aseguramiento de la calidad del proceso y del producto.
- **Gestión de Control de Cambios:** RUP define métodos para controlar, rastrear y monitorear los cambios para permitir un desarrollo iterativo exitoso.
- **Gestión de Requisitos de Software:** RUP describe cómo obtener, organizar y documentar la funcionalidad y restricciones requeridas; rastrear y documentar las compensaciones y decisiones; y capturar y comunicar fácilmente los requisitos comerciales. Las nociones de casos de uso y escenarios proscritos en el proceso han demostrado ser una excelente manera de capturar los requisitos funcionales y garantizar que éstos impulsen el diseño, la implementación y las pruebas de software, lo que hace más probable que el sistema final satisfaga las necesidades del usuario final.

2.3 MODELO PARA LA DEFINICIÓN DE PRÁCTICAS EN INGENIERÍA DE SOFTWARE (BARÓN, 2019)

Este es un modelo para la definición unificada y carente de ambigüedad de la práctica como constructo teórico en Ingeniería de Software. Los componentes se articulan de manera sistémica en el modelo y cumplen funciones específicas que se orientan a definir prácticas bien formadas y nombradas. Los componentes del modelo se muestran en la Figura 6 y se describen más adelante.

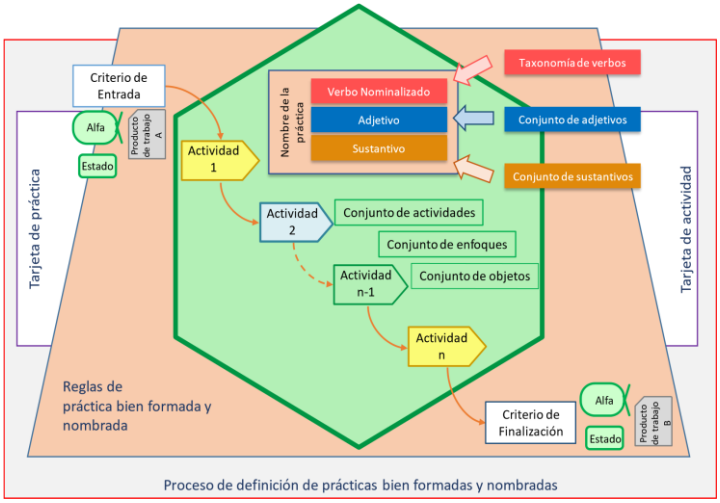

Figura 6. Componentes del modelo (Barón, 2019)

2.3.1 Criterio de entrada de la práctica

Permite determinar las condiciones que se deben cumplir para iniciar la práctica. El criterio de entrada se expresa en términos de sustantivos que pueden ser alfas y sub-alfas y un estado parcial o completo. El producto de trabajo es la evidencia de que un alfa se encuentra total o parcialmente en un estado.

2.3.2 Criterio de finalización de la práctica

Permite determinar las condiciones que se deben cumplir para terminar exitosamente la práctica. El criterio de finalización se expresa en términos de un sustantivo que puede ser alfa o sub-alfa y un estado parcial o completo. El producto de trabajo es la evidencia de que un alfa se encuentra total o parcialmente en un estado.

2.3.3 Conjunto de actividades

Permite definir el proceso de aplicación de la práctica. Entre actividades existen relaciones de secuencias y de transferencia de recursos. Cada actividad del conjunto de actividades de la práctica se conforma con acciones menores que se denominan tarea. La actividad tiene criterio de entrada y criterio de finalización que determinan las condiciones de inicio y finalización exitosa de la actividad.

2.3.4 Flujo de actividades

Permite definir el orden de ejecución de las actividades y la transferencia de recursos entre ellas.

2.3.5 Reglas de práctica bien formada

Una práctica se define como bien formada si el conjunto de actividades cumple con las reglas de coherencia, consistencia y suficiencia.

- **Regla de coherencia:** un conjunto de actividades de una práctica es coherente si el cumplimiento del criterio de finalización de cada actividad aporta en el progreso del sustantivo hacia el criterio de finalización de la práctica.
- **Reglas de consistencia:** un conjunto de actividades de una práctica es consistente si: (i) R1, existe al menos una actividad cuyo criterio de entrada es igual al criterio de entrada de la práctica; (ii) R2, existe al menos una actividad cuyo criterio de finalización es igual al criterio de finalización de la práctica; (iii) R3, para cada actividad del conjunto de actividades de la práctica, su criterio de entrada es igual al criterio de finalización de al menos otra actividad o al criterio de entrada de la práctica; (iv) R4, para cada actividad del conjunto de actividades de la práctica, su criterio de finalización es igual al criterio de entrada de al menos otra actividad o al criterio de finalización de la práctica. Las reglas R3 y R4 son válidas excluyendo las actividades que cumplen las reglas R1 y R2.
- **Regla de suficiencia:** un conjunto de actividades de una práctica es suficiente si el cumplimiento del criterio de finalización de cada actividad permite el progreso sustantivo hasta lograr el cumplimiento del criterio de finalización de la práctica.

2.3.6 Reglas de práctica bien nombrada

Una práctica es bien nombrada si su nombre incluye un verbo nominalizado, que indica lo que se hace con la práctica, un adjetivo que indica cómo se hace y un sustantivo que indica el objeto sobre el cual se aplica la práctica. Para definir los elementos que conforman el nombre de la práctica, en el modelo se propone una taxonomía de verbos nominalizados (ver tabla 1), un conjunto de adjetivos (ver tabla 2) y un conjunto de sustantivos (ver tabla 3).

Tabla 1. Taxonomía de verbos nominalizados propia de requisitos (Barón 2019)

Alfa	**Requisitos**		
Estado	Concebido Acotado Coherente	Aceptable	Tratado Cumplido
Taxonomía de verbos	**Análisis**	**Verificación**	**Verificación**
	Recolección	Aprobación	Aprobación
	Especificación	Aceptación	Aceptación
	Educción	Educción	Educción
	Negociación	Gestión	Gestión
	Priorización		
	Elicitación		

Tabla 2. Conjunto consolidado de adjetivos (Barón, 2019)

Conjunto consolidado de adjetivos			
Ágil	Colaborativo	Evolutivo	Probado
Automatizado	Compartido	Formal	Progresivo
Auto-organizado	Concurrente	Funcional	Prospectivo
Basado en actividades	Conjunto	Global	Repetible
Basado en casos de uso	Continuo	Incremental	Reutilizable
Basado en componentes	Cualitativo	Integrado	Sistemático
Basado en escenarios	Cuantitativo	Interdisciplinario	Sistémico
Basado en riesgos	Dirigido por pruebas	Iterativo	Temprano
Basado en valor	Disciplinado	Metódico	Valor compartido
Bien organizado	Empírico	Orientado al producto	Visual
Causal	Escalonado	Orientado por objetos	
Centrado en la arquitectura	Estructurado	Personalizado	

Tabla 3. Conjunto de sustantivos (Barón, 2019)

Alfas	**Sub-alfas**
Interesados	Representante de los interesados
Oportunidad	Necesidad
Requisitos	Ítem de Requisito
Sistema de software	Elemento del Sistema de Software
Equipo	Miembro del Equipo

Las reglas que rigen el proceso de nombramiento de una práctica son: (i) R1, el verbo nominalizado que se utiliza para nombrar la práctica es el verbo nominalizado que integra el conjunto de actividades que permite conducir el progreso del sustantivo hasta alcanzar el estado que se indica en el criterio de finalización de la práctica; (ii) R2, el adjetivo que se utiliza para nombrar la práctica es el adjetivo que integra los enfoques que se utilizan para realizar cada una de las actividades de la práctica; (iii) R3, el sustantivo que se utiliza para nombrar la práctica es el sustantivo que se indica en el criterio de finalización de la práctica.

2.3.7 Tarjetas de práctica y actividad

La tarjeta de práctica permite visualizar los elementos esenciales que describen la práctica de software. De la misma manera, la tarjeta de actividad permite visualizar los elementos esenciales que describen la actividad. Se utilizan para facilitar el seguimiento a la creación o conversión de prácticas bien formadas y nombradas. En la Figura 7 y Figura 8 se muestran las tarjetas de práctica y actividad respectivamente.

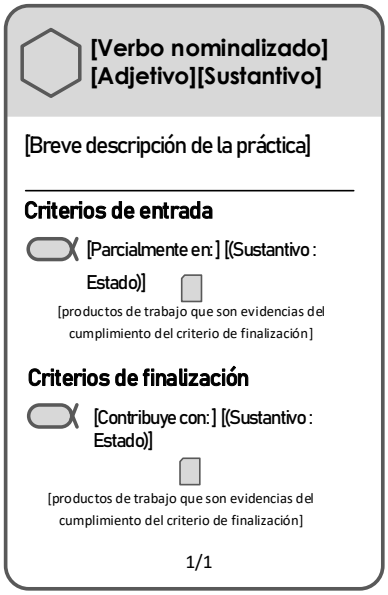

Figura 7. Tarjeta de práctica.

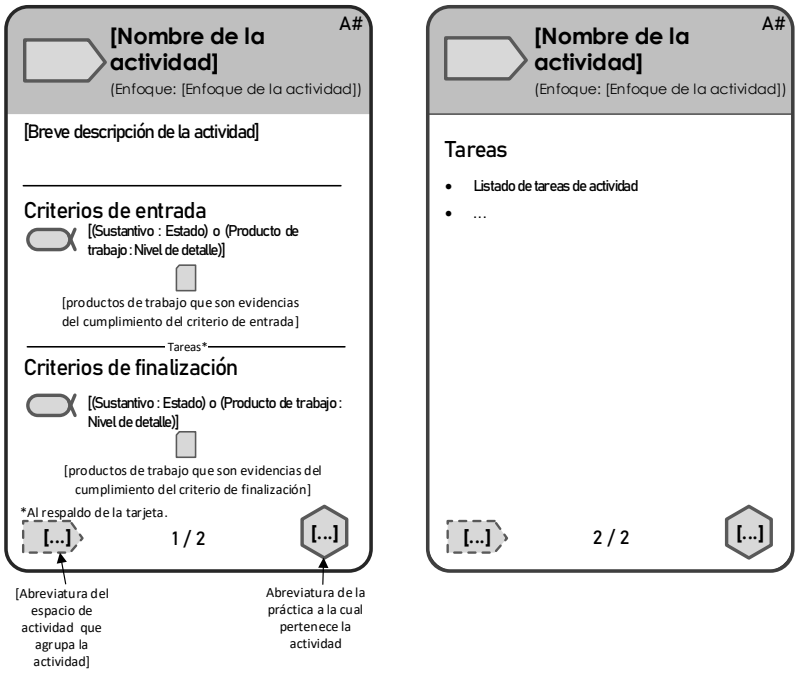

Figura 8. Tarjeta de actividad

2.3.8 Proceso de definición de prácticas bien formadas y nombradas

Los componentes del modelo se integran de manera sistémica para definir un proceso que permite guiar la definición de prácticas bien formadas y nombradas. Este proceso se puede aplicar en la creación de nuevas prácticas de Ingeniería de Software bien formadas y nombradas y en la conversión de prácticas existentes en prácticas bien formadas y nombradas. En la Figura 9 se muestra el proceso de definición de prácticas bien formadas y nombradas.

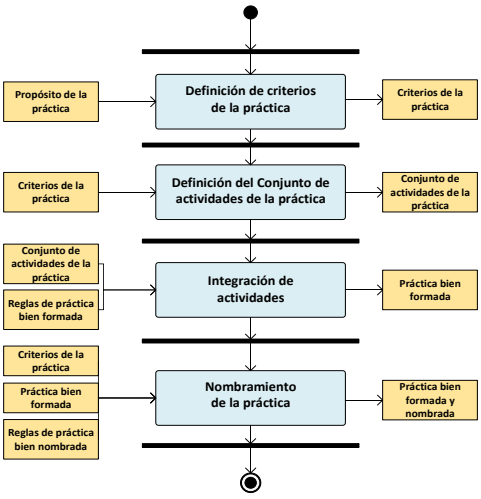

Figura 9. Proceso de definición de prácticas bien formadas y nombradas (Barón, 2019)

3. ESTUDIO SOBRE LAS FORMAS DE REPRESENTACIÓN Y DESCRIPCIÓN DE LA PGR-RUP

Este estudio permite establecer la manera como diferentes autores representan y describen la práctica Gestión de Requisitos de RUP. Este estudio consiste en una Revisión Sistemática de Literatura (RSL) que busca identificar y analizar estudios relevantes para la investigación. Como resultado, se obtiene un reporte de análisis de los estudios relevantes. Este estudio se compone de las actividades planeación, realización y reporte.

3.1. PLANEACIÓN DE LA RSL

En esta actividad se definen los elementos que orientan la RSL, se omite el uso de ventana temporal para criterios de inclusión y exclusión, ya que existen en la literatura de la Ingeniería de Software, artículos y textos clásicos que aún se utilizan para la enseñanza y aprendizaje de métodos de Ingeniería de Software como RUP. Las fuentes digitales de estudios utilizadas fueron consideradas pertinentes y suficientes para la investigación, dado que, son fuentes reconocidas en el contexto académico e industrial de la Ingeniería de Software. Los elementos que orientan la RSL se muestran en la tabla 4.

Tabla 4. RSL sobre la PGR-RUP

Propósito	Pregunta	Cadena de búsqueda
Establecer la manera como diferentes autores representan y describen la práctica Gestión de Requisitos de RUP.	¿Cómo diferentes autores representan y describen la práctica Gestión de Requisitos de RUP?	(("Gestión de Requisitos" **OR** "Requirement Managment") **AND** ("RUP" **OR** "Rational Unified Process"))
Fuentes de estudios		
- EBSCO - *Science Direct* - *Scopus* - Google académico - IEEE		
Criterios de inclusión		
- Prácticas de Gestión de Requisitos - Prácticas de ingeniería de software - Práctica de RUP - Representación de la práctica		
Criterios de exclusión		
- Estudios que orienten a RUP como una metodología ágil. - Estudios sobre metodologías ágiles.		

3.2. REALIZACIÓN DE LA RSL

En esta actividad, se ejecutan las siguientes tareas.

3.2.1. Aplicar la cadena de búsqueda

La cadena de búsqueda que se define en la actividad de planeación, se aplica a los buscadores de cada una de las fuentes digitales de estudios seleccionadas y se obtiene un universo de estudios. Un extracto del universo de estudios se relaciona en la tabla 5.

Tabla 5. Universo de estudios

Autor	Título
(Arias *et al*., 2018)	A Framework Managing Requirements of Software Product Line
(Kroll & Kruchten, 2003)	The Rational Unified Process Made Easy - A Practitioner's Guide to the RUP
(Baruah, 2015)	Requirement Management in Agile Software Environment
(Galic *et al*., 2006)	Using a Single Business Pattern with the Rational Unified Process (RUP)
(Leyva & González, 2006)	Una Adaptación del Proceso Unificado de Desarrollo para la Creación de Portales Basados en Joomla
(Jones, 2010)	Software Engineering Best Practices
(*Rational Software Company*, 1998)	Rational Unified Process
(Passing, 2015)	Requirements Engineering in the Rational Unified Process
(Jacobson, 2000)	El Proceso Unificado de Desarrollo de Software
(Metzner & Niño, 2016)	El Proceso de Desarrollo RUP-GDIS
(Rueda, 2006)	Aplicación de la Metodología RUP para el Desarrollo Rápido de Aplicaciones Basado en el Estándar J2EE
(Vera, 2019)	Análisis de la Metodología RUP en el Desarrollo de Software Académico Mediante la Herramienta Django
(Londoño, 2008)	Análisis de la Ingeniería de Requisitos Orientada por Aspectos Según la Industria del Software
(Ambler, 2005)	A Manager's Introduction to the Rational Unified Process (RUP)
(Arias, 2007)	La ingeniería de Requerimientos y su importancia en el Desarrollo de Proyectos de Software
(Carrizo & Ortiz, 2015)	Modelos del Proceso de Educción de Requisitos: un Mapeo Sistémico
(Tabares *et al.*, 2007)	Un Método para la Trazabilidad de Requisitos en el Proceso Unificado de Desarrollo
(Jamal et al., 2018)	STORE: Security Threat Oriented Requirements Engineering
(Bonfante, 2017)	Reference Model for the Integration of Business Modeling to Requirements Engineering: A Proposal from the Software Industry
(Wellsandt, 2014)	Qualitative Comparison of requirements Elicitation Techniques that are Used to Collect Feedback Information About Product Use
(García, 2018)	Experience of Using a Game of Improving Learning in Software Requirements Elicitation
(Alshazly, 2014)	Detecting Defects in Software Requirements Specification
(Maalem, 2016)	Challenge of Validation in Requirements Engineering
(Jiménez, 2016)	Modelo para la Definición Unificada de la Práctica como Constructo Teórico en Ingeniería de Software

3.2.2. Aplicar los criterios de inclusión y exclusión

Se realiza mediante 3 vueltas de análisis. Se aplica el concepto de semáforo de la siguiente manera: verde, el estudio se incluye en la investigación; rojo, el estudio se descarta para la investigación; amarillo, el estudio se somete a un análisis más riguroso en la siguiente vuelta.

En la primera vuelta se aplican los criterios de inclusión y de exclusión al título de cada uno de los estudios resultantes. En la segunda vuelta se aplican los criterios de inclusión y de exclusión al resumen de cada uno de los estudios que resultaron en amarillo en el análisis de la primera vuelta. En la tercera vuelta se aplican los criterios de inclusión y de exclusión a la introducción de cada uno de los estudios que resultaron en amarillo en el análisis de la segunda vuelta. Para esta vuelta, los estudios que resultaron en amarillo se mantienen en la investigación, para que en un análisis posterior se defina su inclusión o exclusión. Finalmente, se obtienen los estudios relevantes que pasan a la tarea de análisis. En la tabla 6 se muestra el resultado de esta tarea.

Tabla 6. Estudios relevantes para la investigación

Autor	Título
(Rational Software Company, 1998)	Rational Unified Process
(Jacobson, 2000)	El Proceso Unificado de Desarrollo de Software
(Kroll & Kruchten, 2003)	The Rational Unified Process Made Easy - A Practitioner's Guide to the RUP
(Ambler, 2005)	A Manager's Introduction to the Rational Unified Process (RUP)
(Metzner & Niño, 2016)	El Proceso de Desarrollo RUP-GDIS
(Vera, 2019)	Análisis de la Metodología RUP en el Desarrollo de Software Académico Mediante la Herramienta Django
(Jiménez, 2016)	Modelo para la Definición Unificada de la Práctica como Constructo Teórico en Ingeniería de Software

3.2.3. Análisis de los estudios relevantes que describen la PGR–RUP

Teniendo en cuenta los estudios relevantes que resultan de la tarea anterior, se realiza un análisis sobre los mecanismos de representación y descripción de la PGR-RUP. Este análisis permite identificar los componentes de la práctica a partir de la manera cómo diferentes autores la representan y describen. Los criterios que se utilizan en el análisis de los estudios relevantes son los que se identifican en (Barón, 2019) como componentes de una práctica. La nomenclatura utilizada para representar los resultados del análisis es:

N: No se encuentra el componente.

P: Se encuentra el componente parcialmente.

C: Se encuentra el componente completo.

Análisis orientado a la práctica bien formada:

Busca identificar los componentes que constituyen una práctica bien formada, los componentes son los propuestos en (Barón, 2019). El resultado de este análisis se muestra en la tabla 7 y se explica más adelante.

Tabla 7. Análisis de los mecanismos de representación y descripción de la PGR-RUP

Estudio relevante	Componentes de la práctica bien formada								
	Criterios de entrada		Criterios de finalización		Conjunto de actividades	Actividades			Flujo de actividades
	Alfa o sub-alfa	Estado	Alfa o sub-alfa	Estado		Criterio de entrada	Criterio de finalización	Enfoque	
(Rational Software Company, 1998)	P	P	P	P	N	N	P	N	P
(Metzner y Niño, 2016)	P	P	P	P	P	P	P	N	P
(Vera *et al.*, 2019)	N	N	P	P	N	N	N	N	N
(Jacobson *et al.*,2000)	P	C	P	C	C	P	P	N	C
(Jiménez, 2016)	C	N	C	N	C	N	N	C	N

Como se aprecia en las filas de la tabla, ninguno de los estudios relevantes cumple con la totalidad de reglas de práctica bien formada. Por ejemplo, en (Metzner & Niño, 2016) se pueden encontrar implícitos los objetos que la práctica aborda; sin embargo, no se advierten estados. Además, el estudio contiene una serie de tablas generales de las cuales se pueden abstraer las actividades, los criterios de entrada, los criterios de finalización y el flujo de actividades. Sin embargo, el nivel de detalle no es suficiente para facilitar la aplicación de la práctica.

Análisis orientado a la práctica bien nombrada:

Busca identificar los componentes que constituyen una práctica bien nombrada. El resultado de este análisis se muestra en la tabla 8 y se explica más adelante.

Tabla 8. Análisis orientado a una práctica bien nombrada

Estudio relevante	Componentes de práctica bien nombrada		
	Nombre		
	Verbo Nominalizado	Adjetivo	Sustantivo
(Rational Software Company, 1998)	C	N	C
(Metzner y Niño, 2016)	N	N	C
(Vera et al., 2019)	C	N	C
(Jacobson et al., 2000)	C	N	C
(Jiménez, 2016)	C	N	C

Como se aprecia en las filas de la tabla, ninguno de los estudios relevantes cumple con la totalidad de reglas para que sea una práctica bien nombrada. Por ejemplo, en (Rational Software Company, 1998) se puede encontrar un verbo nominalizado, Gestión, un sustantivo, Requisitos; sin embargo, el nombre de esta práctica no tiene un adjetivo que indique la forma como se realiza la práctica.

Análisis orientado a los mecanismos de aplicación seguimiento y evaluación:

Busca identificar, en los estudios que son objeto de análisis, componentes que faciliten aplicar, seguir y evaluar la práctica. El resultado de este análisis se muestra en la tabla 9 y se explica más adelante.

Tabla 9. Análisis orientado a los mecanismos de administración de práctica

Estudios relevantes	Mecanismos de administración de práctica		
	Aplicación	Seguimiento	Evaluación
(Rational Software Company, 1998)	P	N	N
(Metzner y Niño, 2016)	P	P	N
(Vera et al., 2019)	P	N	N
(Jacobson et al., 2000)	P	P	N
(Jiménez, 2016)	P	P	N

Como se aprecia en las filas de la tabla, ninguno de los estudios analizados, integra componentes que permitan realizar de manera completa aplicar, seguir y evaluar la PGR-RUP. Por ejemplo, en (Jiménez, 2016) haciendo uso de las tablas y los gráficos se puede inferir la forma de aplicación y seguimiento de la práctica, pero no se define algún mecanismo de evaluación.

4. CONVERSIÓN DE LA PGR-RUP EN UNA PRÁCTICA BIEN FORMADA Y NOMBRADA

Aquí se presenta la propuesta de (Jiménez, 2016) en la que se describe la práctica PGR-RUP utilizando el núcleo de Essence. Esta descripción es el insumo para aplicar el proceso de definición de prácticas bien formadas y nombradas que propone (Barón, 2019).

4.1 La PGR-RUP según (Jiménez, 2016)

Esta descripción se realiza a través tablas y gráficos. En la tabla 10 se presenta un extracto de la manera como (Jiménez, 2016) realiza la descripción de la PGR-RUP.

Tabla 10. Extracto de la tabla de componentes de la PGR-RUP en la fase Inicio (Jiménez, 2016)

Espacio de actividad	Alfa	Producto de trabajo	Actividad
Explorar posibilidades	Oportunidad	Visión	Desarrollar visión
	Oportunidad	Lista de riesgos	Identificar y evaluar los riesgos
	Oportunidad	Caso de negocio	Desarrollar caso de negocio
	Oportunidad, Interesados	Glosario de negocio	Capturar un vocabulario común de negocio
	Oportunidad, Interesados	Reglas del negocio	Mantener reglas de negocio
	Oportunidad, Interesados	Visión del negocio	Establecer y ajustar objetivos
	Oportunidad, Interesados	Evaluación de la organización objetivo	Evaluar organización objetivo
	Oportunidad, Interesados	Modelo de objetos del negocio, Trabajadores del negocio, entidades del negocio, realización de casos de uso de negocio	Encontrar las entidades y los trabajadores del negocio

4.2 Conversión de la PGR-RUP

Para llevar a cabo la conversión se aplica el proceso de definición de prácticas bien formadas y nombradas.

4.2.1 Actividad 1. Definición de criterios de la práctica

El proceso para la definición de criterios de la práctica se muestra en la Figura 10 y se describe más adelante.

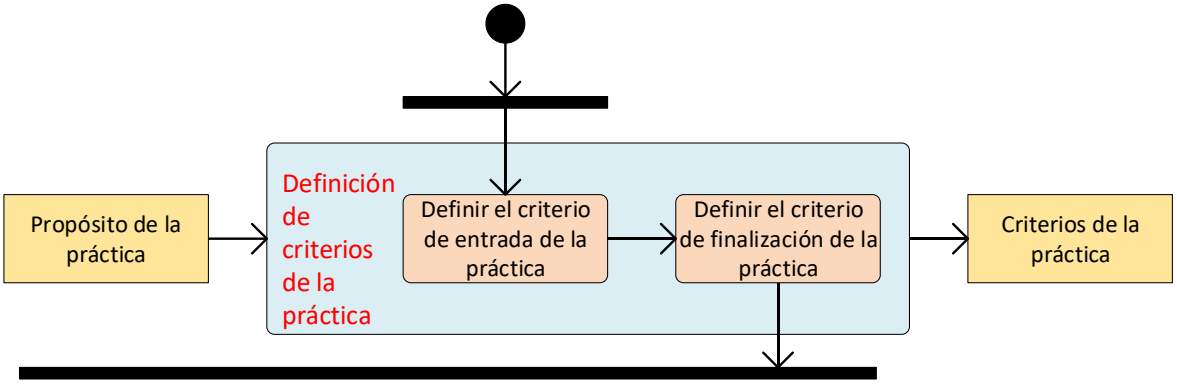

Figura 10. Definición de criterios de la práctica (Barón, 2019)

Definir criterio de entrada de la práctica

De acuerdo con el propósito de esta práctica, el criterio de entrada es (Oportunidad: Identificada), debido a que se ha identificado una oportunidad comercial, social o de negocio que podría abordarse con una solución basada en software.

Definir criterio de finalización de la práctica:

De acuerdo con el propósito de esta práctica, los criterios de finalización son (Requisitos: Dirigido) debido a que se han abordado suficientes requisitos para satisfacer la necesidad de un nuevo sistema de manera que sea aceptable para las partes interesadas y (Sistema software: Arquitectura seleccionada) debido a que se ha seleccionado una arquitectura que aborda el riesgo técnico clave y las restricciones organizativas aplicables.

4.2.2 Actividad 2. Definición del conjunto de actividades de la práctica

El proceso para la definición del conjunto de actividades de la práctica se muestra en la Figura 11 y se describe más adelante.

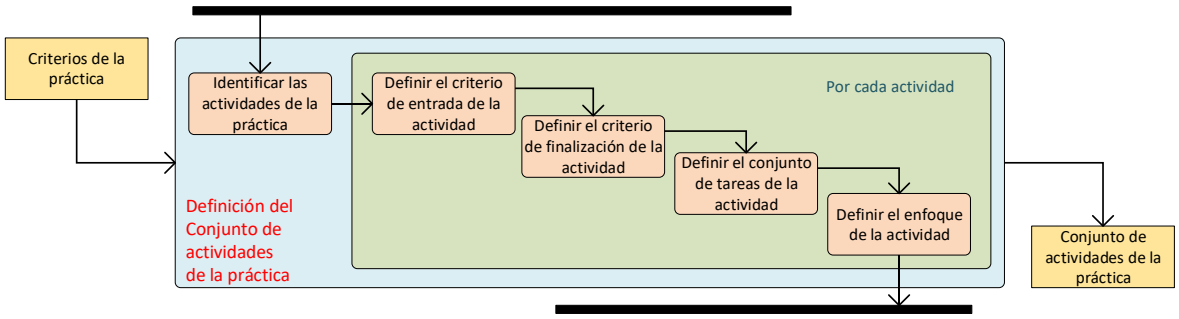

Figura 11. Definición del conjunto de actividades de la práctica (Barón, 2019)

Identificar las actividades de la práctica

Las actividades que conforman la práctica son: Crear visión, Identificar y proyectar el negocio, Formalizar el negocio, Determinar entidades y los trabajadores del negocio, Identificar actores y establecer la arquitectura de negocio, Evaluar viabilidad del negocio, Representar inicialmente los requisitos del negocio, Definir actores y detallar casos de uso, Especificar los requisitos del software, Estructurar el modelo de caso de uso, Priorizar casos de uso, Modelar prototipo de interfaz de usuario, Crear unidad de desarrollo, Diseñar prototipo de interfaz de usuario, Modelar clases, Modelar bases de datos, Diseñar clases, Diseñar bases de datos.

Definir el criterio de entrada de la actividad

Se define el criterio de entrada como condición para iniciar la actividad en términos de la dupla (Sustantivo: Estado) o (Producto de trabajo: Nivel de detalle). Las situaciones particulares que se contemplan para definir el criterio de entrada de la práctica se contemplan también para definir el criterio de entrada de la actividad. Los criterios de entrada se muestran en la tabla 11.

Definir el criterio de finalización de la actividad

Se define el criterio de finalización como condición que permite determinar la realización exitosa de la actividad en términos de la dupla (Sustantivo: Estado) o (Producto de trabajo: Nivel de detalle). La situación particular que se contempla para definir el criterio de finalización de la práctica, se contempla también para definir el criterio de finalización de la actividad. Los criterios de finalización se muestran en la tabla 11.

Definir el conjunto de tareas de la actividad:

Se relacionan las tareas que son parte de cada actividad. Las actividades con sus respectivas tareas se especifican en las tablas 12, 13 y 14 de acuerdo a la fase del ciclo de vida de RUP.

Tabla 11. Definición de criterios y enfoque de actividad

Actividad	Criterio de entrada		Criterio de Finalización		Enfoque
	Sustantivo	Estado	Sustantivo	Estado	
Crear visión	Oportunidad	Identificada	Oportunidad	Con solución requerida	Cualitativo
Identificar y proyectar el negocio	Oportunidad	Con solución requerida	Oportunidad	Con valor establecido	Incremental
Formalizar el negocio	Oportunidad	Con valor establecido	Oportunidad	Viable	Colaborativo
	Interesados	Reconocido	Interesados	Representado	Colaborativo
Determinar entidades y los trabajadores del negocio	Oportunidad	Con valor establecido	Oportunidad	Viable	Incremental
	Interesados	Reconocido	Interesados	Representado	Incremental
Identificar actores y establecer la arquitectura de negocio	Oportunidad	Con valor establecido	Oportunidad	Viable	Colaborativo
	Interesados	Representado	Interesados	Involucrado	Colaborativo
Evaluar viabilidad del negocio	Oportunidad	Con valor establecido	Oportunidad	Viable	Basado en riesgos
Representar inicialmente los requisitos del negocio	Requisitos	Concebido	Requisitos	Acotado	Incremental
Definir actores y detallar casos de uso	Requisitos	Concebido	Requisitos	Acotado	Incremental
	Interesados	Involucrado	Interesados	De acuerdo	Incremental
Especificar los requisitos del software	Requisitos	Acotado	Requisitos	Coherente	Incremental
Estructurar el modelo de caso de uso	Requisitos	Coherente	Requisitos	Aceptable	Estructurado
Priorizar casos de uso	Requisitos	Coherente	Requisitos	Aceptable	Colaborativo
Modelar prototipo de interfaz de usuario	Sistema software	Arquitectura seleccionada	Sistema software	Arquitectura seleccionada	Incremental
Crear unidad de desarrollo	Sistema software	Arquitectura seleccionada	Sistema software	Arquitectura seleccionada	Colaborativo
Diseñar prototipo de interfaz de usuario	Sistema software	Arquitectura seleccionada	Sistema software	Arquitectura seleccionada	Colaborativo
	Oportunidad	Viable	Oportunidad	Dirigido	Incremental
	Interesados	De acuerdo	Interesados	Satisfecho con la implementación	Colaborativo
	Requisitos	Aceptable	Requisitos	Dirigido	Incremental
Modelar clases	Sistema software	Arquitectura seleccionada	Sistema software	Arquitectura seleccionada	Incremental
Modelar bases de datos	Sistema software	Arquitectura seleccionada	Sistema software	Arquitectura seleccionada	Incremental
Diseñar clases	Sistema software	Arquitectura seleccionada	Sistema software	Arquitectura seleccionada	Colaborativo
	Requisitos	Aceptable	Requisitos	Dirigido	Incremental
Diseñar bases de datos	Sistema software	Arquitectura seleccionada	Sistema software	Arquitectura seleccionada	Colaborativo
	Requisitos	Aceptable	Requisitos	Dirigido	Incremental

Tabla 12. Definición de tareas de actividades en la fase de Inicio

Espacio de actividad	Actividad	Tareas
Explorar posibilidades	Crear visión	Identificar requisitos básicos del proyecto. Establecer características claves y limitaciones principales.
	Identificar y proyectar el negocio	Identificar y evaluar riesgos. Modelar caso de negocio.
	Formalizar el negocio	Redactar vocabulario común. Establecer o mantener reglas de negocio. Establecer y ajustar objetivos.
	Determinar entidades y los trabajadores del negocio	Crear modelo de objetos del negocio. Identificar trabajadores del negocio.
Comprender las necesidades de los interesados	Identificar actores y establecer la arquitectura de negocio	Identificar los actores del negocio. Estructurar el modelo de caso de uso. Definir la arquitectura de negocio.
	Evaluar viabilidad del negocio	Evaluar viabilidad con base en riesgos, caso de uso y visión del negocio.
Comprender los requisitos	Representar inicialmente los requisitos del negocio	Crear modelo de análisis y especificaciones.
	Definir actores y detallar casos de uso	Establecer los actores del negocio. Detallar caso de uso.
	Especificar los requisitos del software	Especificar los requisitos.
	Estructurar el modelo de los casos de uso	Estructurar los casos de uso utilizando la especificación de los requisitos y el modelo de casos uso.
Darle forma al sistema	Priorizar casos de uso	Priorizar casos de uso estructurados.
	Modelar prototipo de interfaz de usuario	Modelar prototipo utilizando lista priorizada de casos de uso.
Implementar el sistema	Crear unidad de desarrollo	Identificar unidad de desarrollo. Establecer unidad de desarrollo.
	Diseñar prototipo de interfaz de usuario	Diseñar prototipo utilizando el modelo del prototipo.

Tabla 13. Definición de las tareas de actividades en la fase de Elaboración

Espacio de actividad	Actividad	Tareas
Darle forma al sistema	Modelar clases	Modelar clases que intervienen en el software.
	Modelar bases de datos	Modelar bases de datos que intervienen en el software.

Tabla 14. Definición de las tareas de las actividades en la fase de Construcción

Espacio de actividad	Actividad	Tareas
Implementar el sistema	Diseñar clases	Diseñar clases que intervienen en el software.
	Diseñar bases de datos	Diseñar bases de datos que intervienen en el software.

Definir el enfoque de la actividad:

Se define la manera como se aborda la actividad en términos de un adjetivo; utilizando la información que aporta la fuente de la práctica o el conjunto de adjetivos que hacen parte del modelo. Los enfoques de cada actividad se muestran en la tabla 11.

4.2.3 Actividad 3. Integración de actividades

El proceso para el nombramiento de la práctica se muestra en la Figura 12 y se describe más adelante.

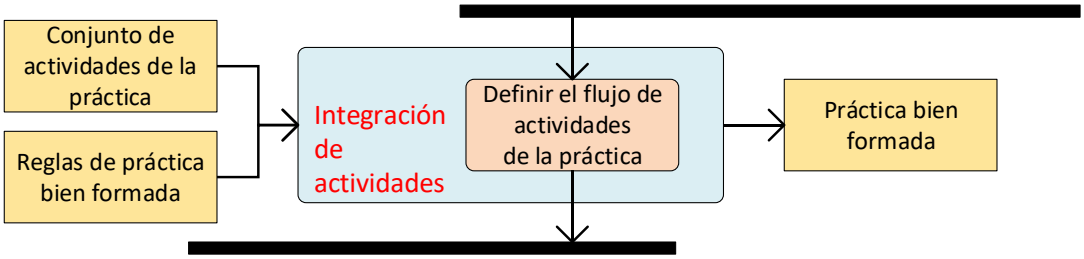

Figura 12. Integración de actividades (Barón, 2019)

Definir el flujo de actividades de la práctica

Se define el orden de ejecución de las actividades y la transferencia de recursos entre ellas aplicando las reglas de práctica bien formada, es decir, las reglas de coherencia, consistencia y suficiencia. En la Figura 13 se presenta el flujo de actividades de la práctica.

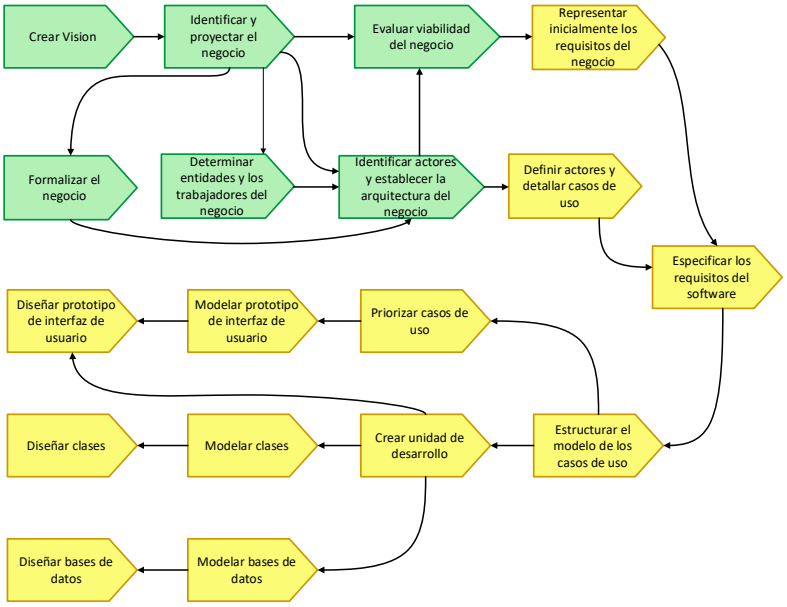

Figura 13. Flujo de actividades de la práctica

4.2.4 Actividad 4. Nombramiento de la práctica

El proceso para el nombramiento de la práctica se muestra en la Figura 14 y se describe más adelante.

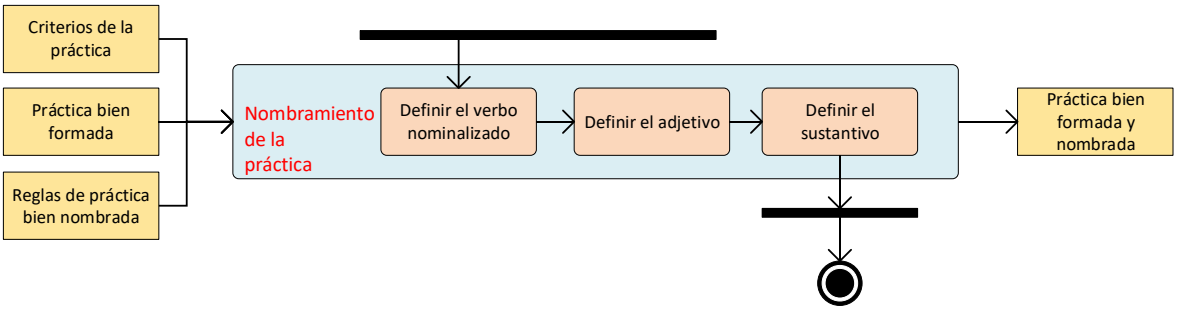

Figura 14. Nombramiento de la práctica (Barón, 2019)

Definir el verbo nominalizado

En el caso de la PGR-RUP, el verbo normalizado que resume el conjunto de actividades que conduce a la continua construcción y progreso del sustantivo requisitos hasta alcanzar el estado dirigido es Gestión. El verbo se puede identificar en la taxonomía de verbos nominalizados propia de requisitos (ver tabla 1).

Definir el adjetivo

En el caso de la PGR-RUP, los adjetivos que integran los enfoques usados para realizar las actividades de la práctica son Iterativo e Incremental. Iterativo debido a la naturaleza de RUP e Incremental debido a que la mayoría de las actividades desarrolladas tienen este enfoque. Los adjetivos se pueden identificar en la taxonomía de adjetivos (ver tabla 2).

Definir el sustantivo

Es el sustantivo que se indica en el criterio de finalización de la PGR-RUP. Requisitos es el sustantivo que, asociado con el estado dirigido, permite la realización exitosa de la práctica (ver tabla 3). De acuerdo con lo anterior, el nombre de la práctica esencializada es: Gestión Iterativa e Incremental de Requisitos de RUP (GIIR-RUP).

La práctica GIIR-RUP se compone de 18 actividades, cada una de ellas con su respectiva tarjeta. En la Figura 15 se presenta la tarjeta de práctica, en la Figura 16 se presenta la tarjeta de la actividad inicial y en la Figura 17 se presenta la tarjeta de la actividad final.

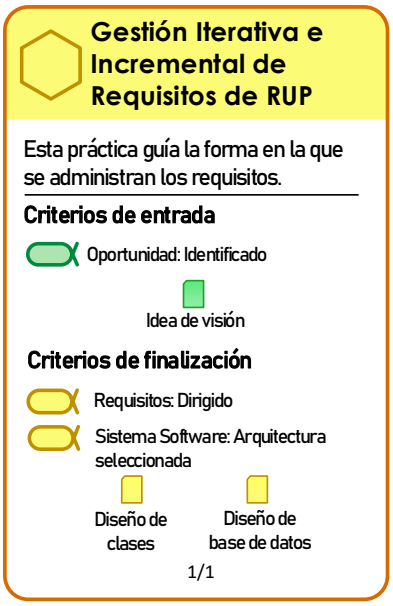

Figura 15. Tarjeta de práctica GIIR-RUP

Figura 16. Tarjeta de la actividad inicial de la práctica GIIR-RUP

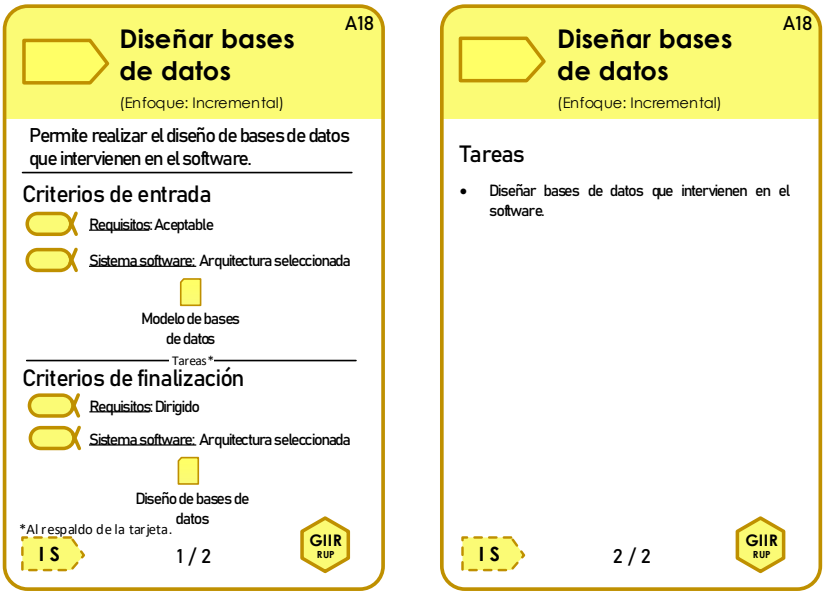

Figura 17. Tarjeta de la actividad final de la práctica GIIR-RUP

5. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se presentó el proceso de esencialización de la PGR-RUP aplicando el modelo que propone (Barón, 2019). La PGR-RUP esencializada se denomina GIIR-RUP. En la práctica GIIR-RUP se identifican y definen claramente sus componentes. De esta manera, la práctica GIIR-RUP facilita a los practicantes aplicar, seguir y evaluar la práctica.

La RSL que orientó el estudio sobre las formas de representación y descripción de la PGR-RUP, se constituyó como una estrategia eficiente para, de manera estructurada e imparcial, identificar y analizar los estudios relevantes para la investigación y para reportar los resultados del estudio.

El Modelo para la Definición de Prácticas en Ingeniería de Software que propone (Barón, 2019), se constituyó en un mecanismo adecuado para definir la práctica GIIR-RUP como una práctica bien formada y nombrada. El modelo permitió identificar y definir los componentes de la práctica GIIR-RUP. Así, se facilita a los practicantes, aplicar, seguir y evaluar la práctica GIIR-RUP.

Essence se constituyó en el marco de trabajo adecuado, que integrado al modelo de (Barón, 2019), permite identificar y definir los componentes de la práctica GIIR-RUP empleando los elementos del núcleo.

Como trabajo futuro se proyecta la construcción de un software que soporte la aplicación, seguimiento y evaluación de la práctica GIIR-RUP. Se proyecta también, validar la práctica GIIR-RUP mediante su aplicación en contextos industriales y académicos como estrategia para promover su uso.

REFERENCIAS

Barón, A. (2019). *Modelo para la Definición Unificada de Prácticas como Constructo Teórico en Ingeniería de Software* (Tesis de Doctorado). Universidad Nacional, Medellin, Colombia.

Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software*. Madrid, España, Addison Wesley.

Jiménez, L. (2016). *Representación en el Núcleo de Semat de Prácticas de Métodos de Desarrollo Basados en Planes* (Tesis de Maestría). Universidad de Nacional de Colombia, Medellín, Colombia.

Londoño, L. F., Anaya, R., & Tabares, M. S. (2013). ANÁLISIS DE LA INGENIERÍA DE REQUISITOS ORIENTADA POR ASPECTOS SEGÚN LA INDUSTRIA DEL SOFTWARE. *Revista EIA*, 5(9), 43-52. Recuperado de https://revistas.eia.edu.co/index.php/reveia/article/view/197

Metzner, C., & Niño, N. (2016). El Proceso de Desarrollo RUP-GDIS. *Revista Venezolana de Computación*, 3(1), 13-22. Recuperado de http://svc.net.ve/ReVeCom/Vol03-No01/ReVeCom-vol03-no01-v1.1.pdf

Object Management Group. (2018). *About the essence specification version 1.2*: *Kernel and Language for Software Engineering*. Recuperado de: https://www.omg.org/spec/Essence/1.2/PDF

Rational Software Company. (1998). *Rational Unified Process: Best practices for software development teams*. Recuperado de https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

Somervile, I. (2011). *Sotware Engineering*. Boston, Massachusetts. Estados Unidos. Addison-Wesley

Tabares, M. S., Barrera, A. F., Arroyave, J. D., & Pineda, J. D. (2013). UN MÉTODO PARA LA TRAZABILIDAD DE REQUISITOS EN EL PROCESO UNIFICADO DE DESARROLLO. *Revista EIA*, 4(8), 69-82. Recuperado de https://revistas.eia.edu.co/index.php/reveia/article/view/186

Vera, D. A., Córdova, L. C., López Bermúdez, R. M., & Pacheco Mendoza, S. R. (2019). *Análisis de la metodología RUP en el desarrollo de software académico mediante la herramienta DJANGO*. *RECIMUNDO*, 3(2), 964-979. Recuperado de https://www.recimundo.com/index.php/es/article/view/486/629

NOTAS BIOGRÁFICAS

Jhonny Edison Chapal Vallejo es egresado del programa de Ingeniería de Sistemas de la Universidad de Nariño, es miembro del Grupo de Investigación Galeras.Net del Departamento de Sistemas de la Universidad de Nariño, actualmente se encuentra realizando el trabajo de grado bajo la modalidad de investigación en la línea de Software y Manejo de Información, la investigación está adscrita al Grupo de Investigación Galeras.Net.

John Snyder Guerrero Riascos es egresado del programa de Ingeniería de Sistemas de la Universidad de Nariño, es miembro del Grupo de Investigación Galeras.Net del Departamento de Sistemas de la Universidad de Nariño, actualmente se encuentra realizando el trabajo de grado bajo la modalidad de investigación en la línea de Software y Manejo de Información, la investigación está adscrita al Grupo de Investigación Galeras.Net.

Alexander Barón Salazar es Profesor Asociado del Departamento de Sistemas de la Universidad de Nariño y miembro del Grupo de Investigación Galeras.Net; su área de investigación es la Ingeniería de software. Su expediente académico se puede resumir así: Ingeniero de Sistemas; Especialista en Ingeniería de Software (Universidad Industrial de Santander); Especialista en Docencia Universitaria (Universidad de Nariño); Especialista en Desarrollo de Software (Universidad Eafit); Magister en Ingeniería Informática (Universidad Eafit) y Doctor en Ingeniería - Sistemas e Informática (Universidad Nacional de Colombia en la Sede Medellín). Como resultado de sus investigaciones ha desarrollado productos de software y ha sido ponente en eventos nacionales e internacionales. También, ha publicado libros y artículos en revistas científicas especializadas.

Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-Compartirlgual 4.0 Internacional.