Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

UTN.BA
ESCUELA DE
POSGRADO

TESIS DE MAESTRÍA

Ingeniería en Sistemas de Información

DEFINICIÓN DE RIESGOS PARA EL PROCESO DE DESPLIEGUE DE SISTEMAS SOFTWARE

Alumno: Esp. Felipe David Ortiz

Directora de la Carrera: Dra. María Florencia Pollo Cattaneo
Directora: Mg. Marisa Daniela Panizzi
Co-Director: Mg. Rodolfo Alfredo Bertone

CABA, 2020

RESUMEN

El despliegue es el proceso por el cual se hace la transferencia del sistema de software a la empresa cliente. Un riesgo es la probabilidad de que ocurra una pérdida y en un proyecto de software, podría presentarse mediante la disminución de la calidad del producto software, el aumento de los costos, el retraso en la finalización o una falla, entre otras pérdidas.

En la actualidad, una condición imperante para el crecimiento de la industria del software es que las empresas ofrezcan productos de mayor calidad, que satisfaga las demandas y exigencias del cliente, pero sobre todo que genere confianza al momento de su uso. Esto se logra mediante la aplicación de modelos y metodologías de gestión de riesgos reconocidos internacionalmente. Sin embargo, en Argentina, la industria del software está compuesta mayoritariamente por pequeñas y medianas empresas (PyMEs), las cuales representan casi el 80% del sector (esto las constituye como un eslabón fundamental en la economía del país), pero en las mismas se hace muy difícil de implementar este tipo de modelos y metodologías debido a que implica una gran inversión en dinero, tiempo y recursos.

En este contexto y entendiendo la necesidad de llevar adelante iniciativas que contribuyan con el desarrollo y mejore la competitividad de dichas empresas, el presente trabajo de tesis tiene como objetivo en primer lugar, aportar a los Ingenieros de Software involucrados en proyectos de desarrollo de software en PyMEs, un conjunto de riesgos a considerar en el proceso de despliegue de los sistemas de software que cubren tres aspectos, el proceso en sí, el producto a implantar y el peopleware que participa en el proceso, así como también los procedimientos para su prevención, mitigación y/o transferencia los cuales también se adaptan a las características de las PyMEs. En segundo y no por tratarse de menor importancia que el primero, fortalecer el proceso de despliegue de sistemas de software que, por considerarse el último eslabón de la cadena de producción de software, en muchas ocasiones no es tan abordado.

Por último, se desarrollan dos estudios de caso aplicando el conjunto de riesgos propuestos, así como también los procedimientos para su prevención, mitigación y/o transferencia en dos PyMEs de desarrollo de software de Argentina. Esto permitió confirmar la viabilidad de la propuesta, así como también robustecer el proceso de despliegue de sistemas de software.

Palabras clave: gestión de riesgos, despliegue de sistemas de software, PyMEs.

ABSTRACT

Deployment is the process through which the software system is transferred to the client company. A risk is the probability that a loss will occur and in a software project, it could occur through a decrease in the quality of the software product, an increase in costs, a delay in completion or a failure, among other losses.

Currently, a prevailing condition for the growth of the software industry is that companies offer higher quality products that satisfy the demands and requirements of the customer, but above all that generates confidence at the time of use. This is achieved through the application of internationally recognized risk management models and methodologies. However, in Argentina, the software industry is mainly made up of small and medium-sized companies (SMEs), which represent almost 80% of the sector (this constitutes them as a fundamental link in the country's economy), but in the same it becomes very difficult to implement this type of models and methodologies because it involves a large investment in money, time and resources.

In this context and understanding the need to carry out initiatives that contribute to the development and improve the competitiveness of these companies, the present thesis work aims in the first place, to contribute to Software Engineers involved in software development projects in SMEs, a set of risks to consider in the software systems deployment process that cover three aspects, the process itself, the product to be implemented and the peopleware that participates in the process, as well as the procedures for their prevention, mitigation and / or transfer which are also adapted to the characteristics of SMEs. Second, and not because it is less important than the first, to strengthen the software system deployment process, which, as it is considered the last link in the software production chain, is often not so addressed.

Finally, two case studies are developed applying the set of proposed risks, as well as the procedures for their prevention, mitigation and / or transfer in two software development SMEs in Argentina. This made it possible to confirm the viability of the proposal, as well as to strengthen the process of deployment of software systems.

Keywords: risk management, deployment of software systems, SMEs.

DEDICATORIAS

A mi mamá Juana y mis hermanos Walter y Teresa,

que estarán siempre en mi corazón.

AGRADECIMIENTOS

A la Escuela de Posgrado de la Universidad Tecnológica Nacional - Facultad Regional Buenos Aires.

A mi directora, la Mg. Marisa Daniela Panizzi, por permitirme ser parte de su grupo de investigación y guiarme de forma cotidiana con sus consejos y observaciones.

A mi director, el Mg. Rodolfo Alfredo Bertone, por sus aportes constantes para poder llevar adelante esta Tesis.

A la directora de la Maestría en Ingeniería en sistemas de información, Dra. Florencia Pollo Cattaneo por su continuo acompañamiento y apoyo.

A mis hijas Paloma y Malena, por el amor con el que me brindaron todas las horas necesarias para poder llevar adelante esta tesis.

A mis tíos Edil y José Luis, por ser mi guía y haberme brindado la oportunidad de estudiar en una etapa muy difícil de la vida.

A mis hermanos Miriam, Cristina, Gustavo, Mirta y Jorge que siempre estuvieron y están a mi lado, en las buenas y en las malas.

A mis mejores amigos Luis, Néstor, Juan y Gustavo quienes siempre me apoyaron para la consecución de este logro académico.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

NOMENCLATURA

SEI	Software Engineering Institute
PMI	Project Management Institute
CMMI	Capability Maturity Model Integration
IS / IT	Information Systems/Information Technologies
SMS	Systematic Mapping Studies
ACM	Association for Computing Machinery
PMBOK	Project Management Body of Knowledge
SRE	Software Risk Evaluation
ISO	International Organization for Standardization
MAGERIT	Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información
PyME	Pequeña y mediana empresa
DoD	Departamento de Defensa

1. INTRODUCCIÓN

Diversos factores pueden afectar a los proyectos de software como, por ejemplo, las modificaciones en las prioridades, una planificación inapropiada, los cambios en los objetivos del proyecto, los requerimientos inexactos, la falta de recursos, etc., según Charette en (Charette R., 2005), uno de los factores más importantes son los riesgos no gestionados.

Gran cantidad de proyectos carecen de enfoques formales de gestión de riesgos. La identificación de estos a menudo depende informalmente de las habilidades y el nivel de experiencia de los administradores de software (Jones C., 1994).

En Argentina el sector SSI (Software y Servicios Informáticos), se caracteriza por la prestación de servicios intangibles, haciendo uso intensivo del conocimiento y la innovación, principales fuentes de generación de ventajas competitivas (CESSI, 2008). Este sector presenta un alto potencial para generar valor agregado al ecosistema productivo del país, promoviendo la generación de empleo calificado y evidenciando un crecimiento exponencial en los últimos años. En el mismo, se observa un claro predominio de las micro, pequeñas y medianas empresas (PyMEs). Así, y según el reporte anual del año 2019 publicado por el Observatorio Permanente de la Industria del Software y Servicios Informáticos (OPSSI) (CESSI, 2020), en Argentina la Industria del Software se compone mayoritariamente por PyMEs, representando casi el 80% del sector, lo que constituye un eslabón fundamental para el país y refuerza la necesidad de llevar adelante iniciativas que contribuyan con el desarrollo y mejora de competitividad de dichas empresas.

La gestión de riesgos del software es una parte crucial de la gestión exitosa de proyectos, pero a menudo no se implementa de manera adecuada en proyectos de software del mundo real gestionados por empresas PyMEs. Una razón es que los gerentes de proyecto carecen de herramientas efectivas o su aplicabilidad se limita a algunos escenarios especiales. (Liu, D. et al., 2009).

El objetivo de este trabajo de tesis consiste en fortalecer el proceso de despliegue de sistemas de software en pequeñas y medianas empresa (PyMEs) mediante la gestión integral de un conjunto de riesgos (prevenir, mitigar y/o transferir). De este modo, se busca incrementar la calidad y capacidad de sus procesos y, en consecuencia, aumentar la competitividad de dichas empresas.

1.1. DESPLIEGUE DE SISTEMAS DE SOFTWARE

El despliegue de sistemas de software es la fase del ciclo de vida de desarrollo en la que se transfiere el producto software al cliente. Este proceso es abordado por algunas metodologías y/o estándares internacionalmente reconocidos, aunque con alcances diferentes. En el caso del estándar ISO/IEC/IEEE 12207 (ISO/IEC/IEEE 12207, 2017) se la denomina "Proceso Transición" e involucra las actividades de Preparación del despliegue, Realización del despliegue y Gestión de los resultados del despliegue y además cada una de estas actividades se descomponen en un conjunto de tareas. Este estándar delega las decisiones de su aplicación a las organizaciones.

En la metodología Métrica v3 (PAE, 2001) al proceso equivalente se lo denomina "Fase de implantación y aceptación del sistema", e incluye las actividades:

- IAS1 Establecimiento del plan de implantación,
- IAS2 Formación necesaria para la implantación,
- IAS3 Incorporación del sistema al entorno de operación,
- IAS4 Carga de datos al entorno de operación,
- IAS5 Prueba de implantación del sistema,
- IAS6 Pruebas de aceptación del sistema,
- IAS7 Preparación del mantenimiento del sistema,
- IAS8 Establecimiento del acuerdo de nivel de servicio,
- IAS9 Presentación y aprobación del sistema y
- IAS10 Paso a producción.

Por otro lado, en el Método de Desarrollo de Sistemas Dinámicos (en inglés, Dynamic Systems Development Method o DSDM) (Agile Business Consortium, 2016) se la denomina "Despliegue" e involucra a las actividades: armado, revisión y despliegue y cierre del proyecto.

En el Proceso Unificado de Rational (en inglés, Rational Unified Proccess o RUP) (IBM, 2007) cuyas actividades son: identificar las estrategias de compatibilidad, conversión y migración, determinar el programa de despliegue, determinar la secuencia de despliegue y determinar las necesidades de formación de los usuarios también se la denomina "Despliegue" y esto mismo ocurre con el Proceso Unificado Ágil (en inglés, Agile Unified Proccess o AUP) (Ambler S., 2016).

Además de las metodologías y/o estándares que contemplan al proceso de despliegue, en el contexto de las metodologías ágiles existen prácticas que permiten transferir el producto software desde el desarrollo a operación, tales como DevOps (acrónimo en inglés de Desarrollo y Operaciones),

movimiento cultural que tiene como objetivo la colaboración de todas las partes interesadas en el desarrollo, implementación y operación de software para entregar un producto o servicio de calidad en el menor tiempo posible (Erich F., 2017) o el Despliegue Continuo (en inglés, Continuous Deployment), estrategia en la que cualquier versión de código que pasa la fase de prueba se libera automáticamente en el entorno de producción y permite liberar pequeñas funcionalidades sin afectar la operatoria del usuario final (Agile Alliance, 2020). Estas soluciones emergentes son solo utilizadas por compañías multinacionales e innovadoras como Google, Amazon, Netflix, LinkedIn, Facebook o Spotify (Diaz J. et al., 2019), que cuentan con recursos económicos y tecnológicos para poder aplicarlas. Por este motivo, esta propuesta de definición de riesgos está orientada a las PyMEs de Argentina que requieren estabilizar sus procesos de software con el propósito de ser competitivas a pesar de contar con recursos tecnológicos y recursos humanos insuficientes (Ianzen, A. et Al, 2013).

El proceso de despliegue contiene prácticas que tienden a presentar problemas, como falta de componentes (externos), descargas incompletas y despliegues erróneos (Jansen, S., 2006). Los problemas que pueden ocurrir en la fase de despliegue se transfieren y eventualmente resuelven como parte de la fase de mantenimiento. Algunas empresas suelen tardar meses y hasta años, en lograr finalizar el despliegue de un sistema de software en su totalidad. Es por esto que un despliegue eficiente de software ahorrará considerablemente recursos en términos de costo y esfuerzo (Subramanian, N., 2017).

A menudo el despliegue de software se realiza en entornos distribuidos y heterogéneos que suman complejidad generando pérdida de tiempo y costos adicionales (Tyndall J., 2012). El despliegue implica una serie de cambios en varios niveles, como procesos, formas de trabajo, tecnología y estructura organizativa. Esto implica una serie de desafíos que deben considerarse como la complejidad de la estructura organizacional existente, el cambio en la forma de trabajar de las personas, falta de experiencia y habilidades, infraestructura de TI insuficiente, falta de soporte técnico y presupuestos inadecuados (Reascos I. et al, 2019).

De acuerdo con Irving Reascos Paredes y João Alvaro Carvalho (Paredes I. et al., 2017), se consideran tres contextos para reconocer las causas de la alta tasa de fallas en los proyectos de despliegue de aplicaciones de software en PyMEs: tecnológico, organizacional y medioambiental. Entre las principales causas tecnológicas se encuentran: la infraestructura heterogénea e incompatible, las pocas capacidades y competencias tecnológicas de las PyMEs, la complejidad de estos sistemas, su ajuste y personalización en la empresa y la mala calidad y seguridad de los datos. Las causas organizativas que se mencionan son: liderazgo deficiente, planificación estratégica baja, costos directos e indirectos mal estimados, errores en las etapas iniciales que luego escalan a la siguiente,

deficiencias en la estructura de la organización y procesos informales, falta de recursos necesarios, bajos niveles de gestión de riesgos, descuido de aspectos sociales como la resistencia del usuario, canales de comunicación informal, capacitación y preparación inadecuadas de los usuarios finales. Finalmente, las causas del medio ambiente son: cambios en las regulaciones gubernamentales y constantes presiones del mercado.

Otros autores plantean que en el proceso de despliegue de sistemas de software se presentan errores o prácticas inadecuadas. El proceso de despliegue comprende prácticas que permiten que el sistema de software se instale, se desinstale o bien se actualice en la empresa-cliente. Estas prácticas vinculadas a la gestión de la configuración del cliente, en algunas ocasiones son afectadas por algunos inconvenientes, como por ejemplo la falta de componentes, instalaciones incompletas o errores de instalación (Jansen et al., 2006). Forbes et al. (Forbes et al., 2003) plantean que el resultado de prácticas de despliegue no estandarizadas e inadecuadas se refleja en los sistemas de información, los cuales son difíciles de mantener y operar.

Por esta razón, el despliegue de una aplicación no es un problema menor, tiene sus dificultades y exige competencias específicas para ejecutarse con éxito, se trata de llevar el cambio a un entorno estable, redefinir el trabajo, las estructuras sociales y alterar el equilibrio de potencia existente (Reascos I. et al, 2019).

1.2. PLANTEAMIENTO DEL PROBLEMA

La gestión del riesgo es particularmente importante para los proyectos de software, debido a la incertidumbre inherente que enfrentan la mayoría de los proyectos de este tipo. Ésta se deriva de requerimientos vagamente definidos, cambios de requerimientos que obedecen a cambios en las necesidades del cliente, dificultades en estimar el tiempo y los recursos requeridos para el desarrollo de software, o bien, se deriva de diferencias en las habilidades individuales. Es necesario anticipar los riesgos; comprender el efecto de estos sobre el proyecto, el producto y la empresa; y dar los pasos adecuados para evitar dichos riesgos (Sommerville I, 2016).

Las encuestas de la industria del software sugieren que solo alrededor de una cuarta parte de los proyectos de software tienen éxito total (es decir, se completan según lo programado, presupuestado y especificado), y anualmente se pierden miles de millones de dólares por fallas o proyectos que no ofrecen los beneficios prometidos (Johnson D., 2009). La naturaleza de los proyectos de IS / IT crea muchos riesgos que deben administrarse con diligencia. (Kwak Y., 2004).

De acuerdo con el informe de CHAOS del Standish Group (The Standish Group, 2018) la tasa de éxito de los proyectos de Ingeniería de Software se mantiene en un rango que oscila del 27% al 31% en los últimos años, mientras que los proyectos modificados (proyectos que se retrasan, superan el presupuesto y tienen menos características y funciones que las especificadas originalmente) o cancelados ascienden por encima del 70%., se presenta en la Tabla 1.1, En la industria del software, la cultura de la gestión de riesgos es: "reconocimiento de riesgo = derrotismo".

	2011	2012	2013	2014	2015	2016	2017
Exitosos	29%	27%	31%	28%	29%	26%	29%
Modificados	49%	56%	50%	55%	52%	53%	53%
Cancelados	22%	17%	19%	17%	19%	21%	18%

Tabla 1.1 Porcentaje de cumplimiento de proyectos de Software (Chaos Report Standish Group, 2018).

A pesar del reconocimiento de la importancia de la gestión de riesgos y sus implicancias en la industria, en la gestión de riesgos en la Ingeniería de Software todavía no presenta avances significativos. Una de las razones de este escenario, es que el riesgo es subjetivo en los proyectos de software. En este sentido, una forma de reducir el sesgo de subjetividad es mediante el uso de métricas, ya que podría ser útil proporcionar a los interesados un mejor conocimiento, control y mejora de los procesos de gestión de riesgos (Menezes J. et al., 2013).

La falta de gestión de riesgos, comunicación y la comprensión de los requisitos son en realidad los principales factores relacionados con la baja tasa de éxito en el desarrollo de software (Caballero, S. et al., 2018).

A pesar de la existencia de metodologías, estándares y prácticas que custodian el proceso de despliegue, continúan ocurriendo una serie de dificultades en el proceso que afectan la calidad de su ejecución. Dado que las PyMEs para ser competitivas en el sector del software no solo a nivel nacional, sino que también internacional, requieren de procesos estabilizados, en este trabajo de tesis se busca fortalecer el proceso de despliegue de sistemas de software mediante la definición de un conjunto de riesgos junto con los procedimientos asociados que permitan prevenir, mitigar y/o transferir los inconvenientes que se presentan en el proceso.

1.3. OBJETIVOS DE LA TESIS

1.3.1 GENERAL

El objetivo general de la presente tesis consiste en la:

> Definición de un conjunto de riesgos para el proceso de despliegue de sistemas de software junto con los procedimientos que permitan prevenir, mitigar y/o transferir los mismos para PyMEs de la República Argentina.

1.3.2 ESPECÍFICOS

Los objetivos específicos de la presente tesis se descomponen en los objetivos que se detallan a continuación:

- OB1. Construcción del estado del arte sobre metodologías, métodos y estándares que abordan la gestión de riesgos en proyectos de software con foco en el proceso de despliegue.
- OB2. Análisis y diseño de un conjunto de riesgos para el proceso de despliegue, así como también los procedimientos que permitan prevenir, mitigar y/o transferir los mismos.
- OB3. Validar el conjunto de riesgos propuestos mediante la realización de estudios de casos.

1.4. METODOLOGÍA DE INVESTIGACIÓN

Para el desarrollo de la tesis, se siguió un enfoque de investigación clásico (Riveros H. et al., 1985) con énfasis en la producción de tecnologías (Sábato J. et al., 1982); identificando los métodos y materiales necesarios.

1.4.1. MÉTODOS

A continuación, se definen los métodos de investigación que se utilizarán en este trabajo de tesis.

1.4.1.1. Mapeo Sistemático de la Literatura (en inglés, Systematic Literature Mapping o SMS)

Los mapeos sistemáticos de la literatura tienen como principal objetivo proporcionar una visión global sobre un tema de interés (con enfoque empírico o no) e identificar la cantidad y tipo de investigación y resultados disponibles sobre el mismo. Para la realización del SMS se considerarán los lineamientos propuestos por Genero (Genero M. et al., 2014) de acuerdo al proceso propuesto por Kitchenham (Kitchenham B., 2007).

1.4.1.2. MÉTODO DESMET

DESMET es una metodología que tiene como objetivo evaluar métodos y herramientas de Ingeniería de Software de forma cualitativa en base al análisis de características (Kitchenham B. et al., 1996). El método DESMET está destinado a ayudar a un evaluador a planificar y ejecutar un ejercicio de evaluación que es imparcial y confiable. DESMET utiliza el término análisis de características para describir una evaluación cualitativa. El análisis de características se basa en identificar los requisitos que tienen los usuarios para una tarea / actividad en particular y mapear esos requisitos para características que debe poseer un método / herramienta destinado a respaldar esa tarea / actividad.

1.4.1.3. PROTOTIPADO EVOLUTIVO EXPERIMENTAL

El prototipado evolutivo experimental (Basili V., 1993) consiste en desarrollar una solución inicial para un determinado problema, generando su refinamiento de manera evolutiva por prueba de aplicación de dicha solución a estudio de casos (problemáticas) de complejidad creciente. El proceso de refinamiento concluye al estabilizarse el prototipo en evolución.

1.4.1.4. ESTUDIO DE CASO

El estudio de caso en Ingeniería de Software es un método de investigación empírica que hace uso de múltiples fuentes de evidencia para investigar un fenómeno dentro de su contexto real y permiten comprenderlo más en profundidad (Genero M. et al., 2014)

Los estudios de caso se caracterizan por (Runeson P. et al., 2012):

- Ser un método de investigación flexible, ya que han de tratar con las complejas y dinámicas características de los fenómenos del mundo real.
- Sus conclusiones, se basan en una cadena de evidencia, recogida de múltiples fuentes de una forma planeada y consistente.
- Añaden conocimiento al ya existente, basándose en una teoría previamente establecida o estableciendo una si no la hubiera con anterioridad.

1.4.2. ABORDAJE METODOLÓGICO

Los métodos de investigación y desarrollo mencionados en la sección 1.2.1, se han aplicado de la siguiente manera:

- Para el OB1 se desarrolló el SMS y se realizó el análisis comparativo de las metodologías y estándares que contemplan al proceso de despliegue mediante el método DESMET.

- Para el OB2 se definió un conjunto de riesgos para el proceso de despliegue, así como también los procedimientos de mitigación mediante el prototipado evolutivo experimental y
- Finalmente, para el OB3 se validó el conjunto de riesgos para el proceso de despliegue de sistemas de software mediante un par de estudios de casos desarrollados en PyMEs de desarrollo de software de Argentina.

1.5. CONTEXTO DE LA INVESTIGACIÓN

La investigación de la presente tesis se encuentra vinculada al Proyecto de Investigación titulado "ESTUDIO DEL PROCESO DE IMPLANTACIÓN DE SISTEMAS INFORMÁTICOS EN EL CONTEXTO INDUSTRIAL DE LA REPÚBLICA ARGENTINA" (SIUTNBA0006576) y un proyecto predecesor titulado "EL IMPACTO DEL FACTOR PEOPLEWARE EN LA IMPLANTACIÓN DE SISTEMAS INFORMATICOS" (EIUTNBA0004347). Ambos proyectos, a cargo de la Mg. Marisa Daniela Panizzi y con el asesoramiento científico y tecnológico del Mg. Rodolfo Alfredo Bertone del Grupo de Ingeniería de Software el Instituto de Investigación en Informática de la Universidad Nacional de La Plata (III-LIDI).

1.6. ESTRUCTURA DE LA TESIS

En el capítulo 1, se brinda una introducción al tema que se aborda en la tesis de Maestría, el problema que se propone resolver, los objetivos a lograr, los métodos de investigación y desarrollo a emplear, así como también el abordaje metodológico y por último se presenta la estructura de organización de la tesis.

En el capítulo 2, se desarrolla la construcción del estado del arte. Se presenta el desarrollo del mapeo sistemático de literatura con el propósito de determinar las metodologías, métodos y estándares que abordan la gestión de riesgos en proyectos de desarrollo de software, son las más estudiadas por la comunidad científica y las más utilizadas en la industria del software. Luego se realiza un análisis comparativo utilizando el método DESMET de evaluación cualitativa basada en el análisis de características y se concluye la comparación con resultados preliminares. Por último, se presentan las conclusiones respecto al estado del arte de los riesgos para el proceso de despliegue de sistemas de software.

El capítulo 3, presenta el desarrollo de la construcción del conjunto de riesgos para el proceso de despliegue de sistemas de software para PyMEs de Argentina, así como también los procedimientos para su prevención, mitigación y/o transferencia.

Por último, en el capítulo 4 se desarrollan dos estudios de caso con el propósito de examinar la viabilidad de la aplicación del conjunto de riesgos propuestos, así como también de los procedimientos para su prevención en el contexto real. El primer estudio de caso se trata del despliegue de funcionalidades de un Portal de Recursos Humanos de una entidad bancaria realizado por una PyME de sistemas de la República Argentina. Y el segundo, se trata del despliegue de funcionalidades de un sistema de gestión de un laboratorio farmacéutico internacional con sede en Argentina, también realizado por una PyME de desarrollo de software.

2. ESTADO DEL ARTE

Para la construcción del estado del arte del presente trabajo de tesis, se realizó un mapeo sistemático de la literatura (en inglés Systematic Mapping Studies o SMS). El objetivo del SMS consiste en cotejar y sistematizar la evidencia empírica para identificar (determinar) cuáles de las metodologías, métodos y estándares que abordan la gestión de riesgos en proyectos de desarrollo de software, son las más estudiadas por la comunidad científica y las más utilizadas en la industria del software (Sección 2.1). Una vez identificadas las metodologías, métodos y estándares que abordan la gestión de riesgo, se decide cuales se consideran para el análisis comparativo con el propósito de identificar si estas custodian el proceso de despliegue de sistemas software (Sección 2.2). Por último, se presentan las conclusiones sobre el estado del arte. (Sección 2.3).

2.1 DESARROLLO DEL MAPEO SISTEMÁTICO DE LITERATURA

El mapeo sistemático se desarrolla de acuerdo a las pautas propuestas por Genero (Genero M. et al, 2014), respetando el proceso propuesto por Kitchenham (Kitchenham B. et al, 2004), el cual se encuentra conformado por tres etapas, las cuales se presentan en la Figura 2.1:

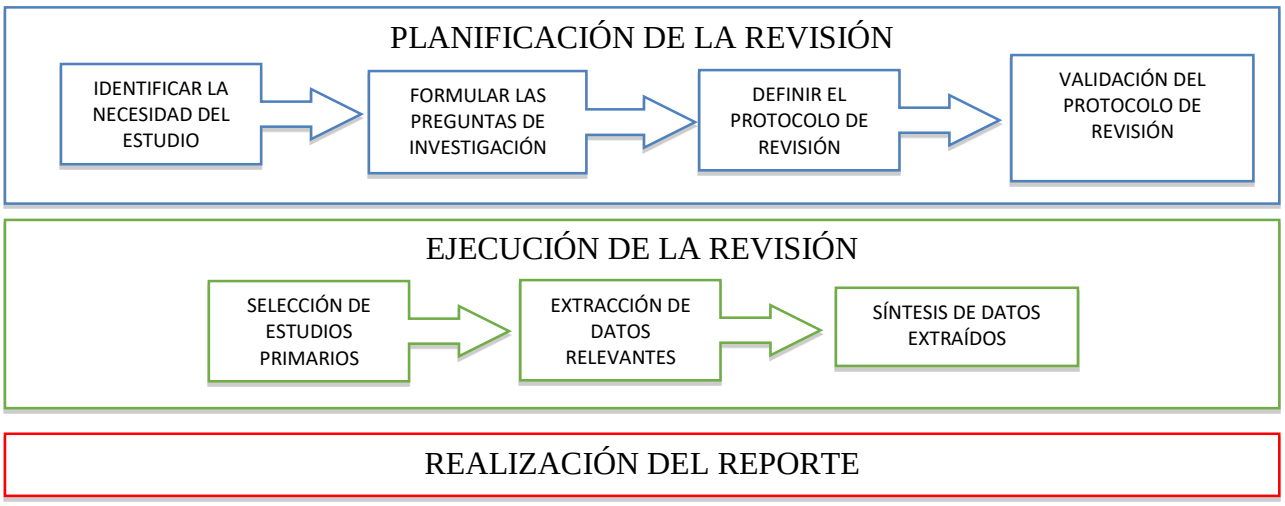


Figura 2.1. Diagrama de las etapas del SMS.

El desarrollo del SMS se encuentra organizado de la siguiente manera: la sección 2.1.1 describe la actividad de planificación la sección 2.1.2 detalla la ejecución de la revisión y finalmente en la sección 2.1.3 se presenta el reporte de los resultados obtenidos.

2.1.1. PLANIFICACION DEL SMS.

La etapa de planificación tiene como propósito definir el ámbito, la viabilidad y los pasos necesarios para realizar un mapeo sistemático de literatura en el marco de la gestión de riesgos de proyectos de software.

Esta actividad se divide en las siguientes tareas: identificar la necesidad de la revisión (sección 2.1.1.1), formular las preguntas de investigación (sección 2.1.1.2) y definir el protocolo de revisión (sección 2.1.1.3) que incluye la estrategia de búsqueda y los criterios de selección de estudios y por último la tarea, validación del protocolo (sección 2.1.1.4.).

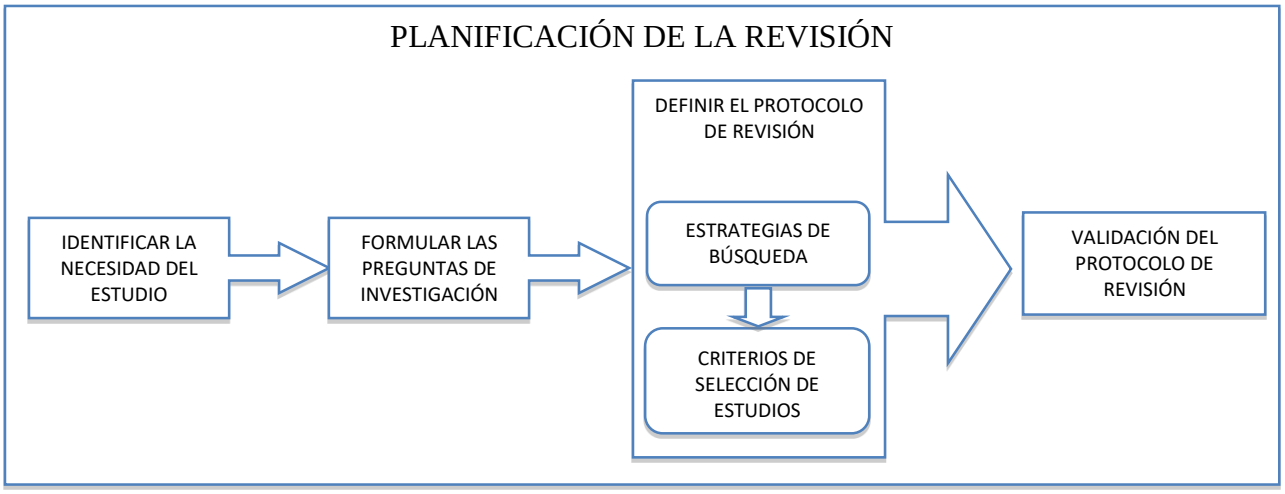


Figura 2.2. Tareas de la Actividad ¨ Planificación¨.

2.1.1.1. IDENTIFICAR LA NECESIDAD DEL ESTUDIO

Con el propósito de poder determinar de qué manera se gestionan los riesgos en el proceso de despliegue de sistemas software, se fija como objetivo de este SMS cotejar y sistematizar la evidencia empírica para identificar cuáles de las metodologías, métodos y estándares que abordan la gestión de riesgos en proyectos de desarrollo de software son las más estudiadas por la comunidad científica y las más utilizadas en la industria del software.

2.1.1.2. FORMULAR LAS PREGUNTAS DE INVESTIGACIÓN

El SMS realizado ayuda a responder la pregunta de investigación principal (PI):

¿Cuál es el estado de arte respecto a las metodologías, métodos y estándares que abordan la gestión de riesgos en proyectos de software?

La pregunta de investigación se descompone en tres sub-preguntas (PI1-PI3) y junto con su motivación (MO) se presentan en la Tabla 2.1.

Referencia	Pregunta de investigación (PI)	Motivación (MO)
PI1	¿Cuáles son las metodologías, métodos o estándares más utilizados que abordan la gestión de riesgos en proyectos de software?	Conocer cuáles son las metodologías, métodos o estándares más utilizados que aborden de forma nula, parcial o total la gestión de riesgos en los proyectos de software.
PI2	**¿**Están contemplados todos los procesos principales de la construcción de un producto software por las metodologías, métodos y estándares que abordan la gestión de riesgos en proyectos de software?	Determinar de qué manera las metodologías, métodos y estándares que abordan la gestión de riesgos custodian las diferentes etapas del desarrollo de software.
PI3	¿Qué tipo de contribución se ha realizado a nivel académico acerca de gestión de riesgos en proyectos de software?	Identificar los aportes realizados por la comunidad científica respecto a la gestión de riesgos de proyectos de software.
PI4	¿Cuáles son los tipos de investigación de los artículos?	Identificar los tipos de investigación de los artículos de acuerdo con la clasificación de Wieringa et al. propuesta en (Wieringa R. et al., 2005)

Tabla 2.1. Preguntas de investigación y motivación.

2.1.1.3. DEFINICIÓN PROTOCOLO DE REVISIÓN

En esta tarea se describen cada uno de los elementos del protocolo de revisión definido para este SMS, la estrategia de búsqueda (Sección 2.1.1.3.1) y los criterios de selección de estudios (Sección 2.1.1.3.2).

2.1.1.3.1 ESTRATEGIAS DE BÚSQUEDA

Se decide la búsqueda automática en las bibliotecas digitales y repositorios que se presentan en la Tabla 2.2. Dado que esta investigación se desarrolla en la República Argentina, se decidió considerar la producción científica del Congreso Argentino de Ciencias de la Computación (CACIC) y del Workshop de Investigadores en Ciencias de la Computación (WICC); ambas actividades científicas organizadas por la Red de Universidades Nacionales con carreras en Informática (RedUNCI)[1].

[1] RedUNCI: Red de Universidades Nacionales con carreras en Informática, sitio: http://redunci.info.unlp.edu.ar/

REF	BIBLIOTECAS Y REP. DIGITALES	OPCIONES
F1	IEEE Xplore	Publicaciones de congresos, revistas
F2	ScienceDirect	Publicaciones de congresos, revistas
F3	Biblioteca digital de ACM	Publicaciones de congresos, revistas
F4	Google Academic	Publicaciones de congresos, revistas
F5	SEDICI[2]	Libro de Actas: Congreso Argentino de Ciencias de la Computación (CACIC), Workshop de Investigadores en Ciencias de la Computación (WICC).

Tabla 2.2. Fuentes utilizadas (F).

Los términos candidatos (T) que se definen para la búsqueda se presentan en la Tabla 2.3.

Referencia	Término
T1	Software
T2	Management
T3	Mitigation
T4	Software Project
T5	Risk
T6	Software Engineering
T7	Risk Assessment
T8	Implementation

Tabla 2.3. Términos de búsqueda (T).

La selección realizada sobre los términos candidatos permite la creación de las cadenas de búsqueda (C) utilizadas en la búsqueda como se muestran en la Tabla 2.4.

Referencia	Cadena
C1	“Software Project” and “Risk”
C2	“Software” and “Risk ” and “Management”
C3	“Software Engineering” and “Risk Assessment”
C4	“Management” and “Risk” and “Implementation”
C5	“Software” and “Risk” and “Mitigation”

Tabla 2.4. Cadenas de búsqueda (C).

2.1.1.3.2 CRITERIOS DE SELECCIÓN DE ESTUDIOS

Se incluirán únicamente aquellos artículos que cumplan con los siguientes criterios:

- Estudios comprendidos en el período del 2008 al 2018.
- Estudios en el idioma inglés y español.
- Estudios que contengan términos candidatos en el título y/o en el resumen.

[2] SEDICI: Repositorio Institucional de la Universidad Nacional de La Plata, http://sedici.unlp.edu.ar/

- Estudios duplicados. En el caso de encontrarse estudios duplicados del mismo o de los mismos autores se considera el más completo.

Los criterios de exclusión definidos en esta revisión son los siguientes:

- Estudios a los que no se tengan acceso.
- Estudios que cuenten solamente con el resumen.
- Estudios que no cumplan los criterios de inclusión.
- Literatura gris, tesis de grado y posgrado y libros.

2.1.1.4. VALIDACIÓN DE PROTOCOLOS DE REVISIÓN

El protocolo de revisión para la construcción del SMS, ha sido validado con los directores de la tesis para asegurar que se han tenido en cuenta todos los aspectos relevantes para dar respuesta a cada una de las preguntas de investigación planteadas en la sección 2.1.1.2.

2.1.2. EJECUCIÓN DE LA REVISIÓN

La actividad ¨Ejecución de la revisión¨ se divide en las siguientes tareas: selección de los estudios primarios (sección 2.1.2.1), extracción de los datos relevantes (sección 2.1.2.2) y, por último, síntesis de los datos extraídos (sección 2.1.2.3). Las tareas que componen la actividad ¨Realización de la revisión¨ se presentan en la Figura 2.3.

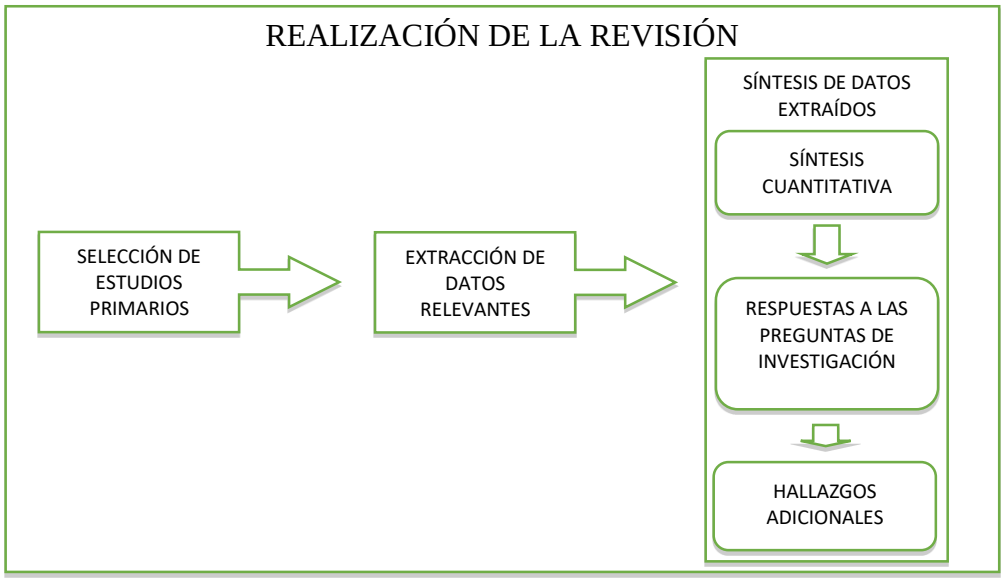


Figura 2.3. Tareas de la Actividad ¨Realización de la revisión¨.

2.1.2.1 SELECCIÓN DE ESTUDIOS PRIMARIOS

Al aplicar las cadenas de búsquedas definidas en la sección 2.1.1.3.1 en las bibliotecas y repositorios digitales arrojó un total de 132 artículos. Dada las características de las bibliotecas y repositorios ha sido necesario ajustar las cadenas por las limitaciones que estos presentaban, el resultado de la búsqueda se presenta en la Tabla 2.5.

La búsqueda se realizó en el título y en el resumen excepto en determinados casos donde se debió buscar en el texto completo. De cada uno de los artículos se guardaron: las cadenas de búsqueda, los metadatos de los artículos encontrados (año, título, autores, etc.) y los resúmenes de los mismos. De los 132 artículos encontrados, se aplicaron los criterios de inclusión y exclusión, se eliminaron los artículos duplicados y resultaron 100 artículos que luego de su lectura completa han sido considerados estudios primarios.

La Tabla 2.5 muestra la cadena de búsqueda utilizada en cada biblioteca digital y su número correspondiente de registros recuperados.

Librería digital	Cadena de búsqueda	Artículos relevantes
ScienceDirect	(TITLE (Software Project OR deployment) AND TITLE (Risk AND Management OR Risk AND Engineering)) AND PUBYEAR > 2008 AND PUBYEAR < 2018	18
IEEE Xplore	((((“Title”:“Software Project” OR “deployment”) AND (“Title”:“Risk” AND “” OR “Software” AND “Engineering” OR “Risk” AND “Implementation” OR “Risk” AND “Mitigation”)))) Filters Applied: Conferences Journals 2008 – 2018	10
ACM	acmdlTitle:((Software Project OR deployment) AND (Risk OR Management OR Risk AND Implementation)) Published since 2008 Content formats: pdf ACM publications: Proceeding ACM publications: Journal	4
Google Academics	((((“Title”:“Software Project” OR “deployment”) AND (“Title”:“Risk” AND “” OR “Software” AND “Engineering” OR “Risk” AND “Implementation” OR “Risk” AND “Mitigation”)))) Filters Applied: Conferences Journals 2008 – 2018	95
SEDICI	http://sedici.unlp.edu.ar/discover?query=risk+software&submit=&filtertype_0=keywords&filtertype_1=dateIssued&filtertype_2=type&filter_relational_operator_1=equals&filter_relational_operator_0=equals&filter_2=Articulo&filter_1=%5B2008+TO+2018%5D&filter_relational_operator_2=equals&filter_0=ingenier%C3%ADa+de+software	5
Total		132

Tabla 2.5. Resultados obtenidos luego de la búsqueda en librerías digitales.

En el caso del repositorio digital SEDICI, no se introduce una cadena de búsqueda, sino que se presenta en la Tabla 2.5. el localizador de recursos uniforme (url por sus siglas en inglés) resultante de la búsqueda en dicho repositorio.

En el Apéndice A, se encuentra el listado de los estudios primarios utilizados para el SMS.

2.1.2.2 EXTRACCIÓN DE LOS DATOS RELEVANTES

Con el propósito de clasificar los datos obtenidos se desarrolló una planilla, la cual permitió registrar los datos extraídos. La misma se encuentra conformada por 4 dimensiones, cada dimensión se corresponde a una pregunta de investigación (PI) y en la columna descripción se presentan los valores que puede tomar cada una de las dimensiones (Tabla 2.6):

Dimension	Descripción
Metadatos	Agrupa las categorías ID (identificador único del artículo), Cadena de búsqueda, Año de publicación, Título, Autor/es, Fuente, Fuente de búsqueda, Tipo de publicación (en Congreso o en Revista), País, Continente, Palabras clave, Cita APA, Cantidad de citas, Problema, Propuesta y Resultados.
PI1/Soluciones:	Agrupa las categorías de las diferentes soluciones existentes respecto a gestionar, mitigar riesgos. Es importante aclarar que la inclusión de una categoría nueva en la dimensión se ha realizado en función de su aparición en los estudios.
PI2/ Contexto:	Agrupa las categorías en los procesos principales de la construcción de un producto software, Requisitos y Análisis, Arquitectura y Diseño, Desarrollo, Pruebas, Despliegue y Mantenimiento. También se consideran los procesos de soporte y de gestión de los proyectos de software, entre ellos: Verificación y Validación, Riesgo, Gestión de configuración. La definición de esta categoría se basa en la ISO/IEC/IEEE-12207 (ISO/IEC/IEEE 12207:2017, 2017).
PI3/Contribución:	Agrupa las categorías en función del aporte realizado en la investigación del estudio. Estas son: Métricas, Herramienta, Modelo, Método, Proceso o Buenas prácticas.
PI4/Propósito:	Agrupa las categorías según el tipo de investigación realizada en el estudio. Siguiendo los lineamientos de (Wieringa R. et Al, 2005), se consideraron las siguientes categorías: evaluación, validación, propuesta de solución, experiencia personal y filosófica.

Tabla 2.6. Dimensiones a considerar en el SMS.

2.1.2.3 SÍNTESIS DE DATOS EXTRAÍDOS

En esta sección (2.1.2.3.1), se realiza un análisis de los estudios primarios obtenidos para dar respuesta a cada una de las preguntas de investigación y en la sección (sección 2.1.2.3.2) se presentan hallazgos adicionales mediante el uso de gráficos y tablas.

2.1.2.3.1 RESPUESTAS A LAS PREGUNTAS DE INVESTIGACIÓN

A continuación, se procede a responder cada una de las preguntas de investigación (PI):

PI1: ¿Cuáles son las metodologías, métodos o estándares más utilizados que abordan la gestión de riesgos en proyectos de software?

En el gráfico 2.1, se visualiza que la metodología más utilizada es CMMI (28 menciones) seguida por PMBOK (26 menciones). A continuación, se encuentra el método de evaluación de riesgos de software (Software Risk Evaluation Method) desarrollado por el SEI (con 23 menciones) y con un total de 11 menciones, se ubican las técnicas de Inteligencia Artificial (entre ellas Machine Learning, Redes Neuronales, etc.). Con menor representatividad, se presentan Risk Management Frameworks (6 menciones), Prince2 (5 menciones), AS/NZS 4360 (3 menciones), ISO 31000 (3 menciones), ISO 12207 (2 menciones), Risk IT (2 menciones) y Magerit (1 mención).

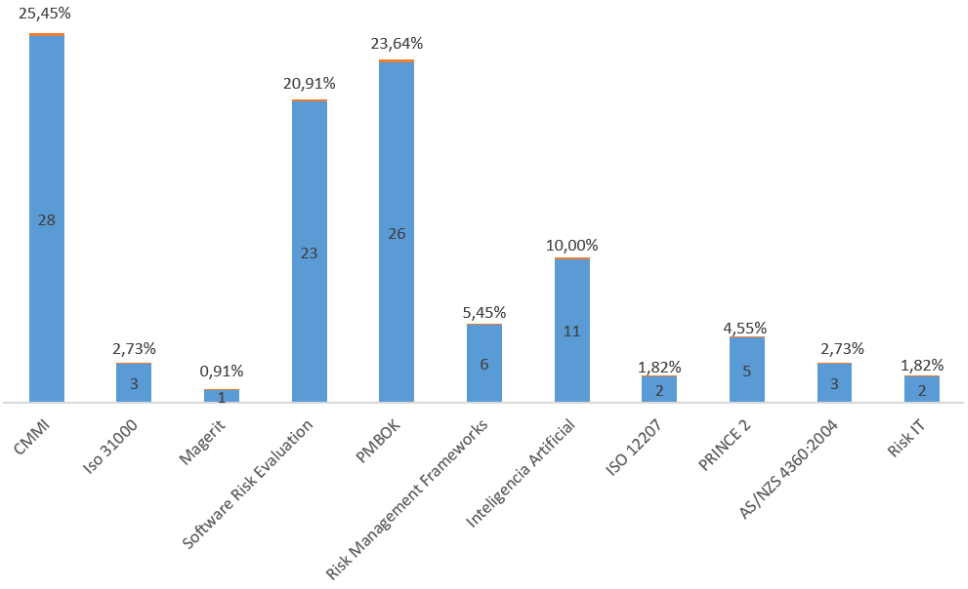


Gráfico 2.1. Cantidad de referencias a metodologías, métodos o estándares que abordan la gestión de riesgos.

PI2: ¿Están contemplados todos los procesos principales de la construcción de un producto software por las metodologías, métodos y estándares que abordan la gestión de riesgos en proyectos de software?

El Gráfico 2.2 muestra el porcentaje de artículos primarios que hacen referencia al proceso de gestión de riesgo (96 %) y al resto de los procesos de soporte y gestión (4 %).

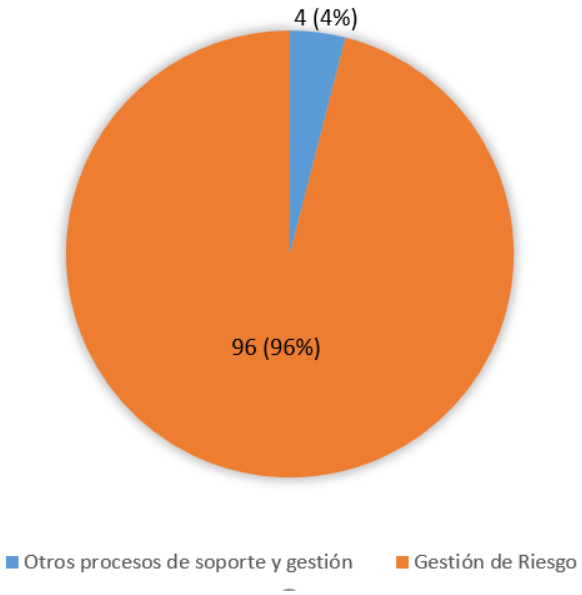


Gráfico 2.2. Porcentaje de artículos primarios por procesos de gestión y otros procesos de soporte y gestión. Colocar en el grafico la cantidad y el porcentaje.

En base al total de artículos primarios que hacen referencia al proceso de gestión de riesgo (96%), en el Gráfico 2.3 se muestra la distribución respecto de los procesos principales de la construcción de un producto software.

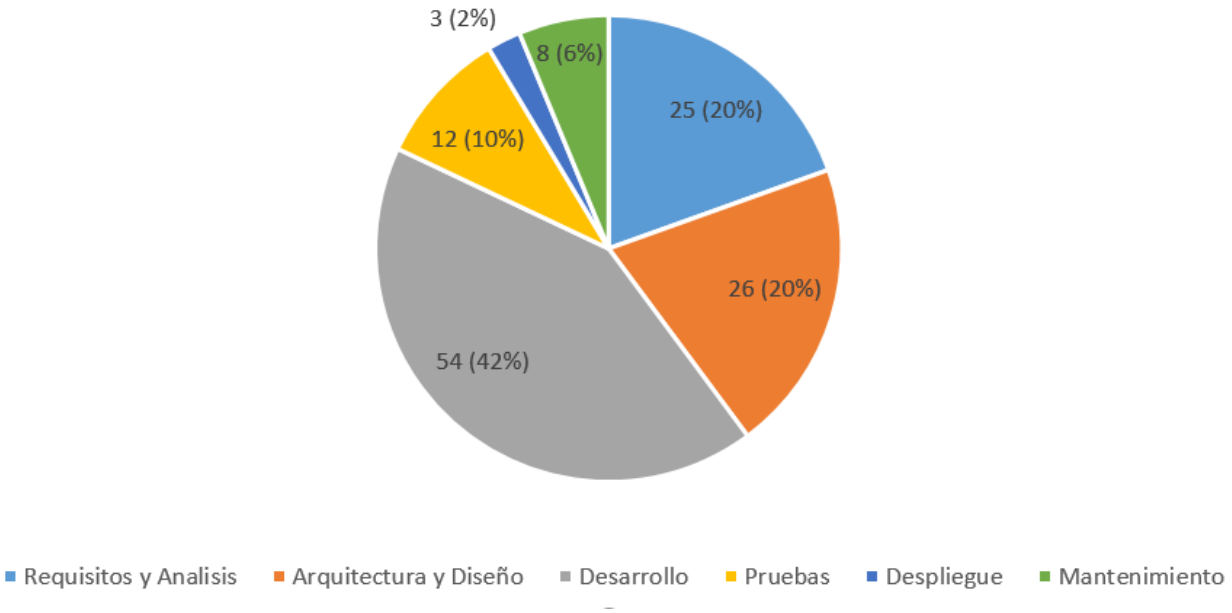


Gráfico 2.3. Porcentaje de estudios de gestión de riesgo por grupos de procesos principales.

PI3: ¿Qué tipo de contribución se ha realizado a nivel académico acerca de gestión de riesgos en proyectos de software?

En el Gráfico 2.4, se muestran los tipos de contribución de los artículos primarios.

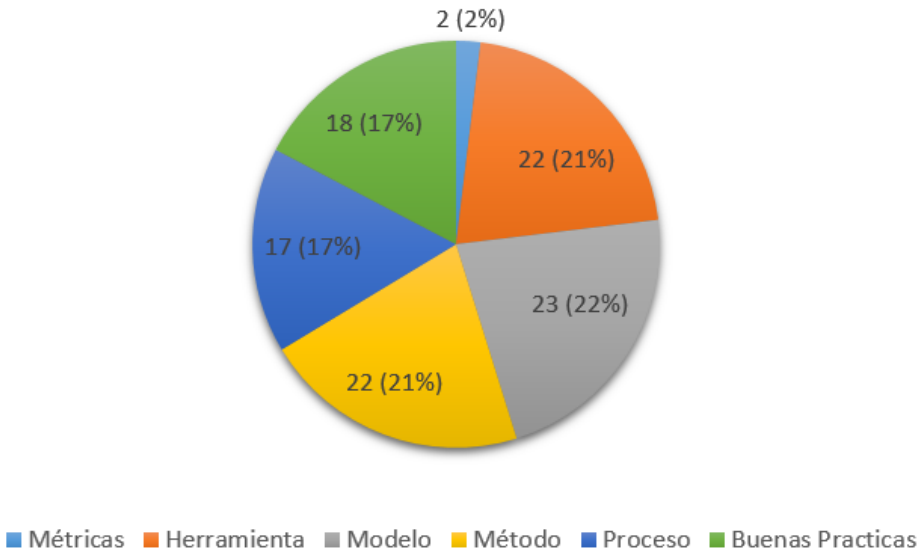


Gráfico 2.4. Porcentaje de artículos primarios según la dimensión contribución.

PI4: ¿Cuáles son los tipos de investigación de los artículos?

En el gráfico 2.5, se presenta el porcentaje de artículos por el tipo de investigación realizada en el estudio.

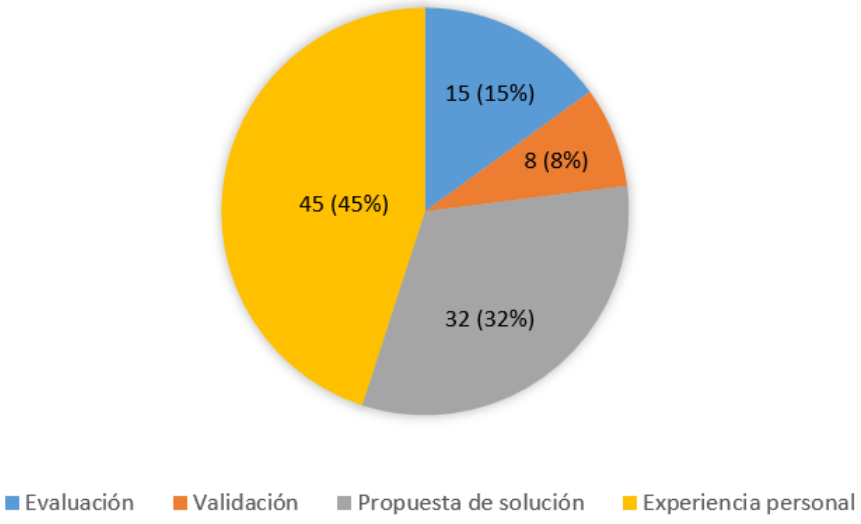


Gráfico 2.5. Porcentaje de artículos primarios por propósito de investigación.

2.1.2.3.2 HALLAZGOS ADICIONALES

En esta sección se muestran algunos hallazgos adicionales del SMS como por ejemplo la cantidad de estudios primarios por año de publicación, los continentes que han aportado más artículos y el tipo de publicación (congreso o revista).

En el gráfico 2.6, se presenta la cantidad de estudios primarios de acuerdo con el año de publicación. Se observa que los picos más altos de publicaciones se presentan en los años 2008 y 2009, con un total de 17% (17 estudios) cada uno.

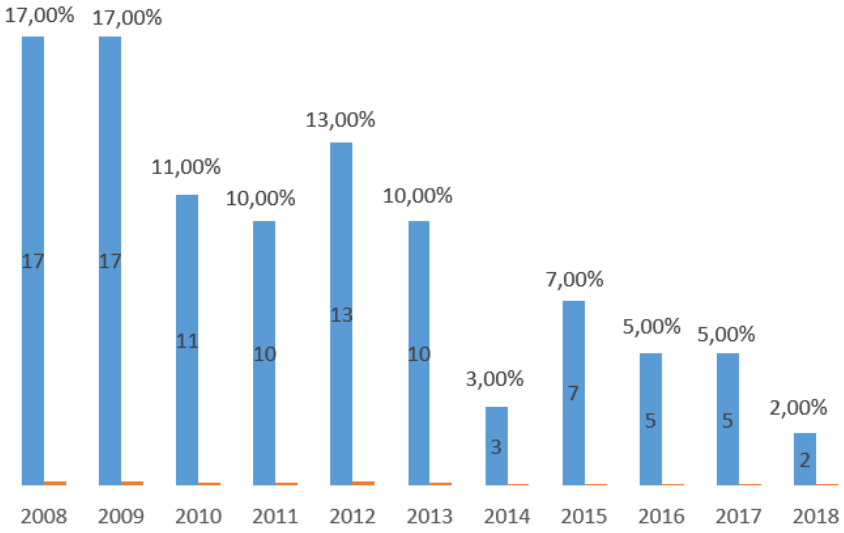


Gráfico 2.6. Cantidad de artículos primarios por año de publicación.

En el gráfico 2.7, se presenta la cantidad de estudios según la fuente de publicación. Cabe destacar que la mayoría de los trabajos encontrados fueron publicados en revistas, lo que representa el 58% (58 estudios) del total, y las publicaciones en congresos representan el 42% restante (42 estudios).

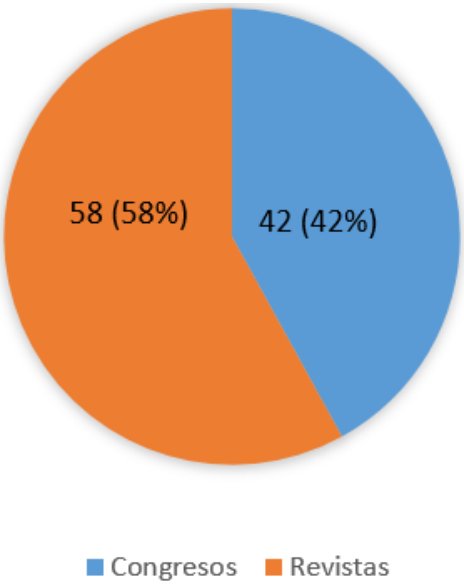


Gráfico 2.7. Distribución porcentual de artículos primarios por tipo de publicación.

En el gráfico 2.8, se presenta una distribución de los congresos respecto a los continentes en los que han sido realizados. El mayor porcentaje corresponde a Asia, con un 49 %, seguido por América del

Norte con un porcentaje del 21 %, Europa con un 16 %, América Latina y Del Caribe con un 12 % y por último Oceanía con un 2 %.

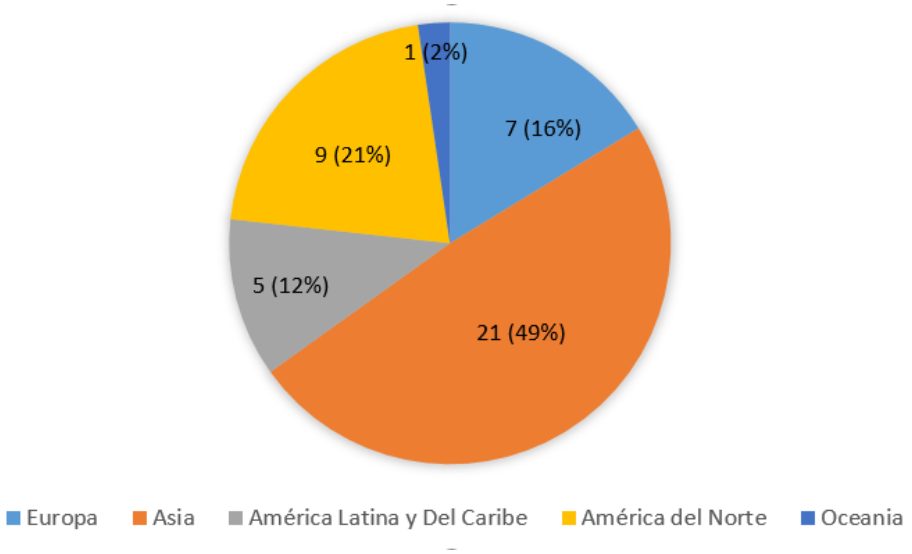


Gráfico 2.8. Distribución porcentual de los artículos primarios por continente de realización del congreso.

En la Tabla 2.7, se presenta una cantidad de artículos por cada una de las revistas.

Revista	Cantidad
Benchmarking: An International Journal	1
Centro de Informática. Universidad Federal de Pernambuco	1
Cloud Computing	1
Communications of the ACM	1
Decision Support Systems	1
First international workshop on Leadership and management in software architecture	1
Future Information Technology and Management Engineering	1
Global Software Engineering Workshops	1
HEC Montréal	1
IAENG international journal of Computer Science	1
IJCIT	1
IJMPB	1
Information management & computer security	1
Int. J. Risk Assessment and Management	1
Int. J.ProductionEconomics	1
International Journal of Accounting Information Systems	1
Cloud Computing	1
Communications of the ACM	1
Decision Support Systems	1
First international workshop on Leadership and management in software architecture	1
Future Information Technology and Management Engineering	1
Global Software Engineering Workshops	1
HEC Montréal	1

Revista	Cantidad
IJCIT	1
IJMPB	1
Information management & computer security	1
Int. J. Risk Assessment and Management	1
Int. J.ProductionEconomics	1
International Journal of Accounting Information Systems	1
International Journal of Information Technology & Decision Making	1
International Transactions in Operational Research	1
Journal of computing and information technology	1
Journal of Software Engineering and Applications	1
Journal of software: evolution and process	1
Journal of Theoretical & Applied Information Technology	1
MIS quarterly	1
Requirements Engineering	1
Risk Analysis: An International Journal	1
Software Engineering Advances	1
Universidad Politécnica de Hong Kong	1
Departamento de Ingeniería, Universidad de Pisa	2
IEEE Transactions on Software Engineering	2
Information & Management	2
Information and Software Technology	2
Information Sciences	2
International Journal of Computer Technology and Applications	2
International Journal of Information Management	2
Journal of Systems and Software	2
Procedia Computer Science	3
Revista Cubana de Ciencias Informáticas	3
International Journal of Project Management	8

Tabla 2.7. Cantidad de artículos primarios por revista.

En el gráfico 2.9, se presenta una cantidad de estudios respecto a su fuente de búsqueda. La mayor cantidad de artículos se corresponde a Google Academic, con un total de 80 estudios primarios. Luego, continúa ScienceDirect con un total de 9 artículos, seguido de IEEE Explore con 6 estudios primarios y por último SEDICI con un total de 3 artículos y la Biblioteca Digital de ACM con 2 artículos.

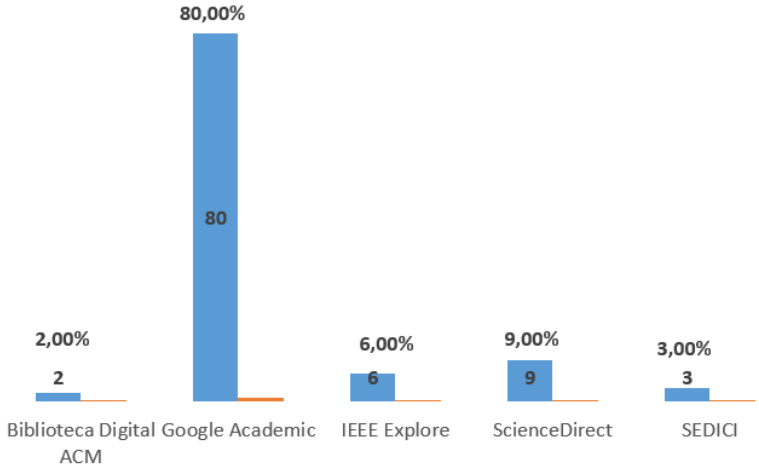


Gráfico 2.9. Cantidad de artículos primarios por fuente de búsqueda.

Finalmente, en el gráfico 2.10, se sintetizan la cantidad de artículos obtenidos por cada una de las cadenas de búsqueda utilizada. El mayor número de estudios se encontró utilizando la cadena "Risk Management Software" con un 51% del total (51 estudios).

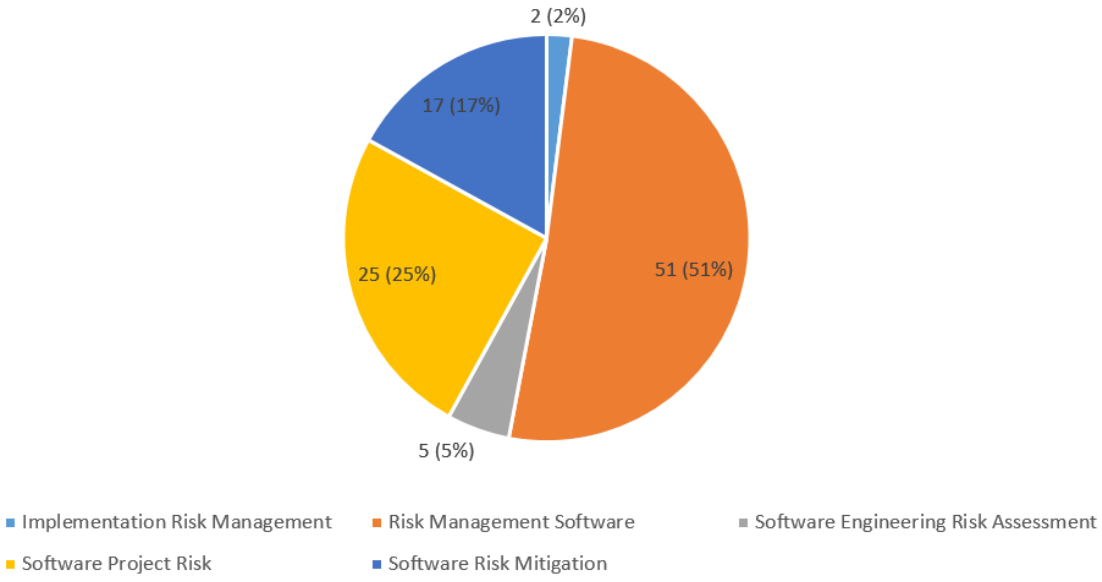


Gráfico 2.10. Cantidad de artículos primarios por cadena de búsqueda.

2.1.3. RESULTADOS DEL SMS.

En este capítulo se presentaron los resultados del mapeo sistemático de la literatura con el propósito de analizar la literatura existente sobre las metodologías, métodos o estándares que abordan la gestión de riesgos en proyectos de software. Se seleccionaron 100 estudios primarios relevantes de un conjunto inicial de 132 artículos resultantes de la búsqueda realizada en IEEE Xplore, ACM, Google Academic, ScienceDirect y SEDICI, en el período comprendido entre el año 2008 y 2018. Una vez analizados los estudios primarios, se concluye que:

- CMMI, PMBOK y Software Risk Evaluation (SRE), se encuentran en un primer rango de menciones con un 25,45%, 23,64% y 20,91% respectivamente.
- En segundo lugar, con menor grado de representatividad, se encuentran las técnicas de Inteligencia Artificial (10%), Risk Management Frameworks (5,45%), Prince2 (4,55%), AS/NZS 4360(2,73%), ISO 31000 (2,73%), ISO 12207(1,82%), Risk IT (1,82%) y Magerit (0,91%).
- De los estudios primarios analizados, el 45% al tipo de investigación "Experiencia personal", el 32% corresponden al tipo de investigación "Propuesta de solución", el 15% al tipo "Evaluación" y el 8% restante al tipo de investigación "Validación".

2.1.4. AMENAZAS A LA VALIDEZ DEL SMS

Las principales amenazas a la validez de un SMS son: el sesgo en la selección de artículos, la inexactitud en la extracción de datos y la clasificación errónea. Para mitigar esta amenaza los estudios primarios que parecían dudosos han sido discutidos con los directores de la tesis. Se utilizaron múltiples fuentes digitales incluyendo revistas y conferencias relevantes en el campo de la Ingeniería de software. El alcance de las revistas y conferencias que se tratan en este SMS es suficientemente amplio para alcanzar una exhaustividad razonable en el campo estudiado. No se incluyeron documentos adicionales como literatura gris (informes técnicos, blogs, informes de investigación, memorias, proyectos, patentes, normas, traducciones científicas, documentos de sociedades científicas, boletines, cuadernos de trabajo, programas de computación, autobiografías, catálogos de productos y servicios de empresas, dosieres, carteles, etc.), ya que el propósito del SMS ha sido identificar las contribuciones realizadas desde la comunidad científica.

Para garantizar un proceso de selección objetivo, se definieron los criterios de inclusión y exclusión y el proceso de extracción y un protocolo de revisión.

La duplicación de artículos es una amenaza potencial a la hora de seleccionar los artículos, pero esto quedó resuelto dado que participaron los directores de la tesis en la selección de los estudios primarios.

2.2 COMPARATIVA DE METODOLOGÍAS, MÉTODOS Y ESTÁNDARES QUE ABORDAN LA GESTIÓN DE RIESGO

En esta sección, se definen las metodologías, métodos y estándares que abordan la gestión de riesgos que se emplean para el análisis comparativo junto a la justificación de su elección (sección 2.2.1), y

se usan los lineamientos del método DESMET (Kitchenham B. et al, 1996) con una adecuación a este trabajo para la evaluación de las metodologías, métodos y estándares considerados (sección 2.2.2).

2.2.1 METODOLOGÍAS, MÉTODOS Y ESTÁNDARES CONSIDERADOS QUE ABORDAN LA GESTIÓN DE RIESGOS

Con base en el reporte de resultados del SMS realizado, se consideran para el análisis comparativo, las metodologías, métodos o estándares que abordan la gestión de riesgos en proyectos de software con mayor representatividad en los estudios primarios. Estas son CMMI (CMMI Institute, 2010), PMBOK (Project Management Institute, 2013) y Software Risk Evaluation (Software Engineering Institute, 1999). No obstante, se decidió considerar la metodología Magerit V3 (Portal de administración electrónica, 2012) dado que es una de las pioneras en análisis y gestión de riesgos de un sistema de información. La misma fue elaborada por el Consejo Superior de Administración Electrónica publicando su primera versión en 1997. Por otro lado, si bien la inteligencia artificial permite gestionar riesgos en proyectos de desarrollo de software, no consideramos técnicas en esta comparativa sino solo propuestas metodológicas.

2.2.2 ANÁLISIS COMPARATIVO

El análisis comparativo de las metodologías, métodos y estándares que abordan la gestión de riesgos en proyectos de software, se descompone en tres partes, en la primera se definen las características a evaluar en cada una de las metodologías, métodos y estándares considerados como así también su justificación (sección 2.2.2.1). En una segunda parte, se describen las particularidades de las metodologías, métodos y estándares (sección 2.2.2.2). En la tercera y última parte, se presenta la evaluación de las metodologías, métodos y estándares junto con la interpretación de los resultados obtenidos (sección 2.2.2.3).

2.2.2.1 DEFINICIÓN Y JUSTIFICACIÓN DE LAS CARACTERÍSTICAS A EVALUAR

Para realizar la comparativa de las metodologías, métodos y estándares que abordan la gestión de riesgos, se utiliza el método DESMET, del cual se selecciona el método la evaluación cualitativa basada en el análisis de características (Kitchenham B. et al, 1996). Este análisis comparativo, tiene el propósito de dar respuesta a la siguiente pregunta de investigación: *¿Los riesgos del proceso de despliegue de un sistema software se encuentran cubiertos en las metodologías, métodos y estándares existentes?*

2.2.2.2 DESCRIPCIÓN DE LAS METODOLOGÍAS, MÉTODOS Y ESTÁNDARES

En esta sección, se presenta una descripción de las particularidades de las metodologías, métodos y estándares considerados, se focaliza en el proceso de despliegue como proceso dentro del proceso de desarrollo de software. Se consideran los aspectos de gestión del proyecto software por las interacciones que se presentan con el proceso de despliegue.

2.2.2.2.1 CMMI

Los modelos CMMI (Capability Maturity Model Integration por sus siglas en inglés) son colecciones de buenas prácticas para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. Estos modelos son desarrollados por equipos de producto con miembros procedentes de la industria, del gobierno y del Software Engineering Institute (CMMI Institute, 2010).

En la actualidad hay tres áreas de interés cubiertas por los modelos de CMMI: Desarrollo, Adquisición y Servicios.

- CMMI para el Desarrollo (CMMI-DEV) donde se tratan procesos de desarrollo de productos y servicios.
- CMMI para la adquisición (CMMI-ACQ) donde se tratan la gestión de la cadena de suministro, adquisición y contratación externa en los procesos del gobierno y la industria.
- CMMI (CMMI-SVC), está diseñado para cubrir todas las actividades que requieren gestionar, establecer y entregar Servicios.

Existen 5 niveles de madurez en CMMI:

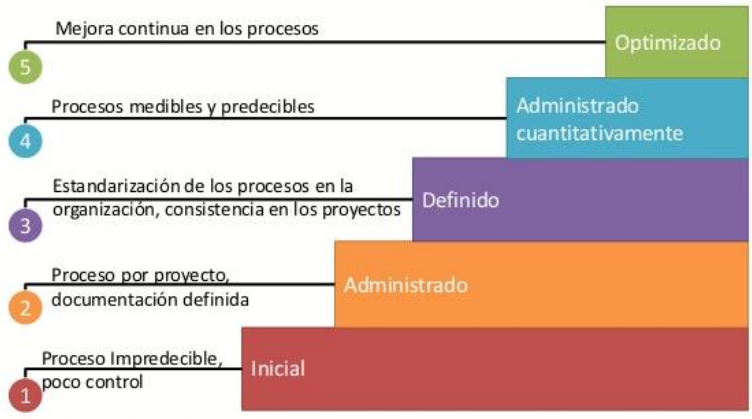


Figura 2.4. Niveles de madurez de CMMI (CMMI Institute, 2010).

- Nivel 1 o Inicial: Ambiente impredecible donde las organizaciones no tienen actividades de control y no están diseñadas.
- Nivel 2 o Administrado: Las actividades de control existen, pero no se ponen en práctica. Los controles dependen básicamente de las personas. No hay un entrenamiento formal ni comunicación de las actividades de control.
- Nivel 3 o Definido: Las actividades de control existen y están diseñadas, han sido documentadas y comunicadas a los empleados, las desviaciones de las actividades de control probablemente no se detecten.
- Nivel 4: Administrado cuantitativamente: Se utilizan herramientas en una forma limitada para soportar las actividades de control.
- Nivel 5 u Optimizado: Es una estructura integrada de control interno con un monitoreo en tiempo real por la gerencia, así como mejoras continuas-auto control, se encuentran cambios más rápidos al momento de detectar errores en los manejos de las actividades o en las personas.

2.2.2.2.2 PMBOK

La guía PMBOK (Project Management Body of Knowledge por sus siglas en inglés) es un instrumento desarrollado por el Project Management Institute, que establece un criterio de buenas prácticas relacionadas con la gestión, la administración y la dirección de proyectos mediante la implementación de técnicas y herramientas que permiten identificar un conjunto de 47 procesos, distribuidos en 5 grupos de procesos (Project Management Institute, 2013).

Se considera frecuentemente como manual de buenas prácticas, las alusiones y remisiones a la guía del proyecto PMBOK son tan universales como necesarias en el ámbito de la dirección y la gestión de proyectos, un ámbito que en el PMBOK se presenta como la convergencia de dos aspectos fundamentales: grupos de procesos, que agrupan todos los procesos y las actividades implicadas en proyectos estandarizados, y áreas de conocimiento, es decir, aquellos aspectos clave cuya consideración debe intervenir en cada uno de los grupos de procesos establecidos.

La guía PMBOK identifica 5 grupos de procesos en los que se incluyen los 47 procesos estándares que intervienen en cualquier proyecto:

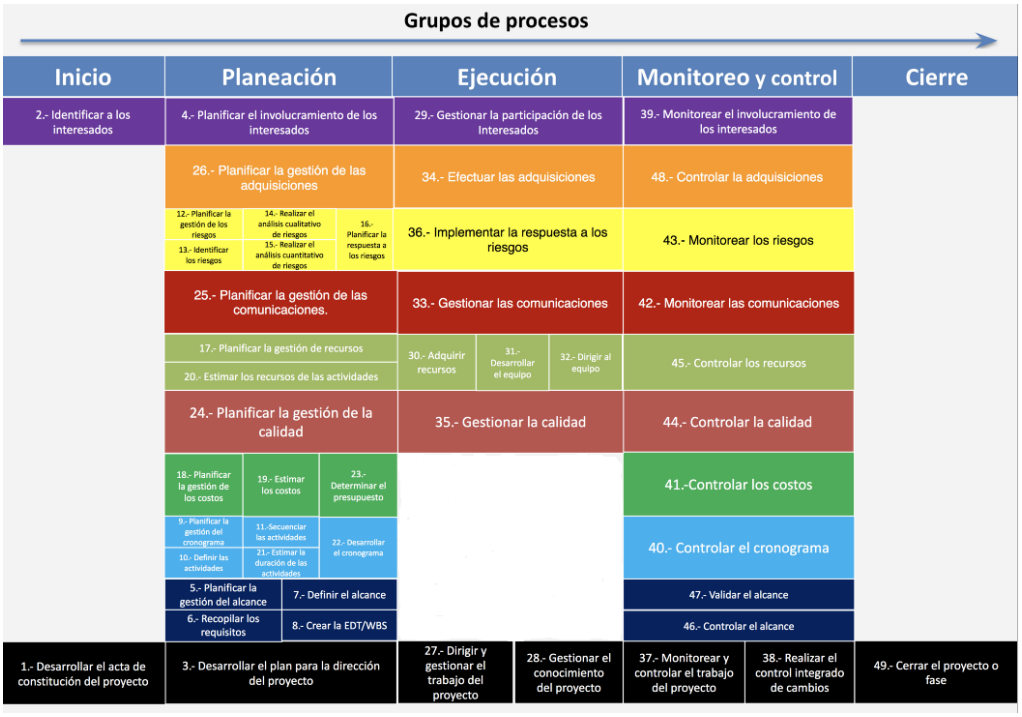


Figura 2.5. Grupos de procesos y actividades de PMBOK (Project Management Institute, 2013).

- Inicio: conformado por 2 procesos menores, cuyo fin es definir un nuevo proyecto o una nueva fase de ejecución de este, y obtener la autorización necesaria para llevarlo a cabo.
- Planeación: este grupo de procesos incluye 24 procesos destinados a la concreción y el establecimiento de objetivos, y al diseño de las estrategias más adecuadas para lograr su consecución.
- Ejecución: incluye 10 procesos implicados en el correcto desempeño, acorde a la estrategia adoptada, de las actividades definidas en el proyecto para la consecución de los fines establecidos.
- Monitoreo y control: doce procesos se inscriben en estos grupos de procesos, todos ellos relacionados con la supervisión y la evaluación del desempeño del proyecto.
- Cierre: formado por un proceso, que cierra el proyecto en su totalidad o alguna fase del mismo refiriendo el grado de aceptación y la satisfacción con el resultado obtenido.

En cada uno de estos grupos de procesos intervienen 10 áreas de conocimiento, que en la guía PMBOK se enuncian y describen del siguiente modo:

- Integración: área directamente relacionada con la dirección de proyectos. Establece los criterios para la correcta gestión, administración y coordinación de los distintos procesos y actividades implicadas.

- Alcance: determina el alcance del proyecto, definiendo todos y cada uno de los procesos y las actividades que se hallan implicados.
- Tiempo: gestión del tiempo de ejecución de los procesos implicados en el proyecto, y monitorización de estos con el fin de cumplir los plazos establecidos.
- Costes: gestión de los costes del proyecto y control de los mismos para mantenerlos dentro de su presupuesto inicial.
- Calidad: determina responsabilidades en los resultados de las actividades y los procesos implicados en el proyecto y en sus fases, y establece las políticas de calidad a las que debe remitirse la evaluación de dichos resultados. Sobre esta área tan fundamental, es altamente recomendable la lectura de la guía las 7 herramientas de calidad imprescindibles, disponible completamente gratis en nuestro apartado de recursos.
- Recursos humanos: gestión y dirección del/los equipos humanos implicados en el proyecto o en cada una de sus fases concretas.
- Comunicaciones: área responsable de la gestión y la administración de los mecanismos, las informaciones, las vías y las estrategias de comunicación entre las distintas estructuras y áreas internas del proyecto, así como de la elaboración de la información sobre el mismo orientada al exterior.
- Riesgos: atiende a la detección, gestión y solución de los riesgos implicados en cada uno de los procesos y fases de los mismos.
- Adquisiciones: área de gestión de procesos de compra de bienes, estructuras, herramientas o servicios externos a los equipos implicados en el proyecto.
- Stakeholders: se refiere a la gestión de los interesados o posibles inversores, a la correcta administración de las expectativas generadas con el proyecto y a la definición de las posibilidades de intervención en el mismo por parte de terceros.

2.2.2.2.3 SOFTWARE RISK EVALUATION

El Método de Evaluación de Riesgo de Software (en inglés Software Risk Evaluation Method o SRE) es un proceso para identificar, analizar y desarrollar estrategias de mitigación de riesgos en un sistema de software mientras está en desarrollo (Software Engineering Institute, 1999).

El proceso de SRE ha estado en desarrollo evolutivo en el SEI desde 1992 y se ha utilizado en más de 50 contratistas del Departamento de Defensa (DoD) y civiles (federales y estatales) y en las oficinas de programas. La versión 1.0 de la Descripción del Método SRE se publicó en diciembre de 1994.

La descripción del método SRE proporciona:

- Una descripción de los principios del método SRE, incluidos conceptos y aplicaciones útiles.
- Información adicional sobre el proceso de SRE para que una organización pueda personalizar de manera responsable el proceso para sus propias necesidades.
- Listas específicas de "resultados clave" para cada paso del proceso que se pueden usar para evaluar la calidad de la ejecución.

La descripción debe permitir a los miembros del personal de mejora de procesos de una Organización realizar una SRE inicial de manera competente sin ayuda externa, y luego mejorar continuamente su proceso a lo largo del tiempo.

2.2.2.2.4 MAGERIT

Magerit (Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información) es la metodología de análisis y gestión de riesgos elaborada por el Consejo Superior de Administración Electrónica de España, como respuesta a la percepción de que la Administración, y, en general, toda la sociedad, dependen de forma creciente de las tecnologías de la información para el cumplimiento de su misión (Portal de administración electrónica, 2012).

La razón de ser de Magerit está directamente relacionada con la generalización del uso de las tecnologías de la información, que supone unos beneficios evidentes para los ciudadanos; pero también da lugar a ciertos riesgos que deben minimizarse con medidas de seguridad que generen confianza.

Magerit interesa a todos aquellos que trabajan con información digital y sistemas informáticos para tratarla. Si dicha información, o los servicios que se prestan gracias a ella, son valiosos, Magerit les permitirá saber cuánto valor está en juego y les ayudará a protegerlo. Conocer el riesgo al que están sometidos los elementos de trabajo es, simplemente, imprescindible para poder gestionarlos. Con Magerit se persigue una aproximación metódica que no deje lugar a la improvisación, ni dependa de la arbitrariedad del analista.

Magerit persigue los objetivos que se detallan en la Figura 2.6:

Figura 2.6. Objetivos de Magerit (Portal de administración electrónica, 2012).

2.2.2.3 EVALUACIÓN DE LAS METODOLOGÍAS, MÉTODOS Y ESTÁNDARES

Para este trabajo se consideró adecuada una visión tridimensional del proceso de despliegue (Vázquez P. et al., 2018): "Proceso/Producto/Persona". La primera dimensión denominada "Proceso" incluye las fases o etapas, actividades y tareas que componen el proceso de despliegue. La segunda dimensión denominada "Producto", contempla características tales como tamaño, complejidad, características de diseño, rendimiento y nivel de calidad. La última dimensión, "Persona", incluyen tanto a profesionales informáticos como usuarios del sistema.

Con base en los resultados del SMS, y con el objetivo de evidenciar de qué manera CMMI (CMMI Institute, 2010), PMBOK (Project Management Institute, 2013) y Software Risk Evaluation (Software Engineering Institute, 1999), cubren la fase de despliegue, se realizó un análisis comparativo mediante el método DESMET (Kitchenham B. et al, 1996) basado en las características: "Proceso/Producto/Persona". Adicionalmente se considera agregar al estudio a Magerit (Portal de administración electrónica, 2012) por tratarse de una de las metodologías pioneras en gestión de riesgos. El análisis comparativo se presenta en la Tabla 2.8.

Fase de despliegue	Metodologías, métodos o estándares considerados			
	CMMI-DEV	PMBOK	S.R.E.	MAGERIT
Dimensión "Proceso"	SI	SI	SI	SI
Dimensión "Producto"	NO	SI	SI	SI
Dimensión "Persona"	NO	NO	NO	NO

Tabla 2.8. Evaluación de las metodologías, métodos y estándares que abordan la gestión de riesgos.

Los resultados permitieron concluir que en la dimensión “Proceso” todas las metodologías, métodos y estándares analizados abordan los riesgos para el proceso de despliegue. En relación con la dimensión “Producto” tanto SOFTWARE RISK EVALUATION como PMBOK y MAGERIT contemplan los riesgos del proceso de despliegue mientras que CMMI no y finalmente, en la dimensión “Persona”, ninguna de las metodologías, métodos o estándares evaluados aborda los riesgos de proceso de despliegue.

2.3 CONCLUSIONES DEL ESTADO DEL ARTE.

La construcción del estado del arte mediante un método de investigación como el mapeo sistemático de la literatura (SMS) permitió evidenciar las metodologías, métodos y estándares que abordan la gestión de riesgo en proyectos de desarrollo de software. Entre estos se encuentran, CMMI, PMBOK y Software Risk Evaluation (SRE), en un primer rango de menciones con un 25,45%, 23,64% y 20,91% respectivamente y con menor grado de representatividad, las técnicas de Inteligencia Artificial (10%), Risk Management Frameworks (5,45%), Prince2 (4,55%), AS/NZS 4360 (2,73%), ISO 31000 (2,73%), ISO 12207 (1,82%), Risk IT (1,82%) y Magerit (0,91%).

Una vez identificadas las contribuciones existentes, con el propósito de identificar los aspectos que cada una de ellas cubrían, se realizó una evaluación basada en características en base a los lineamientos del método DESMET. En este análisis comparativo se utilizaron como características las dimensiones que se consideran que definen al proceso de despliegue “Proceso, Producto y Persona” a lo que se denominó visión tridimensional, lo cual permitió identificar de qué manera las metodologías, métodos y estándares que abordan la gestión de riesgos en el proceso de despliegue de sistemas de software.

Del estado de arte presentado, surge la vacancia de la gestión de riesgos específicos para el proceso de despliegue de sistemas de software, razón por la cual se decide fortalecer este proceso mediante la definición de un conjunto de riesgos que permitan la gestión de los mismos para el proceso de despliegue considerando la visión tridimensional del proceso; es decir riesgos para el proceso, riesgos para el producto y riesgos para el peopleware que participa en el proceso.

3. PROPUESTA DE RIESGOS

En este capítulo se describe el proceso realizado para la construcción de la propuesta de riesgos de esta Tesis, así como también cuales han sido los estándares y metodologías en las cuales se basa y la justificación de la selección de cada una de estas. En la sección 3.1 se presentan las actividades y tareas del proceso de despliegue que se consideran en la solución (Sección 3.1). A continuación, en la sección 3.2 se presenta una clasificación de riegos que se ha considerado como base para esta propuesta. La sección 3.3 explica la visión tridimensional del proceso de despliegue "Proceso, Producto y Persona" sobre la cual se definen los riesgos para cada una de estas dimensiones en la sección 3.4, la sección 3.5 presenta el método de ponderación de estos riesgos y, por último, se presenta una herramienta para el dimensionamiento de los riesgos del proceso de despliegue (Sección 3.6).

3.1 ACTIVIDADES Y TAREAS DE LA ISO 12207

Para la construcción de la propuesta de riesgos, se contemplaron las actividades y tareas del **"proceso de transición"** del estándar ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE 12027, 2017) por ser un estándar reconocido internacionalmente. Este estándar tiene como objetivo principal proporcionar una estructura común para que compradores, proveedores, desarrolladores, personal de mantenimiento, operadores, gestores y técnicos involucrados en el desarrollo de software utilicen un lenguaje común (ISO/IEC/IEEE 12207, 2017). La Figura 3.1. presenta los procesos que conforman el estándar ISO/IEC/IEEE/12207.

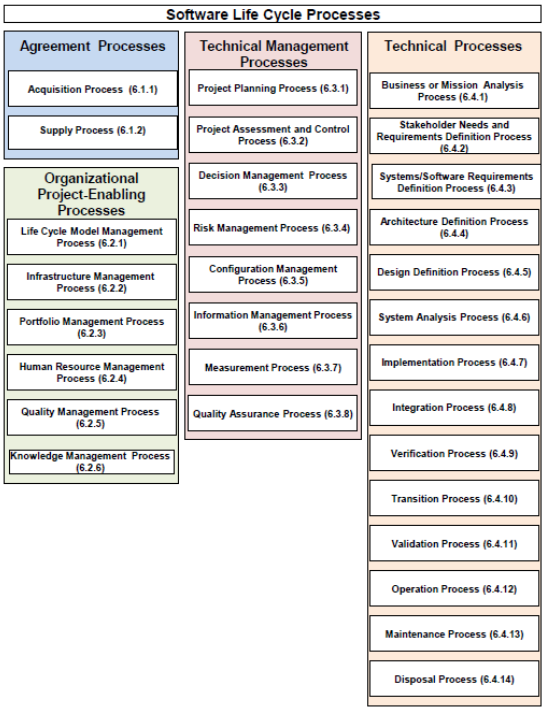


Figura 3.1. Procesos de la norma ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE, 2017).

El grupo de procesos de la ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE, 2017) se compone de los procesos de acuerdo, los procesos organizacionales del proyecto, los procesos del proyecto y por último, los procesos técnicos:

- **Procesos de acuerdo:**
  - Proceso de adquisición.
    - Satisfacer las necesidades del cliente.
    - Identificar necesidades del cliente.
    - Aceptación del producto o servicio.
  - Proceso de suministro.
    - Comprar productos y/o servicios acordes a requisitos establecidos
- **Procesos organizacionales del proyecto:**
  - Proceso de gestión del modelo de ciclo de vida
    - Políticas procesos y procedimientos para el ciclo de vida.
    - Requisitos para su gestión (definición, objetivos, mejora continua, etc.).
  - Proceso de gestión de infraestructuras.
    - Recursos de soporte de procesos durante el ciclo de vida (instalaciones, herramientas, tecnologías, etc.).
  - Proceso de gestión de la cartera de proyectos.

        - Requisitos para justificar la asignación continua de recursos a los proyectos para garantizar los objetivos de una organización.
    - Proceso de gestión de recursos humanos.
        - Requisitos para asegurar la cualificación del personal asignado a los procesos del ciclo de vida.
    - Proceso de gestión de la calidad.
        - Requisitos para alcanzar los objetivos de calidad.

- **Procesos del proyecto:**
    - Proceso de planificación del proyecto.
        - Establece requisitos para:
            - Identificar alcance del proyecto.
            - Identificar tareas y salidas de los procesos.
            - Establecimiento de planes y recursos.
    - Proceso de evaluación y control del proyecto.
        - Requisitos para control del proyecto.
        - Planificación.
        - Costos.
        - Objetivos técnicos.
        - Desviaciones.
    - Proceso de gestión de la decisión.
        - Requisitos de soporte para la toma de decisiones.
    - Proceso de gestión de riesgos.
        - Requisitos para control y monitorización continua de riesgos.
    - Proceso de gestión de la configuración.
        - Requisitos para la integridad y disponibilidad de las salidas de un proyecto.
    - Proceso de gestión de la información.
        - Requisitos para mantener toda la información relevante acerca de los procesos y garantizar su disponibilidad y confidencialidad.
    - Proceso de medición.
        - Requisitos para recolectar y analizar los datos que soportan objetivamente la calidad de los productos y la gestión efectiva de los procesos.

- **Procesos técnicos:**
    - Proceso de definición de requisitos de las partes interesadas (stakeholders).
        - Requisitos para identificar y satisfacer los intereses y de las partes interesadas.

- Proceso del análisis de requisitos del sistema.
  - Requisitos para definir los requisitos técnicos del sistema.
- Proceso de implementación o puesta en funcionamiento.
- Proceso de integración del sistema.
  - Requisitos para integración de los elementos de un sistema:
  - Elementos Software.
  - Hardware.
  - Manuales.
  - Etc.
- Proceso de comprobación de los requisitos del sistema.
  - Requisitos para realizar la comprobación de la conformidad.
- Proceso de instalación del software.
  - Requisitos para instalar el producto software en un entorno objetivo.
- Proceso de apoyo a la aceptación del software.
  - Requisitos para establecer procesos de asistencia que garanticen la satisfacción y confianza del comprador.
- Proceso de operación del software.
  - Requisitos para establecer procesos de ayuda a la operación del sistema.
- Proceso de mantenimiento del software.
  - Requisitos para proveer soporte a coste efectivo del producto software.
- Proceso de retiro del software
  - Retirar un software de un sistema.
  - Terminar las operaciones de mantenimiento.
  - Mantenimiento del entorno después del retiro.
  - Establecimiento de responsabilidades.
  - Cumplimiento de la legislación.
  - Mantenimiento de registros de auditoria.

Para una mejor estructuración de la solución, así como también para que su aplicación en la industria se pueda realizar de manera sistemática en diferentes proyectos de despliegue se decide definir una codificación de la misma.

Se utiliza la propuesta de Runeson et al. (Runeson P. et al., 2012) que propone lineamientos para el diseño de un esquema de codificación para el análisis e interpretación de los datos en los estudios de casos. Estos lineamientos se detallan a continuación:

- Codificar tanto como sea posible.

- Los códigos deben ser priorizados de la siguiente manera:
    - Códigos de alto nivel, basados en preguntas de investigación.
    - Códigos de nivel medio, basados en agrupaciones de códigos: categorías de códigos.
    - El código de bajo nivel es su interpretación del texto (en el campo Comentarios).

De los tipos de codificación propuestos se utiliza el segundo tipo que permite realizar agrupaciones y en este punto de la solución permite codificar los tres grupos de actividades y las tareas que conforman el proceso transición del estándar ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE 12027, 2017).

La codificación se aplica de la siguiente manera:

- Código primario: Actividad (A)
- Código Secundario: Tareas (T)

En la Tabla 3.1, se presenta la codificación resultante para las actividades y las tareas del proceso de despliegue.

Actividad	Tareas
A1 Preparación del despliegue.	T1 Identificar restricciones tecnológicas.
	T2 Disponer de acceso a los entornos sistemas o servicios habilitados.
	T3 Analizar políticas y estándares existentes.
	T4 Definir las políticas de pruebas unitarias.
	T5 Definir prioridades para el despliegue para respaldar la migración y transición de datos y software.
	T6 Identificar restricciones de la estrategia de despliegue
A2 Realización del despliegue.	T7 Realizar o adaptar los elementos de software de acuerdo con la estrategia de despliegue.
	T8 - Registrar evidencia de cumplimiento de los requerimientos.
	T9 Adaptar elementos de hardware y servicios de software.
	T10 Formación de los empleados.
A3 Gestión de los resultados del despliegue.	T11 Registrar los resultados y las anomalías encontradas.
	T12 Mantener la trazabilidad.
	T13 Proporcionar artefactos clave.
	T14 Documentar cumplimiento de expectativas y funcionalidades.
	T15 Evaluar la necesidad u oportunidad de mejoras.

Tabla 3.1 Actividades y las tareas del proceso “Transición”

3.2 TIPIFICACIÓN DE RIESGOS SEGÚN CAPERS JONES

Se utilizaron los diferentes tipos de riesgos propuestos por Capers Jones (Jones C., 1994). Estos surgen de él, sus colegas y muchos de sus clientes quienes se dedican casi a diario a realizar evaluaciones formales de tecnologías de desarrollo de software en los Estados Unidos y en el extranjero. Como resultado de la evaluación de proyectos de software en varios cientos de empresas, han observado y clasificado aquellos riesgos que se presentan de forma constante en este tipo de proyectos como “los riesgos más comunes en proyectos de software”. En la Tabla 3.2 se presenta los tipos de riesgos.

Número	Riesgo
1	Objetivos de mejora ambiguos.
2	Falso niveles de madurez.
3	Cancelación del proceso de despliegue.
4	Nula o inexistente normativa corporativa.
5	Sobrecostos.
6	Requisitos de usuario progresivos.
7	Oficinas abarrotadas.
8	Módulos propensos a errores.
9	Documentación escasa.
10	Agenda o plan de trabajo reducido.
11	Exceso de tiempo de comercialización.
12	Falsas quejas de productividad.
13	Fricción entre el cliente y la empresa proveedora de software.
14	Fricción entre la gestión de software y los altos ejecutivos.
15	Costos de mantenimientos altos.
16	Estimación de costos inexacta.
17	Métricas inadecuadas.
18	Inadecuada estimación de calidad.
19	Dimensionamiento inexacto de los entregables.
20	Evaluaciones inadecuadas.
21	Planes de compensación inadecuados.
22	Repositorios de proyectos inadecuados.
23	Planes de capacitación inadecuados (Ingeniería de software).
24	Planes de capacitación inadecuados (Gestión de proyectos).
25	Mediciones inadecuadas.
26	Métodos de adquisición de paquetes inadecuados.
27	Instalaciones inadecuadas de investigación y referencia.
28	Políticas y estándares de software inadecuados.
29	Análisis de riesgos de proyectos de software inadecuado.
30	Inadecuado análisis de valor del proyecto de software inadecuado.
31	Herramientas y métodos de gestión de despliegue inadecuados (gestión de proyectos).

Número	Riesgo
32	Herramientas y métodos de gestión de despliegue inadecuados (Aseguramiento de la calidad).
33	Herramientas y métodos de gestión de despliegue inadecuados (Ingeniería de software).
34	Herramientas y métodos de gestión de despliegue inadecuados (Documentación técnica).
35	Falta de arquitectura reutilizable.
36	Falta de código reutilizable.
37	Falta de datos reutilizables.
38	Falta de diseños reutilizables.
39	Falta de documentación reutilizable.
40	Falta de estimaciones reutilizables (plantillas).
41	Falta de interfaces reutilizables.
42	Falta de planes de proyectos reutilizables.
43	Falta de requisitos reutilizables
44	Falta de planes de prueba, casos de prueba y datos de prueba reutilizables.
45	Falta de especialización.
46	Larga vida útil de los sistemas obsoletos.
47	Baja productividad.
48	Baja calidad.
49	Baja performance del personal y la gestión del software.
50	Baja satisfacción del usuario.
51	Mala práctica profesional (gestión de proyectos).
52	Mala práctica profesional (personal técnico).
53	Programación errónea.
54	Definiciones parciales del ciclo de vida.
55	Estructuras de organización deficientes.
56	Pocas inversiones en tecnología.
57	Inadecuada política de retención de personal.
58	Planificación de mejoras a corto plazo.
59	Lenta transferencia de tecnología.

Tabla 3.2. Riesgos más comunes en proyectos de software según Jones (Jones C., 1994).

Ha sido necesario realizar algunas adecuaciones a la propuesta de Capers Jones debido a la evolución de la Ingeniería de Software en las últimas décadas. La principal adecuación consistió en transpolar los riesgos que aplican a todo el ciclo de vida del software al proceso de despliegue de sistemas de software.

3.3 TAXONOMÍA DE RIESGOS

Para organizar y focalizar los riesgos por algún tipo de clasificación, se utilizó la propuesta que adopta una visión de riesgos basada en tres enfoques también denominada “visión tridimensional” del

proceso de despliegue propuesta por Vázquez et al. (Vazquez P., et al., 2018). Esta considera una primera dimensión denominada: ¨*Proceso*¨ dado que resultan de interés las fases o etapas, actividades y tareas que lo componen. La segunda dimensión denominada ¨*Producto*¨, la cual contempla las características como la complejidad del producto a instalar, los requisitos de instalación para el producto software, la integración con la infraestructura del cliente, el tamaño, entre otras. La última dimensión denominada ¨*Persona*¨, debido a la existencia del peopleware y su impacto en el proceso de implantación de sistemas informáticos (Panizzi M., et al., 2017).

3.4 RIESGOS DEFINIDOS PARA EL PROCESO DE DESPLIEGUE

Para cada una de las actividades y tareas del estándar ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE, 2017) descriptas en la sección 3.1 y persiguiendo una visión tridimensional del proceso de despliegue "Proceso, Producto y Persona", se definieron los riesgos que se describen en las secciones siguientes. De la misma manera que para la sección 3.3. se decidió utilizar una codificación para las actividades y tareas del proceso transición en este punto de la solución se realiza la codificación para los tres grupos de riesgos para el proceso de despliegue de sistemas de software, resultando la siguiente codificación:

- RPROC (Riesgos para la dimensión proceso),
- RPROD (Riesgos para la dimensión producto) y
- RPERSO (Riesgos para la dimensión persona),

3.4.1 RIESGOS PARA LA DIMENSIÓN "PROCESO"

En la Tabla 3.3, se presenta para cada una de las actividades y sus respectivas tareas, los riesgos propuestos para la dimensión "Proceso" y en la Tabla 3.4 se presenta una descripción de cada uno de los riesgos propuestos para esta dimensión.

Actividad	Tareas	Riesgos
A1 - Preparación del despliegue.	T1 Identificar restricciones tecnológicas.	RProc1 Pocas inversiones en tecnología.
	T2 Disponer de acceso a los entornos sistemas o servicios habilitados.	RProc2 Fricción entre la gestión de software y los altos ejecutivos.
	T3 Analizar políticas y estándares existentes.	RProc3 Nula o inexistente normativa corporativa.
	T4 Definir las políticas de pruebas unitarias.	RProc4 Planes de prueba mal elaborados.
	T5 Definir prioridades para el despliegue para respaldar la migración y transición de datos y software.	RProc5 Agenda o plan de trabajo reducido.
	T6 Identificar restricciones de la estrategia de despliegue.	RProc6 Cancelación del proceso de despliegue.
A2 - Realización del despliegue.	T7 Realizar o adaptar los elementos de software de acuerdo con la estrategia de despliegue.	RProc7 Fricción entre el cliente y la empresa proveedora de software.
	T8 Registrar evidencia de cumplimiento de los requerimientos.	RProc8 Métricas inadecuadas.
	T9 Adaptar elementos de hardware y servicios de software.	RProc9 Sobrecostos.
	T10 Formación de los empleados.	RProc10 Planes de capacitación inadecuados.
A3 – Gestión de los resultados del despliegue.	T11 Registrar los resultados y las anomalías encontradas.	RProc11 Herramientas y métodos de gestión de despliegue inadecuados.
	T12 Mantener la trazabilidad.	RProc12 Repositorios inadecuados.
	T13 Proporcionar artefactos clave.	RProc13 Dimensionamiento inexacto de los entregables.
	T14 Documentar cumplimiento de expectativas y funcionalidades.	RProc14 Baja satisfacción del usuario.
	T15 Evaluar la necesidad u oportunidad de mejoras.	RProc15 Objetivos de mejora ambiguos.

Tabla 3.3 Riesgos de la Dimensión “Proceso” para cada una de las actividades y tareas.

En la Tabla 3.4, se presenta la descripción de los riesgos definidos para la dimensión “Proceso”.

Grupo	Categoría	Descripción
A1 Preparación del despliegue.	RProc1 Poca inversión en tecnología.	Más allá de la necesidad de cumplir normativas, la pobre inversión en tecnología es endémica a la industria y afecta a empresas de todos los tamaños.
	RProc2 Fricción entre la gestión de software y los altos ejecutivos.	La fricción entre los ejecutivos ocurre en la mayoría de las grandes empresas debido a objetivos y/o necesidades opuestas.
	RProc3 Nula o inexistente normativa corporativa.	La falta de políticas corporativas genera confusión y reglas poco claras que podrían generar graves inconvenientes durante el Proyecto.
	RProc4 Planes de prueba mal elaborados.	La errónea documentación de los planes de prueba puede generar que durante las mismas no se tenga en cuenta todos los casos de uso necesarios.
	RProc5 Agenda o plan de trabajo reducido.	Hitos anticipados o entregables tangibles que ocurren significativamente después de sus fechas planificadas y comprometidas.
	RProc6 Cancelación del proceso de despliegue.	Diversas restricciones o inconvenientes en la estrategia pueden generar la cancelación o suspensión de las actividades de despliegue.
A2 Realización del despliegue.	RProc7 Fricción entre el cliente y la empresa proveedora de software.	Enemistad o antagonismo personal que ocurre entre el cliente y los contratistas de software como resultado de malentendidos pueden generar retrasos o incluso la cancelación del proyecto de despliegue.
	RProc8 Métricas inadecuadas.	Métricas de software comunes que violan el estándar o que se comportan de manera paradójica, contra intuitiva o impredecible.
	RProc9 Sobrecostos.	Las consecuencias de los errores de estimación de costos y recursos suelen ser más graves cuando las estimaciones son bajas y se subestiman los recursos que realmente se necesitan.
	RProc10 Planes de capacitación inadecuados.	Una de las causas más importantes detrás de los fracasos en los despliegues es que algunos empleados no conozcan los beneficios del nuevo sistema.
A3 Gestión de los resultados del despliegue	RProc11 Herramientas y métodos de gestión de despliegue inadecuados.	Métodos automatizados de documentación sumados a enfoques metodológicos inadecuados, podrían generar errores en el registro de los resultados y las anomalías encontradas.

Grupo	Categoría	Descripción
A3 Gestión de los resultados del despliegue.	RProc12 Repositorios inadecuados.	Falta de capacidades formales y automatizadas para lidiar con la sincronización, referencias cruzadas, integración y actualización de software.
	RProc13 Dimensionamiento inexacto de los entregables.	Fallas en la estimación del tamaño de los componentes principales de software pueden generar inconvenientes en el despliegue de estos.
	RProc14 Baja satisfacción del usuario.	Los usuarios pueden no estar satisfechos con la facilidad de uso, capacitación, funcionalidad, calidad y confiabilidad del software entregado. Esto puede derivar en poca o nula utilización de este.
	RProc15 Objetivos de mejora ambiguos.	Objetivos para mejorar la productividad o la calidad del software que son abstractos o tan ambiguos que no hay forma de interpretarlos con precisión.

Tabla 3.4 Descripción de riesgos de la dimensión “Proceso”.

3.4.2 RIESGOS PARA LA DIMENSIÓN “PRODUCTO”

En la Tabla 3.5, se presenta para cada una de las actividades y sus respectivas tareas, los riesgos propuestos para la Dimensión “Producto” y en la Tabla 3.6 se presenta una descripción de cada uno de los riesgos propuestos para esta dimensión.

Actividad	Tareas	Riesgos
A1 Preparación del despliegue.	T1 Identificar restricciones tecnológicas.	RProd1 Tecnología novedosa o con poca adopción.
	T2 Disponer de acceso a los entornos sistemas o servicios habilitados.	RProd2 Incompatibilidad con la infraestructura existente.
	T3 Analizar políticas y estándares existentes.	RProd3 Falta de adaptación a nuevas tecnologías.
	T4 Definir las políticas de pruebas unitarias.	RProd4 Falta de componentes.
	T5 Definir prioridades de despliegue para respaldar la migración y transición de datos y software.	RProd5 Formato de datos incompatible.
	T6 Identificar restricciones de la estrategia de despliegue.	RProd6 Poca flexibilidad.
A2 Realización del despliegue.	T7 Realizar o adaptar los elementos de software de acuerdo con la estrategia de despliegue.	RProd7 Mayor complejidad.

Actividad	Tareas	Riesgos
A2 Realización del despliegue.	T8 Registrar evidencia de cumplimiento de los requerimientos.	RProd8 Fallas o Errores en el funcionamiento.
	T9 Adaptar elementos de hardware y servicios de software.	RProd9 Pérdida de características y/o funciones.
	T10 Formación de los empleados.	RProd10 Falta de conocimiento de las funcionalidades del producto.
	T11 Registrar los resultados y las anomalías encontradas.	RProd11 Documentación escasa.
A3 Gestión de los resultados del despliegue.	T12 Mantener la trazabilidad.	RProd12 Inconsistencias de versiones del producto.
	T13 Proporcionar artefactos clave.	RProd13 Funcionalidades incompletas.
	T14 Documentar cumplimiento de expectativas y funcionalidades.	RProd14 Baja calidad del producto software entregado.
	T15 Evaluar la necesidad u oportunidad de mejoras.	RProd15 Ausencia de pruebas de seguridad.

Tabla 3.5 Riesgos de la Dimensión “Producto”.

La Tabla 3.6 presenta la descripción para los riesgos de la dimensión “Producto”.

Grupo	Categoría	Descripción
A1 Preparación del despliegue.	RProd1 Tecnología novedosa o con poca adopción.	Este tipo de tecnología es propensa a fallas y falta de soporte.
	RProd2 Incompatibilidad con la infraestructura existente.	El hardware o el software de base puede no ser compatible con los requerimientos del producto software generando fallos en el despliegue o en su posterior utilización.
	RProd3 Falta de adaptación a nuevas tecnologías.	Si no se adaptan las políticas y/o procedimientos de la organización a la tecnología utilizada puede generar inconvenientes en el despliegue.
	RProd4 Falta de componentes.	Los cambios constantes en los componentes pueden generar que los mismos no sean correctamente trazados y no se encuentren disponibles durante el despliegue generando fallas y retrasos en el mismo.
	RProd5 Formato de datos incompatible.	Si la tecnología seleccionada no soporta el formato de datos a importar, puede generar pérdida de información parcial o total del entorno productivo durante el despliegue.
	RProd6 Poca Flexibilidad.	La falta de flexibilidad del producto para adaptarse a la estrategia de despliegue definida genera retrasos en el cronograma definido.

Grupo	Categoría	Descripción
A2 Realización del despliegue.	RProd7 Mayor complejidad.	El aumento de la complejidad del producto para adaptarlo a la estrategia de despliegue genera retrasos y sobrecostos.
	RProd8 Fallas o Errores en el funcionamiento.	La falta de análisis y registro del cumplimiento de todas las funcionalidades del producto puede generar inconvenientes debido a problemas o errores no detectados.
	RProd9 Pérdida de características y/o funciones.	La adaptación del producto al hardware existente puede generar la pérdida de funcionalidades.
	RProd10 Falta de conocimiento de las funcionalidades del producto.	La falta de conocimiento de los empleados de la organización acerca de las características y funcionalidades del producto puede derivar en una mala o poco eficiente utilización de estas.
A3 Gestión de los resultados del despliegue.	RProd11 Documentación escasa.	Una inadecuada documentación de los resultados y las anomalías encontradas durante el despliegue puede generar que los mismos se repitan o deriven en un funcionamiento errático.
	RProd12 Inconsistencias de versiones del producto.	Los conflictos en el versionado pueden generar errores e inconsistencias en el producto durante el proyecto despliegue.
	RProd13 Funcionalidades incompletas.	Si los artefactos clave se encuentran con funcionalidades incompletas, se producirán inconvenientes con la utilización de estos.
	RProd14 Baja calidad del producto software entregado.	Si no se documenta el cumplimiento de las funcionalidades, se puede ver afectada la calidad del producto final.
	RProd15 Ausencia de pruebas de seguridad.	Las pruebas de seguridad permiten añadir calidad al control de las vulnerabilidades, caso contrario, el producto quedará expuesto a múltiples tipos de ataques que afecten la seguridad de la información almacenada en el mismo.

Tabla 3.6 Descripción de riesgos de la dimensión “Producto”.

3.4.3 RIESGOS PARA LA DIMENSIÓN “PERSONA”

En la Tabla 3.7, se presenta para cada una de las actividades y sus respectivas tareas, los riesgos propuestos para la Dimensión “Persona” y en la Tabla 3.8 se presenta una descripción de cada uno de los riesgos propuestos para esta dimensión.

Actividad	Tareas	Riesgos
A1 Preparación del despliegue.	T1 Identificar restricciones tecnológicas.	RPers1 Falta de especialización en las tecnologías y procesos involucrados.
	T2 Disponer de acceso a los entornos sistemas o servicios habilitados.	RPers2 Usuarios sin permisos de acceso adecuados.
	T3 Analizar políticas y estándares existentes	RPers3 Inadecuada política de retención de personal.
	T4 Definir las políticas de pruebas unitarias.	RPers4 Desconocimiento funcional o del negocio por parte de los usuarios a cargo de las pruebas.
	T5 Definir prioridades de despliegue para respaldar la migración y transición de datos y software.	RPers5 Cambios constantes en las prioridades.
	T6 Identificar restricciones de la estrategia de despliegue.	RPers6 Esfuerzo y/o recursos adicionales.
A2 Realización del despliegue.	T7 Realizar o adaptar los elementos de software de acuerdo con la estrategia de despliegue.	RPers7 Poca experiencia en sistemas actuales.
	T8 Registrar evidencia de cumplimiento de los requerimientos.	RPers8 Falta de pericia.
	T9 Adaptar elementos de hardware y servicios de software.	RPers9 Mala práctica profesional.
	T10 Formación de los empleados.	RPers10 Baja significativa de recursos asignados al proyecto.
A3 Gestión de los resultados del despliegue.	T11 Registrar los resultados y las anomalías encontradas.	RPers11 Desconocimiento de gestión documental.
	T12 Mantener la trazabilidad.	RPers12 Criterios o interpretaciones diversas.
	T13 Proporcionar artefactos clave.	RPers13 Baja productividad.
	T14 Documentar cumplimiento de expectativas y funcionalidades.	RPers14 Falta de colaboración de los usuarios finales.
	T15 Evaluar la necesidad u oportunidad de mejoras.	RPers15 Bajo compromiso.

Tabla 3.7 Riesgos de la Dimensión “Persona”.

En la Tabla 3.8, se presenta la descripción de los riesgos definidos para la dimensión “Persona”.

Grupo	Categoría	Descripción
A1 Preparación del despliegue.	RPers1 Falta de especialización en las tecnologías involucradas.	No reconocer a tiempo la necesidad de habilidades especiales para la gestión del proceso de despliegue.
	RPers2 Usuarios sin permisos de acceso adecuados.	No gestionar con anticipación los permisos de acceso necesarios genera demoras e inconvenientes diversos en el avance del proceso de despliegue.
	RPers3 Inadecuada política de retención de personal.	La constante rotación de personal requiere de incentivos de acuerdo con la generación de los empleados.
	RPers4 Desconocimiento funcional o del negocio por parte de los usuarios a cargo de las pruebas.	No disponer de los usuarios claves puede generar pruebas inconclusas o irreales. Esto deriva en incumplimiento de los requerimientos y problemas de funcionalidad.
	RPers5 Cambios constantes en las prioridades.	Las modificaciones frecuentes en el cronograma de trabajo generan retrasos y malestar en el equipo de trabajo. Esto impacta negativamente en la calidad del producto software.
	RPers6 Esfuerzo y/o recursos adicionales.	La no identificación de restricciones puede generar la necesidad de mayor demanda laboral. Esto además de sobrecostos, genera retrasos en el proyecto debido a la necesidad de inducción del nuevo personal al proyecto.
A2 Realización del despliegue.	RPers7 Poca experiencia en sistemas actuales.	No disponer de conocimiento y experiencia en las tecnologías y procesos que serán reemplazados como parte del proceso de despliegue.
	RPers8 Falta de pericia.	No realizar de forma íntegra la revisión del cumplimiento de los requerimientos de usuario debido a falta de experiencia.
	RPers9 Mala práctica profesional.	Los errores y fallos al intentar adaptar hardware y servicios de software pueden generar inconvenientes en el sistema a instalar.
	RPers10 Baja significativa de recursos asignados al proyecto.	La reducción masiva de personal puede generar un impacto severo en la continuidad del proyecto de software.
A3 Gestión de los resultados del despliegue.	RPers11 Desconocimiento de gestión documental.	No disponer de metodologías de gestión documental genera heterogeneidad en los tipos y calidad de los documentos del proceso de despliegue.
	RPers12 Criterios o interpretaciones diversas.	La falta de criterios unificados o interpretaciones disimiles no permites establecer un modelo de trazabilidad estándar para todo el proceso de despliegue.

Grupo	Categoría	Descripción
A3 Gestión de los resultados del despliegue.	RPers13 Baja productividad.	La baja productividad en el proceso de despliegue puede generar entregables de baja calidad.
	RPers14 Falta de colaboración de los usuarios finales.	La falta de respuesta por parte de los usuarios finales puede generar resultados no satisfactorios en la medición de cumplimiento de las expectativas y las funcionalidades.
	RPers15 Bajo compromiso.	La falta de compromiso por parte de los usuarios no permite detectar obtener registros (feedback) sobre funcionalidades que no cumplan sus expectativas u oportunidades de mejora para el proceso de despliegue o en el caso del personal técnico, inconvenientes que se hayan registrado.

Tabla 3.8 Descripción de riesgos de la dimensión “Persona”.

3.5 MÉTODO DE PONDERACIÓN DE LOS RIESGOS

Para la ponderación de los riesgos definidos, se adoptó la propuesta del estándar ISO/IEC 31010:2009 (ISO/IEC/IEEE, 2009) por considerarse una de las principales referencias de gestión de riesgos para la industria de software a nivel internacional. De acuerdo al mismo, en el análisis de riesgos hay que trabajar con múltiples elementos que hay que combinar en un sistema para ordenarlo por importancia sin que los detalles, perjudiquen la visión de conjunto.

En este trabajo, se utiliza la siguiente escala para calificar el valor de los activos de software, la magnitud del impacto y la magnitud del riesgo:

- MB: Muy bajo
- B: Bajo
- M: Medio
- A: Alto
- MA: Muy alto

Para la obtener la ponderación de cada uno de los riesgos, se deben seguir los siguientes pasos:

Paso 1 - Estimación del impacto: calcular el impacto en base a tablas sencillas de doble entrada donde se modelan impacto, probabilidad y riesgo por medio de escalas cualitativas como se observa en la Figura 3.2.

impacto		*degradación*		
		1%	10%	100%
valor	MA	M	A	MA
	A	B	M	A
	M	MB	B	M
	B	MB	MB	B
	MB	MB	MB	MB

Figura 3.2. Escalas de impacto de riesgos ISO/IEC 31010:2009 (ISO/IEC/IEEE 31010, 2009).

Paso 2 - Estimación del riesgo: modelar impacto, probabilidad y riesgo por medio de escalas cualitativas como las que se muestran en la Figura 3.3:

escalas		
impacto	probabilidad	riesgo
MA: muy alto	**MA:** prácticamente seguro	**MA:** crítico
A: alto	**A:** probable	**A:** importante
M: medio	**M:** posible	**M:** apreciable
B: bajo	**B:** poco probable	**B:** bajo
MB: muy bajo	**MB:** muy raro	**MB:** despreciable

Figura 3.3. Escalas de riesgos ISO/IEC 31010:2009 (ISO/IEC/IEEE, 2009).

Paso 3 – Ponderación: ponderar los riesgos en base al impacto y la probabilidad de ocurrencia como se ilustra en la Figura 3.4:

riesgo		*probabilidad*				
		MB	B	M	A	MA
impacto	MA	A	MA	MA	MA	MA
	A	M	A	A	MA	MA
	M	B	M	M	A	A
	B	MB	B	B	M	M
	MB	MB	MB	MB	B	B

Riesgo = [Probabilidad * Impacto]

Figura 3.4. Escala de ponderación de riesgos de acuerdo con el estándar ISO/IEC 31010:2009 (ISO/IEC/IEEE 31010, 2009).

3.6 HERRAMIENTA PARA EL DIMENSIONAMIENTO DE LOS RIESGOS PARA EL PROCESO DE DESPLIEGUE

Con el objetivo de facilitar a las PyMEs de desarrollo de software de Argentina, el conjunto de riesgos definidos para el proceso de despliegue en las tres dimensiones consideradas, “Proceso”, “Producto” y “Persona”, se desarrolló una herramienta que permite ejecutar de forma práctica y simple el dimensionamiento de los riesgos del despliegue para un determinado proyecto de software. Esta se encuentra en el Apéndice C.

Además, la herramienta se encuentra a disposición de las PyMEs de la industria del software a través del siguiente enlace: https://riesgosdeldespliegue.blogspot.com/

4. VALIDACIÓN DE LA SOLUCIÓN

En este capítulo se describen dos estudios de caso desarrollados en PyMEs desarrolladoras de software de Argentina con el propósito de examinar la viabilidad de la aplicación del conjunto de riesgos definidos en el capítulo 3. El primer estudio de caso se desarrolló en una PyME que se dedica al desarrollo de sistemas de información a medida para clientes de diversos rubros, entre ellos financiero, automotriz, farmacéutico y banca. El segundo estudio de caso se desarrolló en una PyME que se dedica a al desarrollo de sistemas de información exclusivamente para el rubro farmacéutico. Según la clasificación de AFIP, la primera PyME se clasifica como "mediana tramo 2" y la segunda como "Pequeña"[3] en función de la cantidad de empleados. En ambos estudios de casos se presentan los procedimientos para la prevención, mitigación y/o transferencia de los riesgos del proceso de despliegue de sistemas de software. El estudio de caso 1 se presenta en la sección 4.1. y el estudio de caso 2 en la sección 4.2. Es importante mencionar que se tuvo acceso a la documentación del proyecto sujeto a un acuerdo de no revelar el nombre de la PyME, así como el compromiso de informar sobre los hallazgos y recomendaciones a considerar para la gestión de riesgos del proceso de despliegue.

4.1. ESTUDIO DE CASO 1

4.1.1 DISEÑO DEL ESTUDIO DE CASO

En esta sección, se describe el estudio de caso, siguiendo los lineamientos propuestos de Runeson et al. (Runeson P. et al., 2012). El objetivo principal consiste en examinar la viabilidad de la aplicación del conjunto de riesgos, así como también de los procedimientos para su prevención, mitigación y/o transferencia para el proceso de despliegue de sistemas de software en un entorno real con el propósito de refinarlos (si fuese necesario). Según la clasificación de Robson (Robson C., 2002) se enmarca dentro de los estudios exploratorios. Se trabajó con documentación del despliegue de entregas (release) de funcionalidades de un Portal de Recursos Humanos de una entidad bancaria realizado por una PyME de sistemas de la República Argentina.

4.1.2. PREGUNTAS DE INVESTIGACIÓN

Para alcanzar el objetivo se plantean las siguientes preguntas de investigación (PI):

[3] https://pymes.afip.gob.ar/estiloAFIP/pymes/ayuda/default.asp. Página disponible al 18/11/20.

PI1: ¿Se gestionaron adecuadamente los riesgos durante las actividades del proceso de despliegue de sistemas de software?

A través de esta pregunta, se busca obtener la información de los riesgos que se presentaron en la ejecución del proceso de despliegue y el tratamiento dado por la consultora para compararlos con la propuesta realizada.

PI2: ¿De qué manera se puede fortalecer el proceso de despliegue de sistemas software en esta empresa?

Con esta pregunta se intenta determinar la forma en que la consultora puede fortalecer su proceso de despliegue. Para esto se propone la identificación de un conjunto de riesgos junto con sus procedimientos de prevención, mitigación y/o transferencia.

4.1.3. CASO Y UNIDAD DE ANALISIS

Para el caso de estudio, se recopilaron datos de una consultora de sistemas PyME. Esta consultora se encuentra radicada en la Ciudad Autónoma de Buenos Aires, cuenta con un staff de unos 430 empleados y desarrolla sistemas de información a medida para clientes de diversos rubros, entre ellos financiero, automotriz, farmacéutico y banca. En los proyectos de software de esta consultora se combinan prácticas ágiles con metodologías de desarrollo con ciclo de vida iterativo.

El estudio del caso se centra en el análisis de riesgos del despliegue de entregas (release) de un Portal de Recursos Humanos realizado para una entidad bancaria de la República Argentina.

El despliegue de funcionalidades consistió en adicionar nuevas funcionalidades, bajo una estrategia modular, estas son:

- Integración con una nueva fuente de datos,
- Publicación de interfaces de programación de aplicaciones (en inglés, Application Programming Interface o API) con portal de capacitación a distancia,
- Modificación a la interface del usuario final,
- Nuevas alertas y notificaciones de gestión de empleados,
- Modificaciones de aspecto al organigrama de la aplicación y
- Modificaciones a los flujos de aprobaciones.

Este estudio de caso es de caso único holístico (Figura 4.1) según la clasificación de Yin (Yin, R., 2014).

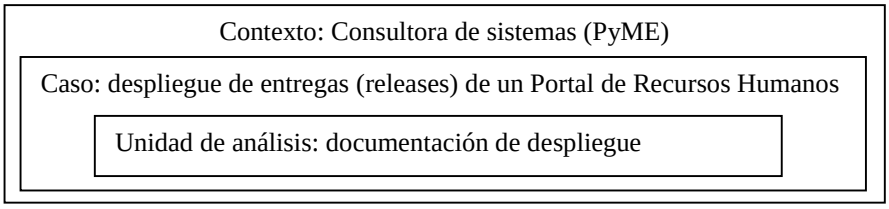


Figura 4.1 Clasificación de estudios de caso basada en la definición de Yin (Yin, R., 2014).

4.1.4. PREPARACION PARA LA RECOLECCIÓN DE DATOS

Para la recolección de los datos de los riesgos del despliegue realizado de nuevas entregas sobre el sistema de gestión, se utilizó una técnica de tercer grado combinada con un método independiente según la clasificación propuesta por Lethbridge et al. (Lethbridge T. et al., 2005). Para recolectar la información sobre los riesgos, se utilizó la codificación propuesta en las secciones 3.1 y 3.4.

En la Tabla 4.1, se presentan presenta la trazabilidad de los documentos analizados para cada actividad del proceso y los riesgos asociados a cada una de las dimensiones (“Proceso”, “Persona” y “Producto”) en el estudio de caso.

Documentos/ Actividades	A1	A2	A3
Planilla de seguimiento de riesgos.	RProc6 RPers3 RProd1	RProc10	RProd15
Informe de avance.	RPers4	RProc7 RProd9	RPers13
Entrega 1 – Informe de cierre.	RProd4	RProc8 RPers9	RProc14 RPers15 RProd13
Entrega 1 – Reporte de despliegue.		RProd8	RProc15 RPers12
Entrega 1 – Resumen de despliegue.		RPers8	RProc11 RProd12
Entrega 1 – Guía de Pruebas del despliegue.	RProc4 RPers2 RProd5	RProd10	
Entrega 1 – Casos de Prueba del despliegue.	RProc4 RPers2 RProd3		
Entrega 1 – Scripts de instalación.	RPers1 RProd2		RProc12
Entrega 1 – Plan de trabajo.	RProc5 RPers5	RProd7 RPers10	
Entrega 1 – Requisitos para el ambiente de instalación.	RProc1 RProd6	RProc9 RPers7	

Documentos/ Actividades	A1	A2	A3
Entrega 1 – Informe de finalización del despliegue.	RProc2 RPers6	RProd9	RProc13 RPers14 RProd14
Entrega 2 – Informe de cierre.	RProd4	RProc8 RPers9	RProc14 RPers15 RProd13
Entrega 2 – Reporte de despliegue.		RProd8	RPers12
Entrega 2 – Resumen de despliegue.		RPers8	RProc11 RProc15 RProd12
Entrega 2 – Guía de Pruebas del despliegue.	RProc4 RPers2 RProd5	RProd10	
Entrega 2 – Casos de Prueba del despliegue.	RProc4 RPers2 RProd3		
Entrega 2 – Scripts de instalación.	RPers1 RProd2		RProc12
Entrega 2 – Plan de trabajo.	RProc5 RPers5	RPers10 RProd7	
Entrega 2 – Requisitos para el ambiente de instalación.	RProc1 RProd6	RProc9 RPers7	
Entrega 2 – Informe de finalización del despliegue.	RProc2 RPers6	RProd9	RProc13 RPers14 RProd14
Documentación General.	RProc3		RPers11 RProd11

Tabla 4.1 Trazabilidad de los documentos analizados para cada actividad del proceso de despliegue.

En la Tabla 4.2 se presenta la ponderación de los riesgos detectados para la dimensión “Proceso” en base al análisis documental.

Actividad	Riesgo	Ponderación	Resultado
A1 Preparación del despliegue.	RProc1	[Probabilidad (A) * Impacto (A)] =	MA
	RProc2	[Probabilidad (M) * Impacto (A)] =	A
	RProc3	[Probabilidad (B) * Impacto (B)] =	B
	RProc4	[Probabilidad (A) * Impacto (MA)] =	MA
	RProc5	[Probabilidad (MA) * Impacto (MA)] =	MA
	RProc6	[Probabilidad (B) * Impacto (A)] =	A
A2 Realización del despliegue.	RProc7	[Probabilidad (B) * Impacto (MA)] =	MA
	RProc8	[Probabilidad (B) * Impacto (M)] =	M
	RProc9	[Probabilidad (M) * Impacto (A)] =	A
	RProc10	[Probabilidad (B) * Impacto (A)] =	A
A3 Gestión de los resultados del despliegue.	RProc11	[Probabilidad (B) * Impacto (M)] =	M
	RProc12	[Probabilidad (A) * Impacto (A)] =	MA
	RProc13	[Probabilidad (A) * Impacto (MA)] =	MA
	RProc14	[Probabilidad (M) * Impacto (A)] =	A
	RProc15	[Probabilidad (B) * Impacto (B)] =	B

Tabla 4.2 Ponderación de los riesgos de la dimensión “Proceso” para el proceso de despliegue.

En la Tabla 4.3 se presenta la ponderación de los riesgos detectados para la dimensión “Producto” en base al análisis documental.

Actividad	Riesgo	Ponderación	Resultado
A1 Preparación del despliegue.	RProd1	[Probabilidad (B) * Impacto (A)] =	A
	RProd2	[Probabilidad (M) * Impacto (A)] =	A
	RProd3	[Probabilidad (M) * Impacto (A)] =	A
	RProd4	[Probabilidad (B) * Impacto (MA)] =	A
	RProd5	[Probabilidad (A) * Impacto (A)] =	MA
	RProd6	[Probabilidad (MA) * Impacto (A)] =	MA
A2 Realización del despliegue.	RProd7	[Probabilidad (M) * Impacto (M)] =	M
	RProd8	[Probabilidad (A) * Impacto (M)] =	A
	RProd9	[Probabilidad (M) * Impacto (A)] =	A
	RProd10	[Probabilidad (A) * Impacto (A)] =	MA
A3 Gestión de los resultados del despliegue.	RProd11	[Probabilidad (A) * Impacto (M)] =	A
	RProd12	[Probabilidad (A) * Impacto (A)] =	MA
	RProd13	[Probabilidad (B) * Impacto (A)] =	A
	RProd14	[Probabilidad (M) * Impacto (A)] =	A
	RProd15	[Probabilidad (A) * Impacto (MA)] =	MA

Tabla 4.3 Ponderación de los riesgos de la dimensión “Producto” para el proceso de despliegue.

En la Tabla 4.4 se presenta la ponderación de los riesgos detectados para la dimensión “Persona”.

Actividad	Riesgo	Ponderación	Resultado
A1 Preparación del despliegue.	RPers1	[Probabilidad (A) * Impacto (A)] =	MA
	RPers2	[Probabilidad (B) * Impacto (A)] =	A
	RPers3	[Probabilidad (M) * Impacto (A)] =	A
	RPers4	[Probabilidad (B) * Impacto (A)] =	A
	RPers5	[Probabilidad (A) * Impacto (MA)] =	MA
	RPers6	[Probabilidad (A) * Impacto (A)] =	MA
A2 Realización del despliegue.	RPers7	[Probabilidad (A) * Impacto (A)] =	MA
	RPers8	[Probabilidad (MB) * Impacto (A)] =	M
	RPers9	[Probabilidad (B) * Impacto (A)] =	A
	RPers10	[Probabilidad (M) * Impacto (A)] =	A
A3 Gestión de los resultados del despliegue.	RPers11	[Probabilidad (B) * Impacto (M)] =	M
	RPers12	[Probabilidad (A) * Impacto (A)] =	MA
	RPers13	[Probabilidad (B) * Impacto (A)] =	A
	RPers14	[Probabilidad (A) * Impacto (A)] =	MA
	RPers15	[Probabilidad (A) * Impacto (MA)] =	MA

Tabla 4.4 Ponderación de los riesgos de la dimensión “Persona” para el proceso de despliegue.

4.1.5. ANÁLISIS E INTERPRETACION DE LOS RESULTADOS

A continuación, se presentan los resultados que dan respuesta a las preguntas de investigación definidas para este estudio de caso:

PI1: ¿Se gestionaron adecuadamente los riesgos durante las actividades del proceso de despliegue de sistemas de software?

En base a la documentación analizada, se logró evidenciar falencias en la gestión de riesgos propuestos para las actividades del proceso de despliegue:

Actividad 1 - Preparación del despliegue: Los informes de avance del despliegue permitieron evidenciar que, debido a las escasas inversiones en tecnología realizadas en los últimos años, los recursos (hardware y software de base) asignados al entorno productivo no cumplían con los requerimientos mínimos solicitados por la consultora para realizar el despliegue de acuerdo con el plan de trabajo establecido.

De acuerdo con los reportes de despliegue analizados, los técnicos (empleados del banco) no disponían de los conocimientos y aptitudes necesarias para el correcto despliegue de scripts y seguimiento de las guías enviadas por la consultora. Esto se debe a que los técnicos que participaron del despliegue original se desvincularon de la organización y fueron reemplazados por personal con poca experiencia tanto técnica como funcional.

De la documentación general del proyecto se desprende que el banco no dispone de una adecuada política de retención de personal lo que genera rotación frecuente.

Actividad 2 - Realización del despliegue: Según los informes de avance del proyecto de despliegue, los inconvenientes técnicos mencionados en la etapa anterior debido a la desvinculación de personal técnico con experiencia en las tecnologías intervinientes y una mayor complejidad del producto generaron fricciones entre la consultora y los directivos del banco debido al incumplimiento de los plazos establecidos en el plan de trabajo. Incluso se llegó a activar una cláusula de penalidad a la consultora.

Durante el análisis documental se evidenciaron planes de prueba incompletos y métricas de despliegue inadecuadas. De acuerdo con los informes de finalización del proyecto de despliegue, la consultora debió afrontar sobrecostos por no contar con procedimientos de gestión documental exigidos por el banco en el contrato y en la política corporativa. Por otro lado, fue necesario sumar recursos técnicos de la consultora para cubrir la falta de pericia técnica de los empleados del banco a quienes de debió capacitar para realizar los futuros despliegues.

Estos inconvenientes técnicos sumados a una agenda de trabajo muy exigente por razones y necesidades internas del banco (evidenciada en los informes de cierre), fueron algunas de las causas que produjeron demoras muy importantes y fricciones entre diferentes sectores de la organización que incluso llegaron a considerar en varias oportunidades la cancelación del proyecto de despliegue.

Actividad 3 - Gestión de los resultados del despliegue: Inconvenientes con los repositorios de software (falta de permisos necesarios, versiones anteriores, falta de componentes, etc.) sumado al bajo compromiso y desconocimiento de los técnicos del banco, generaron múltiples inconvenientes durante el despliegue. Estos inconvenientes técnicos impactaron fuertemente en la calidad del producto final y en la satisfacción de los usuarios quienes vieron afectada su productividad debido a fallas en las funcionalidades de la aplicación una vez finalizado el despliegue.

En los informes de finalización del despliegue, se evidenció además que hubo un dimensionamiento erróneo de los entregables y que no se realizaron las pruebas de seguridad necesarias, esto generó accesos de los usuarios finales a información sensible de recursos humanos.

PI2: ¿De qué manera se puede fortalecer el proceso de despliegue de sistemas software en esta empresa?

Una adecuada gestión de riesgos permite evitar o disponer de procedimientos para la mitigación de los mismos. A continuación, en la Tabla 4.5, se presentan los procedimientos recomendados a la

consultora de sistemas a fin de prevenir, mitigar y/o transferir cada uno de los riesgos asociados las dimensiones “Proceso”. Es importante aclarar que para la construcción de los procedimientos se aplicó el mismo esquema de codificación definido en las secciones 3.1. y 3.4.

Riesgo (RProc)	Procedimiento (PProc)
RProc1 Pocas inversiones en tecnología	PProc1 Las mediciones de software precisas son el mejor método de prevención para este tipo de riesgo. La metodología se basa en gestionar adecuadamente los costos, plazos, y otros factores cuantitativos y cualitativos asociados con los proyectos de despliegue.
RProc2 Fricción entre la gestión de software y los altos ejecutivos.	PProc2 Una vez que se genera fricción entre los ejecutivos principales y la gestión del software, no es fácil continuar con el proyecto correctamente. Algunos de los enfoques para el control introducen cambios radicales, como la externalización de la gestión del software y la reducción de tamaño de entregables durante el despliegue.
RProc3 Nula o inexistente normativa corporativa.	PProc3 Contar con una adecuada política corporativa permite disponer de objetivos claros y sin ambigüedades durante el proyecto de despliegue forzando la utilización de normas y procedimientos.
RProc4 Planes de prueba mal elaborados.	PProc4 Uno de los métodos para prevenir este tipo de riesgo es elaborar el plan de pruebas de despliegue durante la fase de análisis y diseño, para anticipar de esta forma los requerimientos necesarios. La metodología de pruebas de software dependerá de la que se esté utilizando para la gestión del proyecto.
RProc5 Agenda o plan de trabajo reducido.	PProc5 Existen varios métodos de estimación de proyectos de despliegue con el objetivo de mitigar este tipo de riesgos como por ejemplo la utilización de juicio de expertos, la utilización de modelos de estimación, la descomposición del plan de trabajo y la comparación por analogía con otros proyectos similares entre otros.
RProc6 Cancelación del proceso de despliegue.	PProc6 La prevención más efectiva es la planificación y la estimación del proyecto de despliegue. Esto es, metas bien definidas y tareas asignadas de forma adecuada. Se debe mantener además una comunicación fluida entre todos los participantes.
RProc7 Fricción entre el cliente y la empresa proveedora de software.	PProc7 A fin de minimizar la probabilidad de fricción entre clientes y contratistas y las consecuencias que esto puede traer al proyecto de despliegue, es recomendable contar con personal de legales capacitado en el dominio del software, a fin de que pueda ejecutar las cláusulas contractuales de ser necesario.
RProc8 Métricas inadecuadas.	PProc8 Las analogías con el uso de métricas en otros proyectos es uno de los métodos más efectivos para prevenir métricas incorrectas durante el despliegue. Cuanto más importante sea la cantidad de proyectos analógicos (no menor a 25), más efectivo será el resultado.
RProc9 Sobrecostos.	PProc9 A medida que el proyecto avanza es más dificultoso el control de los costos asociados. Los sobrecostos se pueden producir por varios motivos. La mejor forma de mitigación consiste en un seguimiento detallado del proyecto de despliegue. Cualquier exceso de tiempos o recursos utilizados puede generar sobrecostos. En particular la utilización de horas extras de trabajo para el personal puede ser un factor que desencadene el riesgo.

Riesgo (RProc)	Procedimiento (PProc)
RProc10 Planes de capacitación inadecuados.	PProc10 Se deben cubrir con la suficiente antelación, todos y cada uno de los aspectos necesarios de formación y capacitación para todos los integrantes del proyecto de despliegue incluyendo técnicos y usuarios finales. Se debe registrar cada una de las capacitaciones realizadas y evaluar su nivel de cumplimiento de acuerdo con las necesidades del proyecto.
RProc11 Herramientas y métodos de gestión de despliegue inadecuados.	PProc11 El enfoque más efectivo para prevenir la utilización de herramientas de Ingeniería de Software inadecuadas durante el proyecto de despliegue es realizar encuestas y generar métricas de las herramientas más utilizadas por la industria del software.
RProc12 Repositorios inadecuados.	PProc12 Uno de los pasos preventivos más efectivos para un control de configuración inadecuado de los repositorios a utilizar durante el despliegue del producto software es llevar a cabo un análisis completo de todos los tipos de componentes que se fueron producidos, cómo se conectan y con qué frecuencia se actualizan.
RProc13 Dimensionamiento inexacto de los entregables.	PProc13 La metodología de prevención más efectiva para los errores de dimensionamiento es la medición precisa de los tamaños de todos los entregables y los recursos necesarios para producir los mismos. Es recomendable además la utilización de métricas durante el despliegue de los mismos.
RProc14 Baja satisfacción del usuario.	PProc14 La satisfacción del usuario es un tema complejo y multifacético. Algunos de los pasos preventivos que parecen efectivos incluyen especialistas en experiencia de usuario. Además, las encuestas de satisfacción del usuario son el mecanismo de control básico para garantizarla durante el despliegue.
RProc15 Objetivos de mejora ambiguos.	PProc15 El establecimiento de un programa formal de medición de software y la adopción de métricas funcionales son medidas preventivas efectivas para eliminar objetivos ambiguos durante el despliegue de software.

Tabla 4.5 Procedimientos asociados a los riesgos de la dimensión “Proceso” para el estudio de caso 1.

En la Tabla 4.6 se presentan los procedimientos para la dimensión “Persona”.

Riesgo (RPers)	Procedimiento (PPers)
RPers1 Falta de especialización en las tecnologías y procesos involucrados.	PPers1 Un método de prevención y/o mitigación de este tipo de riesgo es crear un inventario de las habilidades de los empleados dentro de la empresa y establecer criterios de especialización y planes de estudio de capacitación de acuerdo con el proyecto de despliegue.
RPers2 Usuarios sin permisos de acceso adecuados.	PPers2 A fin de evitar retrasos e inconvenientes durante el despliegue, se recomienda analizar y solicitar con la suficiente antelación todos permisos de acceso necesarios para todos los integrantes del proyecto de despliegue incluyendo técnicos y usuarios finales de acuerdo con las metodologías establecidas por la Organización.

Riesgo (RPers)	Procedimiento (PPers)
RPers3 Inadecuada política de retención de personal.	PPers3 El enfoque más efectivo para prevenir la baja de recursos durante el proyecto de despliegue es que la organización disponga de una adecuada política de recursos humanos que incluyan incentivos extras con el cumplimiento de hitos de proyecto. No solo en lo económico sino también desde otros aspectos profesionales.
RPers4 Desconocimiento funcional o del negocio por parte de los usuarios a cargo de las pruebas.	PPers4 A fin de minimizar este riesgo, los integrantes del proyecto de despliegue deben ser cuidadosamente seleccionados de acuerdo con su rol dentro de la organización, su compromiso y conocimiento funcional del negocio. Otro aspecto importante es lograr una correcta comunicación entre los integrantes del proyecto.
RPers5 Cambios constantes en las prioridades.	PPers5 Contar con una adecuada dirección y seguimiento del proyecto de despliegue es la mejor forma de prevenir este riesgo. Se deben realizar reuniones de seguimiento frecuentes en los cuales se analiza la viabilidad de los cambios propuestos y el impacto que tienen en los tiempos de proyecto.
RPers6 Esfuerzo y/o recursos adicionales.	PPers6 Disponer de un plan de inducción ágil y estructurado durante el despliegue, reduce los inconvenientes asociados con este riesgo. El nuevo personal debe poder sumarse y asumir sus responsabilidades de forma transparente.
RPers7 Poca experiencia en sistemas actuales.	PPers7 Se deben analizar y documentar todas las plataformas e interfaces que serán parte del proyecto de despliegue y seleccionar a los técnicos idóneos en las mismas a fin de mitigar este riesgo. De ser necesaria capacitación sobre alguna de ellas, se debe realizar con la antelación necesaria.
RPers8 Falta de pericia.	PPers8 Se recomiendan que los recursos asignados al proyecto de despliegue dispongan de conocimiento técnico y del negocio a fin de asegurar el correcto funcionamiento de cada uno de los entregables. Cada una de las tareas debe ser correctamente documentada.
RPers9 Mala práctica profesional.	PPers9 Uno de los pasos preventivos más efectivos para mitigar este riesgo es establecer un plan de despliegue en base a juicio de expertos en cada uno de los componentes involucrados (hardware y servicios de software) a fin de adaptarlos para un correcto despliegue.
RPers10 Baja significativa de recursos asignados al proyecto.	PPers10 Se deben establecer políticas claras dentro de la Organización para que los recursos asignados al proyecto de despliegue no sean reasignados a otras tareas. Del mismo modo, a nivel legal se recomienda establecer con los contratistas la necesidad de mantener la cantidad y nivel técnico de los recursos asignados.
RPers11 Desconocimiento de gestión documental.	PPers11 Una de las mejores prácticas para reducir o mitigar este riesgo es capacitar a los recursos asignados al proyecto de despliegue acerca de las mejores prácticas de la metodología de gestión documental seleccionada para el proyecto de despliegue. En ocasiones es recomendable tercerizar esta tarea en personal especializado.
RPers12 Criterios o interpretaciones diversas.	PPers12 Es recomendable definir un modelo de trazabilidad estándar para todo el proceso de despliegue que incluya a los participantes del proyecto, las fuentes (documentos y modelos) y los objetos o artefactos para ser trazados. Estos elementos y su evolución se deben identificar explícitamente en cada flujo del proyecto despliegue.

Riesgo (RPers)	Procedimiento (PPers)
RPers13 Baja productividad.	PPers13 Un adecuado seguimiento de las tareas asignadas a cada uno de los integrantes del proyecto de despliegue es la mejor manera de minimizar la baja productividad. Se recomienda llevar a adelante una adecuada documentación y reuniones periódicas de seguimiento a fin de resolver desvíos o retrasos.
RPers14 Falta de colaboración de los usuarios finales.	PPers14 Para mitigar este riesgo, es importante que la alta dirección de la organización haga propia y difunda al personal afectado por el proyecto de despliegue la importancia de realizar las pruebas de funcionamiento del producto software a fin de evitar inconvenientes en la operatoria.
RPers15 Bajo compromiso.	PPerso15 La metodología más efectiva es trabajar desde distintos aspectos (técnicos, humanos, etc.) a fin de que todos los integrantes del proyecto de despliegue (cliente y contratistas) sientan el proyecto como propio y desafiante.

Tabla 4.6 Procedimientos asociados a los riesgos de la dimensión “Persona” para el estudio de caso 1.

En la Tabla 4.7 se presentan los procedimientos asociados a la dimensión “Producto”.

Riesgo (RProd)	Procedimiento (PProd)
RProd1 Tecnología novedosa o con poca adopción.	PProd1 Es recomendable utilizar tecnología novedosa, pero con la suficiente madurez y soporte local a fin de evitar inconvenientes durante el proyecto de despliegue. De ser posible, se debería realizar un análisis de proyectos similares para verificar su adaptabilidad a las funcionalidades necesarias.
RProd2 Incompatibilidad con la infraestructura existente.	PProd2 Para mitigar este riesgo, es imprescindible realizar de forma previa un análisis exhaustivo del complimiento de los requerimientos de hardware o software de base necesarios para el despliegue. Todas las tareas deben ser correctamente documentadas y validadas.
RProd3 Falta de adaptación a nuevas tecnologías.	PProd3 Dada el surgimiento de nuevas tecnologías como por ejemplo DevOps y/o despliegue continuo, es necesario adaptan las políticas y/o procedimientos de la organización a la seleccionada para el proyecto de despliegue. De ser necesario, se recomienda sumar consultoría externa para llevar adelante esta tarea de forma adecuada.
RProd4 Falta de componentes.	PProd4 Se debe garantizar que todos los componentes vinculados a los entregables se encuentren disponibles en al momento de realizar el despliegue. La mejor forma de realizar esto es a través de una adecuada trazabilidad de los mismos.
RProd5 Formato de datos incompatible.	PProd5 A fin de prevenir este riesgo, se deben seleccionar conjuntos de datos de cada una de las tecnologías afectadas por el proceso de despliegue a fin de validar la compatibilidad de los mismo durante su importación a la nueva tecnología. Estas pruebas deben ser documentadas y supervisadas.
RProd6 Poca flexibilidad.	PProd6 Definir la estrategia de despliegue de forma clara y concreta permitirá elegir la mejor tecnología para el proyecto de software de modo de que la misma tenga la capacidad de adaptarse a los cambios que puedan surgir durante el despliegue.
RProd7 Mayor complejidad.	PProd7 Se recomienda definir y documentar de forma concreta las funcionalidades y el alcance a cumplir por el producto software a fin de evitar el incremento de costos y tiempo durante el proyecto de despliegue.

Riesgo (RProd)	Procedimiento (PProd)
RProd8 Fallas o Errores en el funcionamiento.	PProd8 Se debe utilizar una metodología para el registro del cumplimiento de todas las funcionalidades del producto durante el despliegue estableciendo revisiones con el objetivo de garantizar que todos los aspectos técnicos y funcionales fueron cubiertos
RProd9 Perdida de características y/o funciones.	PProd9 Es recomendable cumplir con todos los requerimientos de hardware necesario para evitar adaptar el producto debido a incompatibilidad técnica durante el despliegue. Se recomienda realizar un chequeo de forma previa junto con especialistas.
RProd10 Falta de conocimiento de las funcionalidades del producto.	PProd10 Documentar de forma completa todas las funcionalidades del producto software en un manual de usuario final permite aprovechar al máximo sus características y asegurar su correcto despliegue.
RProd11 Pobre documentación.	PProd11 Disponer de una base de conocimientos permite registrar los resultados y las anomalías encontradas durante el despliegue. Las mismas sirve para detectar problemas recurrentes y mejorar el proceso de forma continua.
RProd12 Inconsistencias de versiones del producto.	PProd12 Mantener un control de versiones sobre toda la documentación de análisis y diseño, difundir lo antes posible las últimas versiones y alertar a todo el equipo es una de las mejores formas de prevenir y/o mitigar este riesgo durante el despliegue.
RProd13 Funcionalidades incompletas.	PProd13 Determinar con antelación los parámetros de configuración de componentes de artefactos clave (Por ejemplo: Librerías, parámetros de Shell Scripts, entre otros). Se debe garantizar la completitud de todos los componentes dentro de los repositorios de software durante el despliegue. De forma adicional se deben identificar los requerimientos en las estaciones de trabajo (plugins, componentes active X, etc.).
RProd14 Baja calidad.	PProd14 Revisar de forma exhaustiva y registrar el cumplimiento de todas las funcionalidades del producto software a fin de garantizar la calidad durante el despliegue. Es recomendable utilizar una lista previamente definida junto con los usuarios clave del proyecto.
RProd15 Pruebas de seguridad sin realizar.	Prod15 Se recomienda que durante el proyecto de despliegue se aplique metodologías de ciberseguridad de acuerdo con las mejores prácticas del mercado y los procedimientos definidos por la Organización.

Tabla 4.7 Procedimientos asociados a los riesgos de la dimensión “Producto” para el estudio de caso 1.

4.1.6. AMENAZAS A LA VALIDEZ

Para analizar la validez del estudio, se tuvieron en cuenta los factores propuestos por Lethbridge T. et al. (Lethbridge T. et al, 2005).

- Validez de constructo. Los resultados se obtuvieron en base al análisis documental de un conjunto de riesgos para el proceso de despliegue de sistemas de software en un contexto real, lo que nos permitió responder a las preguntas de investigación definidas, determinando su pertinencia e idoneidad para el caso.

- Validez interna. La documentación utilizada pertenece a un caso real, un despliegue de nuevas entregas de un sistema de gestión de un laboratorio farmacéutico internacional con sede en la República Argentina. Para lograr una mayor precisión y validez del proceso estudiado, se reconoce la necesidad de combinar la fuente de datos (documentación del proyecto) con otro tipo de fuente, como entrevistas y / o grupos focales para garantizar una "triangulación de datos (fuente)". Además, los datos cualitativos recopilados y analizados podrían combinarse con datos cuantitativos resultantes del proyecto, asegurando así una "Triangulación Metodológica".
- Validez externa. El uso de un solo estudio de caso puede limitar la generalización de los resultados. Sin embargo, se realizaron dos estudios preliminares en (Ortiz F. et al., 2019) y (Ortiz F. et al., 2020) y en este caso se considera necesario informar sobre estos hallazgos, ya que sirve como un incentivo para que otros investigadores repliquen la aplicación de los riesgos en diferentes estudios de casos.
- Fiabilidad. Los datos del estudio fueron recopilados por un solo investigador. Aunque fueron analizados con los directores de tesis, esto puede considerarse como una amenaza para la investigación. Para agregar un mayor grado de confiabilidad, sería aconsejable que otro investigador aplique la plantilla con la codificación diseñada en otros estudios de casos.

4.1.7. LECCIONES APRENDIDAS

De este estudio de caso, se obtuvieron las siguientes lecciones aprendidas:

- Selección del método: Se necesitaba una validación de un conjunto de riesgos, así como también de los procedimientos para su prevención, mitigación y/o transferencia para el proceso de despliegue de sistemas de software en un entorno real con el propósito de refinarlos (si fuese necesario). Los resultados obtenidos permitieron analizar la aplicación del conjunto de riesgos definidos en un entorno real, por lo tanto, se considera que el método utilizado ha dado los resultados esperados.
- Datos recolectados. Si bien se ha revisado la documentación del proceso de despliegue de sistemas de software con el propósito de analizar de qué modo se gestionaron los riesgos; se considera que el caso se podría ver fortalecido si los datos recolectados se complementan con otra fuente o con datos cuantitativos.
- Codificación seleccionada. El esquema de codificación seleccionado para el diseño de la plantilla de recolección y análisis de los datos resultó adecuado y permitió de manera sistemática el registro de la información de los riesgos.

- Reporte de resultados. Si bien el caso se compone de dos preguntas de investigación, se considera que el trabajo realizado tuvo en cuenta un nivel de detalle adecuado para la comprensión del fenómeno bajo estudio.

4.1.8. CONCLUSIONES DEL ESTUDIO DE CASO

Se presentaron los resultados de un estudio de caso para determinar la viabilidad de la aplicación de un conjunto de riesgos, así como también de los procedimientos para su prevención, mitigación y/o transferencia para el proceso de despliegue de sistemas de software en un entorno real.

Este consistió en el análisis de riesgos del despliegue de nuevas entregas de un Portal de Recursos Humanos realizado en una entidad bancaria de la República Argentina a cargo de una PyME de sistemas. Después de llevar a cabo el estudio de caso, se concluye que:

- La primera pregunta nos permitió identificar a través del análisis documental falencias dentro de la gestión de riesgos, entre las cuales se puede mencionar la falta de especialización del personal del proyecto, intereses desencontrados entre las áreas intervinientes e incumplimiento de requisitos del ambiente de instalación.
- La segunda pregunta nos permitió diseñar un conjunto de procedimientos recomendados (presentados en la sección 4.1.5) para que la empresa mejore su proceso de despliegue, así como para introducir buenas prácticas de gestión de riesgos para futuros despliegues de sistemas de software.

Las lecciones aprendidas del caso permitieron evidenciar que el método de investigación ha sido acertado para validar la propuesta de los riesgos definidos para el proceso de despliegue de sistemas de software

4.2. ESTUDIO DE CASO 2

4.2.1 DISEÑO DEL ESTUDIO DE CASO

En esta sección, se describe un estudio de caso, siguiendo los lineamientos propuestos de Runeson et al. (Runeson P. et al., 2012). El objetivo principal consiste en examinar la viabilidad de la aplicación del conjunto de riesgos, así como también de los procedimientos para su prevención, mitigación y/o transferencia para el proceso de despliegue de sistemas de software en un entorno real con el propósito de refinarlos (si fuese necesario). Según la clasificación de Robson (Robson C., 2002) se enmarca dentro de los estudios exploratorios. Se trabajó con documentación del despliegue de una entrega

(release) de funcionalidades de un sistema de gestión de un laboratorio farmacéutico internacional con sede en la ciudad de Buenos Aires realizado por una PyME de sistemas de la República Argentina.

4.2.2. PREGUNTAS DE INVESTIGACIÓN

Para alcanzar el objetivo se plantean las siguientes preguntas de investigación (PI):

PI1: ¿Se gestionaron adecuadamente los riesgos durante las actividades del proceso de despliegue de sistemas de software?

A través de esta pregunta, se busca obtener la información de los riesgos que se presentaron en la ejecución del proceso de despliegue y el tratamiento dado por la consultora para compararlos con la propuesta realizada.

PI2: ¿De qué manera se puede fortalecer el proceso de despliegue de sistemas software en esta empresa?

Con esta pregunta se intenta determinar la forma en que la consultora puede fortalecer su proceso de despliegue. Para esto se propone la identificación de un conjunto de riesgos junto con sus procedimientos de prevención, mitigación y/o transferencia.

4.2.3. CASO Y UNIDAD DE ANALISIS

Para el estudio de caso, se recopilamos datos de una consultora de sistemas PyME. Esta consultora se encuentra ubicada en la Ciudad Autónoma de Buenos Aires, cuenta con un staff total de unos 10 empleados y construye sistemas a medida para clientes del rubro farmacéutico utilizando prácticas de metodologías ágiles dado que muchos de sus clientes así lo requieren.

El estudio del caso se centra en el análisis de riesgos del despliegue una entrega (release) de un sistema de gestión financiera de un laboratorio farmacéutico internacional con sede en la República Argentina.

El despliegue de funcionalidades consistió en adicionar nuevas funcionalidades, bajo una estrategia modular, estas son:

- Integración con una nueva forma de pago online
- Modificación de formularios y documentos fiscales existentes para adecuarlos a la normativa y generación de nuevos reportes.

El estudio es de caso único holístico (Figura 4.2) según la clasificación de Yin (Yin, R., 2014).

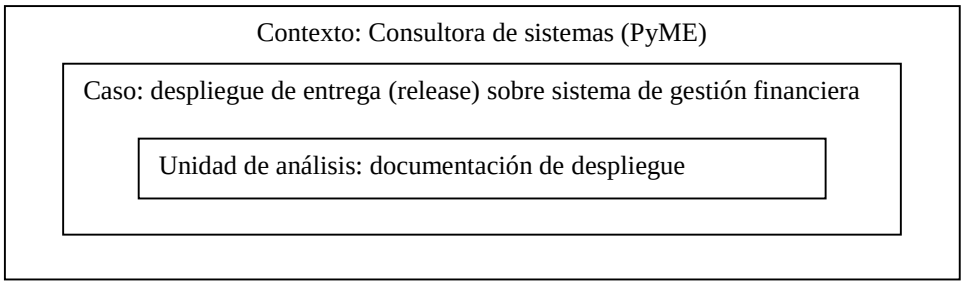


Figura 4.2 Clasificación de estudios de caso 2 basada en la definición de Yin (Yin, R., 2014).

4.2.4. PREPARACION PARA LA RECOLECCIÓN DE DATOS

Para la recolección de los datos de los riesgos del despliegue realizado de esta entrega (release) sobre el sistema de gestión financiera, se utilizó una técnica de tercer grado combinada con un método independiente según la clasificación propuesta en (Lethbridge T. at Al, 2005). Para recolectar la información sobre los riesgos, se utilizó la codificación propuesta en las secciones 3.1 y 3.4.

En la Tabla 4.8, se presentan presenta la trazabilidad de los documentos analizados y los riesgos asociados a cada una de las dimensiones (“Proceso”, “Persona” y “Producto”) en el estudio de caso.

Documentos/ Actividades	A1	A2	A3
Informe de avance.	RProc6	RProc9	RProc14 RPers13 RProd14
Resumen de despliegue.	RProd2	RProc7	RProd11
Guía de Pruebas del despliegue.	RProc4 RPers1	RPers7	
Scripts de instalación.	RPers6	RPers8	RProc12
Plan de trabajo.	RProc1 RPers4	RProc10	RPers12
Requisitos para el ambiente de instalación.	RPers2 RProd4	RProd8	RProd12
Documentación General.	RProd6	RProd10	RProd15

Tabla 4.8 Trazabilidad de los documentos analizados para el estudio de caso 2.

En la Tabla 4.9 se presenta la ponderación de los riesgos detectados para la dimensión “Proceso” en base al análisis documental.

Actividad	Riesgo	Ponderación	Resultado
A1 Preparación del despliegue.	RProc1	[Probabilidad (A) * Impacto (A)]=	MA
	RProc4	[Probabilidad (A) * Impacto (M)]=	A
	RProc6	[Probabilidad (B) * Impacto (M)]=	B
A2 Realización del despliegue.	RProc7	[Probabilidad (M) * Impacto (M)]=	M
	RProc9	[Probabilidad (A) * Impacto (M)]=	A
	RProc10	[Probabilidad (A) * Impacto (A)]=	MA
A3 Gestión de los resultados del despliegue.	RProc12	[Probabilidad (A) * Impacto (MA)]=	MA
	RProc14	[Probabilidad (M) * Impacto (M)]=	M

Tabla 4.9 Ponderación de los riesgos de la dimensión “Proceso” para el estudio de caso 2.

En la Tabla 4.10 se presenta la ponderación de los riesgos detectados para la dimensión “Producto” en base al análisis documental.

Actividad	Riesgo	Ponderación	Resultado
A1 Preparación del despliegue.	RProd2	[Probabilidad (M) * Impacto (M)]=	M
	RProd4	[Probabilidad (A) * Impacto (A)]=	MA
	RProd6	[Probabilidad (M) * Impacto (A)]=	A
A2 Realización del despliegue.	RProd8	[Probabilidad (M) * Impacto (A)]=	A
	RProd10	[Probabilidad (M) * Impacto (M)]=	M
A3 Gestión de los resultados del despliegue.	RProd11	[Probabilidad (A) * Impacto (A)]=	MA
	RProd12	[Probabilidad (A) * Impacto (MA)]=	MA
	RProd14	[Probabilidad (A) * Impacto (M)]=	A
	RProd15	[Probabilidad (A) * Impacto (A)]=	MA

Tabla 4.10 Ponderación de los riesgos de la dimensión “Producto” para el caso de estudio 2.

En la Tabla 4.11 se presenta la ponderación de los riesgos detectados para la dimensión “Persona”

Actividad	Riesgo	Ponderación	Resultado
A1 Preparación del despliegue.	RPers1	[Probabilidad (M) * Impacto (M)]=	M
	RPers2	[Probabilidad (A) * Impacto (M)]=	A
	RPers4	[Probabilidad (A) * Impacto (A)]=	MA
	RPers6	[Probabilidad (MA) * Impacto (MA)]=	MA
A2 Realización del despliegue.	RPers7	[Probabilidad (A) * Impacto (A)]=	MA
	RPers8	[Probabilidad (A) * Impacto (M)]=	A
A3 Gestión de los resultados del despliegue.	RPers12	[Probabilidad (M) * Impacto (A)]=	A
	RPers13	[Probabilidad (A) * Impacto (A)]=	MA

Tabla 4.11 Ponderación de los riesgos de la dimensión “Persona” para el estudio de caso 2.

4.2.5. ANÁLISIS E INTERPRETACION DE LOS RESULTADOS

A continuación, se presentan los resultados que dan respuesta a las preguntas de investigación definidas para este estudio de caso:

PI1: ¿Se gestionaron adecuadamente los riesgos durante las actividades del proceso de despliegue de sistemas de software?

En base a la documentación analizada, se logró evidenciar falencias en la gestión de riesgos propuestos para las actividades del proceso de despliegue:

Actividad 1 - Preparación del despliegue: Los informes de avance permitieron evidenciar que las guías enviadas por la consultora no contenían la totalidad de los pasos necesarios y fue necesaria la intervención de los técnicos de la consultora para asistir a los empleados del laboratorio farmacéutico en varias oportunidades para poder realizar las tareas previas al despliegue.

Por otro lado, el resumen de despliegue evidenció incumplimiento de los requisitos para el ambiente de instalación (Hardware y Software de base) los cuales generaron demoras y sobrecarga del tiempo de los recursos asignados al proyecto de despliegue.

Actividad 2 - Realización del despliegue: Según los informes de avance del proyecto de despliegue, fue necesario sumar recursos varios técnicos adicionales de la consultora (con conocimientos específicos), para analizar inconvenientes que surgieron durante la realización del despliegue y hacer modificaciones sobre algunos de los scripts de instalación. Estos inconvenientes técnicos sumados a la necesidad de comenzar a operar con las nuevas formas de pago de forma inmediata fueron motivo de quejas por parte del cliente.

Actividad 3 - Gestión de los resultados del despliegue: Como se mencionó anteriormente, múltiples inconvenientes con los scripts de instalación sumado a la inexistencia de un plan alternativo (workarround) de las guías de despliegue, generaron demoras durante la actividad de realización del despliegue. Si bien la calidad del producto final fue la esperada, se evidenció insatisfacción parte del cliente quien vio afectada su sistema productivo durante varias horas más de las establecidas en el plan de acción inicial.

PI2: ¿De qué manera se puede fortalecer el proceso de despliegue de sistemas software en esta empresa?

Una adecuada gestión de riesgos permite evitar o disponer de procedimientos para la mitigación de los mismos. A continuación, en la Tabla 4.12, se presentan los procedimientos recomendados a las

consultoras de sistemas a fin de prevenir, mitigar y/o transferir cada uno de los riesgos asociados las dimensiones “Proceso”.

Riesgo (RProc)	Procedimiento (PProc)
RProc1 Pocas inversiones en tecnología.	PProc1Se deben medir adecuadamente los factores cuantitativos y cualitativos asociados con los proyectos de despliegue con el objetivo de cubrir el impacto en la performance del proyecto actual y futuros proyectos.
RProc4 Planes de prueba mal elaborados.	PProc4 Se recomienda involucrar a los usuarios clave para el armado de los planes de pruebas de software a fin de que los mismos aborden el mayor universo de funcionalidades posibles.
RProc6 Cancelación del proceso de despliegue.	PProc6 Para mitigar este riesgo, es clave una adecuada planificación y la estimación del proyecto de despliegue. Se debe mantener además una comunicación fluida entre todos los participantes y definir claramente el alcance y los plazos del mismo.
RProc7 Fricción entre el cliente y la empresa proveedora de software.	PProc7 A fin de minimizar la probabilidad de fricción entre clientes y contratistas y las consecuencias que esto puede traer al proyecto de despliegue, es recomendable contar con personal de legales capacitado en el dominio del software, a fin de que pueda ejecutar las cláusulas contractuales de ser necesario.
RProc9 Sobrecostos.	PProc9 Los sobrecostos se pueden producir por varios motivos. La mejor forma de mitigación consiste en un seguimiento detallado del proyecto de despliegue y analizar correctamente las modificaciones al alcance inicial.
RProc10 Planes de capacitación inadecuados.	PProc10 Se deben cubrir con la suficiente antelación, todos y cada uno de los aspectos necesarios de formación y capacitación para todos los integrantes del proyecto de despliegue incluyendo técnicos y usuarios finales.
RProc12 Repositorios inadecuados.	PProc12 Uno de los pasos preventivos más efectivos para un control de configuración inadecuado de los repositorios a utilizar durante el despliegue del producto software es llevar a cabo un análisis completo de todos los tipos de componentes que se fueron producidos, cómo se conectan y con qué frecuencia se actualizan.
RProc14 Baja satisfacción del usuario.	PProc14 Las encuestas de satisfacción del usuario son el mecanismo de control básico para garantizarla durante el despliegue.

Tabla 4.12 Procedimientos asociados a los riesgos de la dimensión “Proceso” para el estudio de caso 2.

En la Tabla 4.13 se presentan los procedimientos para la dimensión “Persona”.

Riesgo (RPers)	Procedimiento (PPers)
RPers1 Falta de especialización en las tecnologías y proc. involucrados.	PPers1 Un método de prevención y/o mitigación de este tipo de riesgo es crear un inventario de las habilidades de los empleados dentro de la empresa y establecer criterios de especialización y planes de estudio de capacitación de acuerdo con el proyecto de despliegue.
RPers2 Usuarios sin permisos de acceso adecuados.	PPers2 A fin de evitar retrasos e inconvenientes durante el despliegue, se recomienda analizar y solicitar con la suficiente antelación todos permisos de acceso necesarios para todos los integrantes del proyecto de despliegue incluyendo técnicos y usuarios finales de acuerdo con las metodologías establecidas por la Organización.

Riesgo (RPers)	Procedimiento (PPers)
RPers4 Desconocimiento funcional o del negocio por parte de los usuarios a cargo de las pruebas.	PPers4 A fin de minimizar este riesgo, los integrantes del proyecto de despliegue deben ser cuidadosamente seleccionados de acuerdo con su rol dentro de la organización. Los integrantes deben ser aquellos funcionales que conocen adecuadamente el negocio y los técnicos especialistas en cada una de las plataformas involucradas.
RPers6 Esfuerzo y/o recursos adicionales.	PPers6 Disponer de un plan de inducción ágil y estructurado durante el despliegue, reduce los inconvenientes asociados con este riesgo. El nuevo personal debe poder sumarse y asumir sus responsabilidades de forma transparente.
RPers7 Poca experiencia en sistemas actuales.	PPers7 Se deben analizar y documentar todas las plataformas e interfaces que serán parte del proyecto de despliegue y seleccionar a los técnicos idóneos en las mismas a fin de mitigar este riesgo. De ser necesaria capacitación sobre alguna de ellas, se debe realizar con la antelación necesaria.
RPers8 Falta de pericia.	PPers8 Se recomiendan que los recursos asignados al proyecto de despliegue dispongan de conocimiento técnico y del negocio a fin de asegurar el correcto funcionamiento de cada uno de los entregables.
RPers11 Desconocimiento de gestión documental.	PPers11 Para mitigar este riesgo es capacitar a los recursos asignados al proyecto de despliegue acerca de las mejores prácticas de la metodología de gestión documental.
RPers12 Criterios o interpretaciones diversas.	PPers12 Es recomendable definir un modelo de trazabilidad estándar para todo el proceso de despliegue que incluya a los participantes del proyecto, las fuentes (documentos y modelos) y los objetos o artefactos para ser trazados. Estos elementos y su evolución se deben identificar explícitamente en cada flujo del proyecto despliegue.
RPers13 Baja productividad.	PPers13 Se recomienda llevar a adelante una adecuada documentación y reuniones periódicas de seguimiento a fin de resolver desvíos o retrasos.

Tabla 4.13 Procedimientos asociados a los riesgos de la dimensión “Persona” para el estudio de caso 2.

En la Tabla 4.14 se presentan los procedimientos asociados a la dimensión “Producto”

Riesgo (RProd)	Procedimiento (PProd)
RProd2 Incompatibilidad con la infraestructura existente.	PProd2 Para mitigar este riesgo, es imprescindible realizar de forma previa un análisis exhaustivo del complimiento de los requerimientos de hardware o software de base necesarios para el despliegue. Todas las tareas deben ser correctamente documentadas y validadas.
RProd4 Falta de componentes.	PProd4 Se debe garantizar que todos los componentes vinculados a los entregables se encuentren disponibles en al momento de realizar el despliegue. La mejor forma de realizar esto es a través de una adecuada trazabilidad de los mismos.
RProd6 Poca flexibilidad.	PProd6 Definir la estrategia de despliegue de forma clara y concreta permitirá elegir la mejor tecnología para el proyecto de software de modo de que la misma tenga la capacidad de adaptarse a los cambios que puedan surgir durante el despliegue.

Riesgo (RProd)	Procedimiento (PProd)
RProd8 Fallas o Errores en el funcionamiento.	PProd8 Se debe utilizar una metodología para el registro del cumplimiento de todas las funcionalidades del producto durante el despliegue estableciendo revisiones con el objetivo de garantizar que todos los aspectos técnicos y funcionales fueron cubiertos
RProd10 Falta de conocimiento de las funcionalidades del producto.	PProd10 Documentar de forma completa todas las funcionalidades del producto software en un manual de usuario final permite aprovechar al máximo sus características y asegurar su correcto despliegue.
RProd11 Pobre documentación.	PProd11 Disponer de una base de conocimientos permite registrar los resultados y las anomalías encontradas durante el despliegue.
RProd12 Inconsistencias de versiones del producto.	PProd12 Mantener un control de versiones sobre toda la documentación de análisis y diseño, difundir lo antes posible las últimas versiones y alertar a todo el equipo es una de las mejores formas de prevenir y/o mitigar este riesgo durante el despliegue.
RProd14 Baja calidad.	PProd14 Revisar de forma exhaustiva y registrar el cumplimiento de todas las funcionalidades del producto software a fin de garantizar la calidad durante el despliegue.
RProd15 Pruebas de seguridad sin realizar.	Prod15 Se recomienda que durante el proyecto de despliegue se aplique metodologías de ciberseguridad de acuerdo con las mejores prácticas del mercado y los procedimientos definidos por la Organización.

Tabla 4.14 Procedimientos asociados a los riesgos de la dimensión “Producto” para el estudio de caso 2.

4.2.6. AMENAZAS A LA VALIDEZ

Para analizar la validez del estudio, se tuvieron en cuenta los factores propuestos en (Lethbridge T. et al, 2005).

- Validez de constructo. Los resultados se obtuvieron en base al análisis documental de un conjunto de riesgos para el proceso de despliegue de sistemas de software en un contexto real, lo que nos permitió responder a las preguntas de investigación definidas, determinando su pertinencia e idoneidad para el caso.
- Validez interna. La documentación utilizada pertenece a un caso real, un despliegue de nuevas entregas de un sistema de gestión de un laboratorio farmacéutico internacional con sede en la República Argentina. Para lograr una mayor precisión y validez del proceso estudiado, se reconoce la necesidad de combinar la fuente de datos (documentación del proyecto) con otro tipo de fuente, como entrevistas y / o grupos focales para garantizar una "triangulación de datos (fuente)". Además, los datos cualitativos recopilados y analizados podrían combinarse con datos cuantitativos resultantes del proyecto, asegurando así una "Triangulación Metodológica".
- Validez externa. El uso de un solo estudio de caso puede limitar la generalización de los resultados. Sin embargo, se realizaron dos estudios preliminares en (Ortiz F. et al, 2019) y (Ortiz F. et al, 2020) y en este caso se considera necesario informar sobre estos hallazgos,

ya que sirve como un incentivo para que otros investigadores repliquen la aplicación de los riesgos en diferentes estudios de casos.

- Fiabilidad. Los datos del estudio fueron recopilados por un solo investigador. Aunque fueron analizados con los directores de tesis, esto puede considerarse como una amenaza para la investigación. Para agregar un mayor grado de confiabilidad, sería aconsejable que otro investigador aplique la plantilla con la codificación diseñada en otros estudios de casos.

4.2.7. LECCIONES APRENDIDAS

De este estudio de caso, se obtuvieron las siguientes lecciones aprendidas:

- Selección del método: Se necesitaba una validación de un conjunto de riesgos, así como también de los procedimientos para su prevención, mitigación y/o transferencia para el proceso de despliegue de sistemas de software en un entorno real con el propósito de refinarlos (si fuese necesario). Los resultados obtenidos permitieron analizar la aplicación del conjunto de riesgos definidos en un entorno real, por lo tanto, se considera que el método utilizado ha dado los resultados esperados.
- Datos recolectados. Si bien se ha revisado la documentación del proceso de despliegue de sistemas de software con el propósito de analizar de qué modo se gestionaron los riesgos; se considera que el caso se podría ver fortalecido si los datos recolectados se complementan con otra fuente o con datos cuantitativos.
- Codificación seleccionada. El esquema de codificación seleccionado para el diseño de la plantilla de recolección y análisis de los datos resultó adecuado y permitió de manera sistemática el registro de la información de los riesgos.
- Reporte de resultados. Si bien el caso se compone de dos preguntas de investigación, se considera que el trabajo realizado tuvo en cuenta un nivel de detalle adecuado para la comprensión del fenómeno bajo estudio.

4.2.8. CONCLUSIONES DEL CASO DE ESTUDIO

Se presentaron los resultados de un estudio de caso para determinar la viabilidad de la aplicación de un conjunto de riesgos, así como también de los procedimientos para su prevención, mitigación y/o transferencia para el proceso de despliegue de sistemas de software en un entorno real. Este consistió en el análisis de riesgos del despliegue de una entrega (release) sobre el sistema de gestión financiera en un laboratorio farmacéutico internacional con sede en la República Argentina a cargo de una PyME de sistemas. Después de llevar a cabo el estudio de caso, se concluye que:

- La primera pregunta nos permitió identificar a través del análisis documental falencias dentro de la gestión de riesgos, entre las cuales se puede mencionar guías de despliegue incompletas, scripts de instalación con errores y demoras con alto impacto en el proyecto de despliegue.
- La segunda pregunta nos permitió crear un conjunto de procedimientos recomendados (presentados en la sección 4.2.5) para que la empresa mejore su proceso de despliegue, así como para introducir buenas prácticas de gestión de riesgos para futuros despliegues de sistemas de software.

Las lecciones aprendidas del caso permitieron evidenciar que el método de investigación ha sido acertado para validar la propuesta.

6. CONCLUSIONES

Con el objetivo de proponer iniciativas que contribuyan con el desarrollo y mejore la competitividad de las pequeñas y medianas empresas (PyMEs) de Argentina, en este trabajo se ha presentado una propuesta de riesgos, así como también de los procedimientos para su prevención, mitigación y/o transferencia.

Se detallan a continuación los resultados obtenidos en cada uno de los objetivos específicos que, en su conjunto, han permitido alcanzar el objetivo general.

El primer objetivo específico planteaba la construcción del estado del arte sobre metodologías, métodos y estándares que abordan la gestión de riesgos en proyectos de software con foco en el proceso de despliegue. Se ha presentado un método de investigación empírica de Ingeniería de Software, mapeo sistemático de la literatura (SMS). Este permitió sistematizar la evidencia empírica de las metodologías, métodos y estándares que abordan la gestión de riesgo en proyectos de desarrollo de software. Una vez identificadas, se realizó una evaluación basada en características en base a los lineamientos del método DESMET. Se consideraron como características a comparar una visión tridimensional del proceso de despliegue (Proceso-Producto-Persona). Esto permitió identificar de qué manera las metodologías, métodos y estándares que abordan la gestión de riesgos en proyectos de desarrollo de software, soportan el proceso de despliegue de sistemas software.

El segundo objetivo específico planteaba el análisis y diseño de un conjunto de riesgos para el proceso de despliegue, así como también los procedimientos que permitan evitar, mitigar y/o transferir los mismos. Del estado de arte presentado, surgió la vacancia de la gestión de riesgos específicos para el proceso de despliegue de sistemas de software, razón por la cual se propuso fortalecer este proceso mediante la definición de un conjunto de riesgos que permitan la gestión de los mismos para el proceso de despliegue considerando la visión tridimensional del proceso. Para la construcción de la propuesta de riesgos, se contemplaron las actividades y tareas del proceso de transición del estándar ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE, 2017) por ser un estándar reconocido internacionalmente. Para cada una de las actividades y tareas del mencionado estándar y siguiendo con la visión tridimensional del proceso de despliegue “Proceso, Producto y Persona”, se realizó una propuesta de riesgo de según la clasificación de Capers Jones (Jones C., 1994) con adecuaciones al trabajo realizado y a la evolución de la Ingeniería de Software en las últimas décadas. Para la ponderación de los riesgos, se adoptó la propuesta del estándar ISO/IEC 31010:2009 (ISO/IEC/IEEE,

2009) por considerarse una de las principales referencias de gestión de riesgos para la industria de software a nivel internacional.

El tercer y último objetivo específico planteaba validar el conjunto de riesgos propuestos mediante la realización de estudios de casos. Esta propuesta de riesgos, así como también de los procedimientos para su prevención, mitigación y/o transferencia fue aplicada en dos estudios de caso con el objetivo principal de examinar su viabilidad de aplicación para el proceso de despliegue de sistemas de software en un entorno real. En el primer caso, se trabajó con documentación del despliegue de entregas de funcionalidades de un Portal de Recursos Humanos de una entidad bancaria realizado por una PyME de sistemas de la República Argentina y en el segundo caso con documentación del despliegue de entregas de funcionalidades de un sistema de gestión de un laboratorio farmacéutico internacional con sede en la ciudad de Buenos Aires realizado por una PyME de sistemas de la República Argentina.

En ambos casos, se identificaron falencias dentro de la gestión de riesgos y esto permitió crear un conjunto de procedimientos recomendados para que las empresas PyMEs adopten buenas prácticas de gestión de riesgos para futuros despliegues de sistemas de software.

Se puede concluir que en lo que refiere a gestión de riesgos en el proceso de despliegue de software, tanto la propuesta de riesgos como los procedimientos asociados para ambas PyMEs de desarrollo de software de Argentina permitieron confirmar la viabilidad de la propuesta. Esto permite robustecer el proceso de despliegue de sistemas de software para que este tipo de empresas cuenten con procesos definidos y establecidos en mira a lograr una mejor competitividad dentro la industria del software.

7. FUTURAS LÍNEAS DE INVESTIGACIÓN

Una vez definidos los riesgos para el proceso de despliegue de sistemas de software, así como también los procedimientos para para la prevención, mitigación y/o transferencia y validados en dos estudios de casos en PyMEs de desarrollo de software de Argentina a continuación, se detallan líneas de investigación futuras:

- En primer lugar, evolucionar la herramienta presentada para el dimensionamiento de los riesgos, esto se puede lograr mediante el desarrollo de una aplicación web o mobile de modo que las PyMEs puedan obtener de forma práctica y simple un diagnóstico de los riesgos del despliegue.
- Dado que los dos estudios de casos se tratan de despliegues de sistemas de software, en particular en sistemas de información, de esto se desprende una segunda línea de investigación que consiste en escalar la solución a otros tipos de sistemas de software además de incrementar el número de estudios de casos de despliegues de sistemas de información.
- Por último, dado que la solución se encuentre disponible para cualquier PyME de software, una vez que estas la hayan aplicado, recolectar información mediante una encuesta de tipo TAM (Technology Acceptance Model) con el objetivo de analizar el nivel de aceptación y de utilidad de la aplicación del conjunto de riesgos para el proceso de despliegue de sistemas de software.

8. REFERENCIAS

Agile Alliance. Agile Glosarry. https://www.agilealliance.org. Página vigente al 02/11/2020

Agile Business Consortium. DSDM. https://www.agilebusiness.org. Página vigente al 02/11/2020.

Ambler S. The Agile Unified Process. http://www.ambysoft.com/unifiedprocess/agileUP.html Página vigente al 08/04/2020.

Bannerman P. (2008). Risk and risk management in software projects: A reassessment. 81, nro. 12, págs. 2118-2133. Journal of Systems and Software.

Basili V. (1993). The Experimental Paradigm in Software Engineering. Experimental Software Engineering Issues: Critical Assessment and Future Directions. Computer Science, Vol. 706.

Caballero S., Kuna H. (2018). Análisis y gestión de riesgo en proyectos software. XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018), 26 y 27 de abril, Universidad Nacional del Nordeste. ISBN: 978-987-3619-27-4.

Cámara de Software y Servicios Informáticos - CESSI. Anuario de la Industria Argentina de TI 2007/2008, http://www.cessi.org.ar/argentina/anuario_2007-2008.php. Página vigente al 02/12/2020

Cámara de Software y Servicios Informáticos - CESSI. Sector SSI / OPSSI Coyuntura 2019-2020, https://www.cessi.org.ar. Página vigente al 02/12/2020

Charette R. (2005). Why software fails [software failure]. IEEE spectrum, 42(9), 42-49.

CMMI Institute. (2010). Capability Maturity Model Integration. Obtenido de https://cmmiinstitute.com/ Página vigente al 02/11/2020

Dhlamini J., Nhamu I., Kaihepa A. (2009). Intelligent risk management tools for software development. In Proceedings of the 2009 Annual Conference of the Southern African Computer Lecturers' Association (pp. 33-40). ACM.

Díaz J., Perez J., Yague A., Villegas A., de Antona A. (2019) DevOps in Practice – A Preliminary Analysis of Two Multinational Companies. In: Franch X., Männistö T., Martínez-Fernández S. (eds) Product-Focused Software Process Improvement. PROFES 2019. Lecture Notes in Computer Science, vol. 11915. Springer, Cham.

Erich F., Amrit C., Daneva M. (2017) “A Qualitative Study of DevOps Usage in Practice”. Software: Evolution and Process, 29, pp. 1-20.

Forbes J., Baker E. (2003). Improving Hardware, Software, and Training Deployment Processes. In: Proceedings of 19th International Conference on Software Maintenance, pp. 377-380. IEEE, The Netherlands.

Genero M., Cruz-Lemus J., Piattini Velthuis M. (2014). Métodos de investigación en ingeniería del software. Editorial Ra-Ma.

Ianzen A., Mauda E.C., Paludo M.A., Reinehr S., Malucelli A. (2013). Software process improvement in a financial organization: an action research approach. Computer Standard & Interfaces, 36, pp 54–65.

IBM. Rational Software. Péraire C., Edwards M, Fernandes A., Mancin E. y Carroll K. (2007). The IBM Rational Unified Process for Systems.

IEEE ISO/IEC/IEEE 12207:2017(E) (2017). Systems and software engineering — Software life cycle processes.

International Organization for Standardization, «ISO/IEC 31010:2009,» (2009). https://www.iso.org/standard/51073.html. Página vigente al 08/04/2020.

Jansen S., Brinkkemper S. (2006). Definition and validation of the key process of release, delivery and deployment for product software vendors: Turning the ugly duckling into a swan IEEE International Conference on Software Maintenance, ICSM, art. no. 4021334, pp. 166-175.

Jones C. (1994). Assessment and control of software risk, Yourdon Press.

Johnson D. L. (2009). Risk Management and the Small Software Projects. Obtenido de http://www.sei.cmu.edu/iprc/sepg2006/johnson.pdf. Página vigente al 02/10/2019.

Kitchenham B. (2007). Guidelines for Performing Systematic Literature Reviews in Software Engineering. Durham, UK: Software Engineering Group, School of Computer Sciences and Mathematics, Keele University, and Department of Computer Science, University of Durham.

Kimer T. (2006). Software Engineering Techniques: Design for Quality. 227. IFIP International Federation for Information Processing.

Kwak Y. (2004). Project risk management: lessons learned from software development environment. 24, no 11, págs. 915-920. Technovation.

Lethbridge T., Sim S., Singer J. (2005). Studying software engineers: data collection techniques for software field studies. Empir Softw Eng 10(3):311–341.

Liu D., Wang Q., Xiao J. (2009). The role of software process simulation modeling in software risk management: A systematic review. 3rd International Symposium on Empirical Software Engineering and Measurement (págs. 302-311). Empirical Software Engineering and Measurement.

Menezes J., Wanderley M., Gusmão C., Moura H. (2016). Application of Metrics for Risk Management in Environment of Multiple Software Development Projects, pp. 504-511. SCITEPRESS-Science and Technology Publications, Lda.

Ortiz F., Davila M, Panizzi M, Bertone R. (2019). State of the art determination of risk management in the implantation process of computing systems. Congreso Internacional sobre Avances en Nuevas Tendencias y Tecnologías (ICAETT 2019). Ecuador, Guayaquil Ecuador, 29 al 31 de mayo.

PAe, Métrica versión 3 (2001). Portal de Administración Electrónica. Gobierno de España.

Paredes I., Carvalho J. (2017). Research in Progress: Understanding the process of implantation IT Enterprise Applications in Small and Medium Enterprises (SMEs). In Atas da Conferência da Associação Portuguesa de Sistemas de Informação, Vol. 17, No. 17, pp. 270-283.

Portal de administración electrónica. (2012). MAGERIT v.3: Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Obtenido de https://administracionelectronica.gob.es. Página vigente al 11/10/2020

Project Management Institute. (2013). https://www.pmi.org/pmbok-guide-standards. Obtenido de PMBOK® Guide and Standards.

Reascos I., Carvalho J., Bossano S. (2019). Implanting IT Applications in Government Institutions: A Process Model Emerging from a Case Study in a Medium-Sized Municipality. In Proceedings of the 12th International Conference on Theory and Practice of Electronic Governance, pp. 80-85.

Riveros H., Rosas L. (1985). El Método Científico Aplicado a las Ciencias Experimentales. Editorial Trillas. México.

Robson C. (2002). Real world research 2nd edition. Blackwell.

Runeson P., Höst M., Rainer A., Regnell B. (2012). Case study research in software engineering: guidelines and examples. Wiley Publishing, Hoboken.

Sábato J., Mackenzie M. (1982). La Producción de Tecnología. Editorial Nueva Imagen. México.

Software Engineering Institute. (1999). Software Risk Evaluation Method. Obtenido de https://resources.sei.cmu.edu/asset_files/TechnicalReport/1999_005_001_16799.pdf. Página vigente al 11/11/2020

Sommerville I. (2016). Ingeniería de Software. Editado por Pearson Educación. Versión 10.

Subramanian N. (2017). The software deployment process and automation. CrossTalk, 30 (2), pp. 28-34.

Tyndall J. (2012). Building an effective software deployment process. In Proceedings of the 40th annual ACM SIGUCCS conference on User services, pp. 109-114.

Vazquez P., Panizzi M., Bertone R. (2018). Estimación del esfuerzo del proceso de implantación de software basada en el método de puntos de caso de uso. VI Congreso Nacional de Ingeniería Informática / Sistemas de Información (CoNaIISI 2018). Universidad Nacional de Mar del Plata. Del 29 al 30 de noviembre de 2018. ISBN 978-987-4998-15-6.

Wieringa R., Maiden N., Mead N., Rolland C. (2005). Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. Requirements Engineering, 11, pp. 102–107.

Yin, R.K. (2014). Case study research: design and methods. 5th Edition. Sage Publications.

APÉNDICE A. LISTADO DE ESTUDIOS PRIMARIOS UTILIZADOS EN EL SMS.

A continuación, se presentan los artículos primarios utilizados para la realización del SMS:

[EP1] Menezes J. J., Wanderley M., Gusmão C., & Moura H. (2016). Application of Metrics for Risk Management in Environment of Multiple Software Development Projects. (págs. 504-511). SCITEPRESS-Science and Technology Publications, Lda. doi:10.5220/0005859705040511

[EP2] Pasha M., Qaiser G., & Pasha U. (2018). A Critical Analysis of Software Risk Management Techniques in Large Scale Systems. 6, págs. 12412-12424. IEEE ACCESS. doi:10.1109/ACCESS.2018.2805862

[EP3] Bannerman P. L. (2008). Risk and risk management in software projects: A reassessment. 81, no 12, págs. 2118-2133. Journal of Systems and Software. https://doi.org/10.1016/j.jss.2008.03.059

[EP4] Peng Y., Kou G., Wang G., Wang H., & Ko F. I. (2009). Empirical evaluation of classifiers for software risk management. vol. 8, no 04, págs. 749-767. International Journal of Information Technology & Decision Making. doi:https://doi.org/10.1142/S0219622009003715

[EP5] Pérez Moya O., Zulueta Véliz Y. (2013). Proceso para gestionar riesgos en proyectos de desarrollo de software., 7, no 2, págs. 67-82. La Habana, Cuba. Obtenido de http://scielo.sld.cu/scielo.php?pid=S2227-18992013000200009&script=sci_arttext&tlng=en

[EP6] Charalambos L. Iacovou, Nakatsu Robbie (2008). A Risk profile of offshore-outsourced development projects. 51, no 6, págs. 89-94. New York, NY, USA: Communications of the ACM.

[EP7] Islam S., & Dong W. (2008). Human factors in software security risk management. International workshop on Leadership and management in software architecture (págs. 13-16). ACM. doi:10.1145/1373307.1373312

[EP8] Cordero Morales D., Ruiz Constanten Y., Torres Rubio Y. (2013). Sistema de Razonamiento Basado en Casos para la identificación de riesgos de software. 7, no 2, págs. 222-239. La Habana, Cuba. Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992013000200010

[EP9] Gontijo Tavares R., Silva C. & Diniz de Souza A. (2016). Risk management analysis in Scrum software projects. Minas Gerais, Brasil. https://doi.org/10.1111/itor.12401

[EP10] Mon A., Estayno M. G., & López Gil F. (2010). Desarrollo de una propuesta metodológica para la implementación de Sistemas de Tecnologías de la Información. (págs. 479-483). XII Workshop de Investigadores en Ciencias de la Computación. Obtenido de http://hdl.handle.net/10915/19555

[EP11] Bertone R., Thomas P., Taquias D., & Pardo S. (2010). Herramienta para la Gestión de Riesgos en proyectos de software. (págs. 567-576). XVI Congreso Argentino de Ciencias de la Computación. Obtenido de http://hdl.handle.net/10915/19289

[EP12] Caballero, S., & Kuna, H. D. (2018). Análisis y gestión de riesgo en proyectos software. XX Workshop de Investigadores en Ciencias de la Computación.

[EP13] Shareeful I. (2009). Software Development Risk Management Model – A Goal Driven Approach. (págs. 5-8). ACM. doi:10.1145/1595782.1595785

[EP14] Da Silva Lópes J., Braga J., Resende Filho M. (2015). Systems dynamics model for decision support in risk assessment in Systems dynamics model for decision support in risk assessment in software projects., vol. 27, no 12, págs. 976-989. Brazil. https://doi.org/10.1002/smr.1754

[EP15] Zardari S. (2009). Software risk management. (págs. 375-379). International Conference on Information Management and Engineering. doi:10.1109/ICIME.2009.138

[EP16] De Bakker K., Boonstra A., & Wortmann H. (2010). Does risk management contribute to IT project success? A meta-analysis of empirical evidence. 28, no 5, págs. 493-503. International Journal of Project Management. doi:https://doi.org/10.1016/j.ijproman.2009.07.002

[EP17] Nakatsu R. T., & Iacovou C. L. (2009). A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study. 46, no 1, págs. 57-68. Information & Management. doi:https://doi.org/10.1016/j.im.2008.11.005

[EP18] Gefen, D., Wyss, S., & Lichtenstein, Y. (2008). Business familiarity as risk mitigation in software development outsourcing contracts. MIS quarterly, (págs. 531-551). doi:10.2307/25148855

[EP19] Odzaly, E. E., Greer, D., & Sage, P. (2009, October). Software risk management barriers: An empirical study. In 2009 3rd International Symposium on Empirical Software Engineering and Measurement (pp. 418-421). IEEE.

[EP20] Hossain E., Babar M. A., Paik H. Y. & Verner J. (2009). Risk identification and mitigation processes for using scrum in global software development: A conceptual framework. (págs. 457-464). Software Engineering Conference. doi:10.1109/APSEC.2009.56

[EP21] Chowdhury A., & Arefeen, S. (2011). Software risk management: importance and practices. IJCIT, ISSN, 2078-5828.

[EP22] Chowdhury A. & Arefeen, S. (2011). Software risk management: importance and practices. IJCIT, ISSN, 2078-5828.

[EP23] Hu Y., Zhang X., Ngai E. W. T., Cai R. & Liu M. (2013). Software project risk analysis using Bayesian networks with causality constraints. 56, págs. 439-449. Decision Support Systems. https://doi.org/10.1016/j.dss.2012.11.001

[EP24] Xiaosong L., Shushi L., Wenjun C., & Songjiang F. (2009). The application of risk matrix to software project risk management. (págs. 480-483). International Forum on Information Technology and Applications. doi: 10.1109/IFITA.2009.542

[EP25] Hu Y., Du J., Zhang X., Hao X., Ngai, E. W. T., Fan, M., & Liu, M. (2013). An integrative framework for intelligent software project risk planning. 55, no 4, págs. 927-937. Decision Support Systems. https://doi.org/10.1016/j.dss.2012.12.029

[EP26] López C., & Salmerón J. L. (2012). Monitoring software maintenance project risks. 5, págs. 363-368. Procedia Technology. doi:https://doi.org/10.1016/j.protcy.2012.09.040

[EP27] López C., & Salmerón J. L. (2014). Dynamic risks modelling in ERP maintenance projects with FCM. 256, págs. 25-45. Information Sciences. doi:https://doi.org/10.1016/j.ins.2012.05.026

[EP28] Kwan T. W., & Leung H. K. (2011). A risk management methodology for project risk dependencies. 37, no 5, págs. 635-648. IEEE Transactions on Software Engineering. doi:10.1109/TSE.2010.108

[EP29] Hayashi, A., & Kataoka, N. (2008, December). Risk management method using data from EVM in software development projects. In 2008 International Conference on Computational Intelligence for Modelling Control & Automation (pp. 1135-1140). IEEE.

[EP30] Neves S., Da Silva C., Salomón V., Da Silva A., & Sotomonte B. (2014). Risk management in software projects through knowledge management techniques: cases in Brazilian incubated technology-based firms. International Journal of Project Management. 32, 1, International Journal of Project Management. doi.org/10.1016/j.ijproman.2013.02.007

[EP31] Tianyin P. (2011). Development of software project risk management model review. 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), (págs. 2979-2982). doi:10.1109/AIMSEC.2011.6011139

[EP32] Arnuphaptrairong T. (2011). Top ten lists of software project risks: Evidence from the literature survey. International MultiConference of Engineers and Computer Scientists, (págs. 1-6). Obtenido de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.420.3166&rep=rep1&type=pdf

[EP33] Alhawari S., Karadsheh L., Talet A. N. & Mansour, E. (2012). Knowledge-based risk management framework for information technology project. 32, no 1, págs. 50-65. International Journal of Information Management. https://doi.org/10.1016/j.ijinfomgt.2011.07.002

[EP34] Tsigaa Z., Emesa M., Smitha A. (2017). Implementation of a risk management simulation tool. 121, págs. 218-223. CENTERIS - International Conference on Project Management. doi:https://doi.org/10.1016/j.procs.2017.11.030

[EP35] Sanchez H., Robert B., Bourgault M., Pellerin R. (2008). Risk management applied to projects, programs, and portfolios. International journal of managing projects in Business, 2, no 1, págs. 14-35. https://doi.org/10.1108/17538370910930491

[EP36] Kajko-Mattsson M., & Nyfjord J. (2008). State of Software Risk Management Practice. 35, no 4. IAENG international journal of Computer Science. Obtenido de https://www.researchgate.net/profile/Mira_Kajko-Mattsson

[EP37] Wanderley M., Menezes Jr J., Gusmão C., & Lima, F. (2015). Proposal of risk management metrics for multiple project software development. 64, págs. 1001-1009. Procedia Computer Science. https://doi.org/10.1016/j.procs.2015.08.619

[EP38] Kumar C., & Yadav D. K. (2015). A probabilistic software risk assessment and estimation model for software projects. 54, págs. 353-361. Procedia Computer Science. 10.1016/j.procs.2015.06.041

[EP39] Li J., Li M., Wu D., & Song H. (2012). An integrated risk measurement and optimization model for trustworthy software process management.191, págs. 47-60. Information Sciences. doi:https://doi.org/10.1016/j.ins.2011.09.040

[EP40] Shrivastava S. V. & Rathod U. (2017). A risk management framework for distributed agile projects. 85, págs. 1-15. Information and software technology. https://doi.org/10.1016/j.infsof.2016.12.005

[EP41] Fu Y., Li M., & Chen F. (2012). Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix. 30, no 3, págs. 363-373. International Journal of Project Management. https://doi.org/10.1016/j.ijproman.2011.08.004

[EP42] Hu Y., Zhang X., Sun X., Liu M. & Du, J. (2009). An intelligent model for software project risk prediction. (págs. 629-632). Information Management, Innovation Management and Industrial Engineering. doi:10.1109/ICIII.2009.157

[EP43] Sarigiannidis L., & Chatzoglou P. D. (2011). Software development project risk management: A new conceptual framework. 4, no 5, págs. 293-305. JSEA. Obtenido de https://www.researchgate.net/profile/Lazaros_Sarigiannidis/publication/215550449_Software_development_project_risk_management_A_new_conceptual_framework/links/0fcfd4fbf848704d37000000.pdf

[EP44] Sanderson J. (2011). Risk, uncertainty and governance in megaprojects: A critical discussion of alternative explanations. 30, no 4, págs. 432-443. International Journal of Project Management 30. https://doi.org/10.1016/j.ijproman.2011.11.002

[EP45] Marcelino-Sádaba S., Pérez-Ezcurdia A., Lazcano A. M. E., & Villanueva P. (2014). Project risk management methodology for small firms. 32, no 2, págs. 327-340. International journal of project management. doi:https://doi.org/10.1016/j.ijproman.2013.05.009

[EP46] El-Masri M., & Rivard S. (2012). Towards a Design Theory for Software Project Risk Management Systems. Obtenido de https://aisel.aisnet.org/icis2012/proceedings/ResearchInProgress/18/

[EP47] Chawan P. M., Patil J., & Naik R. (2013). Software risk management. 2, no 5, págs. 60-66. International Journal of Computer Science and Mobile Computing. Obtenido de https://s3.amazonaws.com/academia.edu.documents/31212316/V2I5201325.pdf

[EP48] Pfeifer J., Barker K., Ramirez-Marquez J., Morshedlou N. (2013). Quantifying the risk of project delays with a genetic algorithm. 170, págs. 34-44. Int. J. Production Economics. https://doi.org/10.1016/j.ijpe.2015.09.007

[EP49] Venkateshwara M., One Ki (Daniel) Lee (2010). A New Perspective on GDSD Risk Management. Global Software Engineering (ICGSE), (págs. 219-227). doi:10.1109/ICGSE.2010.33

[EP50] Khan Q. & Ghayyur S. (2010). Software Risks and Mitigation in Global Software Development. Journal of Theoretical & Applied Information Technology, 22, no 1. Obtenido de http://jatit.org/volumes/research-papers/Vol22No1/8Vol22No1.pdf

[EP51] De Farias Junior I., De Azevedo R., De Moura H., & Da Silva D. (2012). Elicitation of communication inherent risks in distributed software development. Global Software Engineering Workshops, 37-42. doi: 10.1109/ICGSEW.2012.18

[EP52] Sadiq, M., Rahman, A., Ahmad, S., Asim, M., & Ahmad, J. (2010, May). esrcTool: a tool to estimate the software risk and cost. In 2010 Second International Conference on Computer Research and Development (pp. 886-890). IEEE.

[EP53] Krishnan M. (2015). Software development risk aspects and success frequency on spiral and agile model. International Journal of Innovative research in computer and communication Engineering. doi:10.15680/ijircce.2015.0301024

[EP54] Tao Y. (2008). A study of software development project risk management. (págs. 309-312). Future Information Technology and Management Engineering. doi:10.1109/FITME.2008.125

[EP55] Djemame K., Armstrong D., Kiran M. & Jiang M. (2011). A risk assessment framework and software toolkit for cloud service ecosystems. Cloud Computing, (págs. 119-126). Obtenido de https://s3.amazonaws.com/academia.edu.documents/30913868/cloud_computing_2011_5_20_2015 6.pdf

[EP56] Purdy G. (2010). ISO 31000: 2009—setting a new standard for risk management. 30, no 6, págs. 881-886. Risk Analysis: An International Journal. doi:10.1111/j.1539-6924.2010.01442.x

[EP57] Kenett R., & Raphaeli O. (2008). Multivariate methods in enterprise system implementation, risk management and change management. 9, no 3, pág. 258. International Journal of Risk Assessment and Management. doi=10.1.1.103.9863&rep=rep1&type=pdf

[EP58] Elzamly A. & Hussin B. (2015). Modelling and evaluating software project risks with quantitative analysis techniques in planning software development. 23, no 2, págs. 123-139. Journal of computing and information technology. doi:https://doi.org/10.2498/cit.1002457

[EP59] Fitsilis P. (2008). Comparing PMBOK and Agile Project Management software development processes. Computer and Information Sciences and Engineering, (págs. 378-383). Obtenido de https://link.springer.com/chapter/10.1007/978-1-4020-8741-7_68

[EP60] Rabbi M. F., & Mannan K. O. B. (2008). A review of software risk management for selection of best tools and techniques. CIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing., (págs. 773-778). doi:10.1109/SNPD.2008.127

[EP61] Asnar Y., Giorgini P., & Mylopoulos J. (2011). Goal-driven risk assessment in requirements engineering. 16, no 2, págs. 101-116. Requirements Engineering. https://doi.org/10.1007/s00766-010-0112-x

[EP62] Manalif E., Capretz L. F., Nassif A. B., & Ho D. (2012). Fuzzy-ExCOM software project risk assessment. (págs. 320-325). Machine Learning and Applications. doi:10.1109/ICMLA.2012.193

[EP63] Dehondt II G., Nezlek G. (2009). The cost of risk in offshore systems development. (pág. 718). Obtenido de http://aisel.aisnet.org/amcis2009

[EP64] Laporte C. (2008). A software engineering lifecycle standard for very small enterprises. European Conference on Software Process Improvement, (págs. 129-141). Berlin. doi:https://doi.org/10.1007/978-3-540-85936-9_12

[EP65] Thamhain H. (2013). Managing risks in complex projects. Project management journal, 44, no 2, págs. 20-35. https://doi.org/10.1002/pmj.21325

[EP66] Khan M. A., Khan S., & Sadiq M. (2012). Systematic review of software risk assessment and estimation models. 1, pág. 298. International Journal of Engineering and Advanced Technology. doi=10.1.1.677.830&rep=rep1&type=pdf

[EP67] Elzamly, A., & Hussin, B. (2015). Modelling and evaluating software project risks with quantitative analysis techniques in planning software development. Journal of computing and information technology, 23(2), 123-139.

[EP68] Shahzad B., & Al-Mudimigh A. S. (2010). Risk identification, mitigation and avoidance model for handling software risk. (págs. 191-196). Computational Intelligence, Communication Systems and Networks. doi:10.1109/CICSyN.2010.82

[EP69] Avdoshin, S., & Pesotskaya, E. (2011, October). Software risk management. In 2011 7th Central and Eastern European Software Engineering Conference (CEE-SECR) (pp. 1-6). IEEE.

[EP70] Hauge O., Conradi R., Cruzes D., Sandanger Velle K. (2010). Risks and risk mitigation in open source software adoption: bridging the gap between literature and practice. (2010). International Conference on Open Source Systems, (págs. 105-118). Berlin. https://doi.org/10.1007/978-3-642-13244-5_9

[EP71] Khan S. (2009). An approach to facilitate software risk identification. International Conference on Computer, Control and Communication, (págs. 1-5). doi:10.1109/IC4.2009.4909208

[EP72] Franch Gutiérrez J., Susi A., Annosi M. C., Ayala Martínez C. P., Glott R., Gross D. & Ameller D. (2013). Managing risk in open source software adoption. International Joint Conference on Software Technologies, (págs. 258-264). Obtenido de http://hdl.handle.net/2117/23157

[EP73] Zhou L., Vasconcelos A., & Nunes M. (2008). Supporting decision making in risk management through an evidence-based information systems project risk checklist. 16, no 2, págs. 166-186. Information management & computer security. doi:https://doi.org/10.1108/09685220810879636

[EP74] Muriana C., & Vizzini G. (2017). Project risk management: A deterministic quantitative technique for assessment and mitigation. International Journal of Project Management, (págs. 569-572). doi: 10.1109/IMCEC.2016.7867274

[EP75] Aloini D., Dulmin R., & Mininno V. (2012). Risk assessment in ERP projects. Information Systems.

[EP76] Kwan, T. W., & Leung, H. K. (2011). A risk management methodology for project risk dependencies. IEEE Transactions on Software Engineering, 37(5), 635-648.

[EP77] Nyfjord J., & Kajko-Mattsson M. (2008). Integrating risk management with software development: State of practice. IAENG International Conference on Software Engineering. Obtenido de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.148.7388&rep=rep1&type=pdf

[EP78] Samantra C., Datta S., Mahapatra S. S., & Debata, B. R. (2016). Interpretive structural modelling of critical risk factors in software engineering project. 23, no 1, págs. 2-24. Benchmarking: An International Journal. https://doi.org/10.1108/BIJ-07-2013-0071

[EP79] Singh B., Sharma K. D., & Chandra S. (2012). A new model for software risk management. 3, no 3, págs. 953-956. International Journal of Computer Technology and Applications. Obtenido de https://www.researchgate.net/profile/Bharat_Singh12/publication/254864379_A_New_Model_for_

Software_Risk_Management/links/00b7d51ff7c60a4fcd000000/A-New-Model-for-Software-Risk-Management.pdf

[EP80] Mathkour H. I., Shahzad B., & Al-Wakeel S. (2011). Software risk management and avoidance strategy. 3. International conference on machine learning and computing. Obtenido de https://www.researchgate.net/profile/Basit_Shahzad2/publication/234790514_Software_Risk_Management_and_Avoidance_Strategy/links/58930ad1458515aeac957046/Software-Risk-Management-and-Avoidance-Strategy.pdf

[EP81] Islam, S., & Dong, W. (2008, May). Human factors in software security risk management. In Proceedings of the first international workshop on Leadership and management in software architecture (pp. 13-16).

[EP82] Warkentin M., Moore R., Bekkering E., Johnston A. (2009). Analysis of systems development project risks: An integrative framework. DATABASE for Advances in Information Systems, 40, no 2, págs. 8-27. doi:10.1145/1531817.1531821

[EP83] Sharma S., & Ram B. (2016). Causes of Human Errors in Early Risk assesment in Software Project Management. Second International Conference on Information and Communication Technology for Competitive Strategies, (pág. 11). doi:10.1145/2905055.2905069

[EP84] Knodel J., Naab M., Bouwers E., & Visser J. (2015). Software risk management in practice: Shed light on your software product. Software Analysis, Evolution and Reengineering, (págs. 592-594). doi:10.1109/SANER.2015.7081884

[EP85] Li M., Li J., Song H., & Wu D. (2009). Risk management in the trustworthy software process: a novel risk and trustworthiness measurement model framework. (págs. 214-219). Fifth International Joint Conference. doi:10.1109/NCM.2009.283

[EP86] Salmeron, J. L., & Lopez, C. (2010). A multicriteria approach for risks assessment in ERP maintenance. Journal of systems and software, 83(10), 1941-1953.

[EP87] Pérez Moya, O., & Zulueta Véliz, Y. (2013). Proceso para gestionar riesgos en proyectos de desarrollo de software. Revista Cubana de Ciencias Informáticas, 7(2).

[EP88] Dan T. & Xiao-Hong K. (2008). Risk Management, Centralized EIP System Upgrade Deployment. Knowledge Acquisition and Modeling Workshop, (págs. 150-153). doi:10.1109/KAMW.2008.4810447

[EP89] Khan, S. (2009, February). An approach to facilitate software risk identification. In 2009 2nd International Conference on Computer, Control and Communication (pp. 1-5). IEEE.

[EP90] Feng N. & Xie J. (2009). A method for software project risk identification based on GA. International Conference on Industrial Engineering and Engineering Management, (págs. 618-621). doi:10.1109/ICIEEM.2009.5344515

[EP91] Gandhi R, Link G, Germonprez M. (2018). Open Data Standards for Open Source Software Risk Management Routines: An Examination of SPDX. ACM Conference on Supporting Groupwork, (págs. 219-229). doi:10.1145/3148330.3148333

[EP92] Sun S. (2009). Study on Software Project Risk Priority Management and Framework Based on Information Management System. Information Science and Engineering (págs. 2402-2405). IEEE. doi:10.1109/ICISE.2009.1133

[EP93] Arnold V., Benfordb T., Canadaa J., Sutton S. (2011). The role of strategic enterprise risk management and organizational flexibility in easing new regulatory compliance. International Journal of accounting information systems, 12, no 3, págs. 171-188. https://doi.org/10.1016/j.accinf.2011.02.002

[EP94] Groth K., Zhu D., Mosleh A. (2008). Hybrid methodology and software platform for probabilistic risk assessment. Annual Reliability and Maintainability Symposium. IEEE. doi:10.1109/RAMS.2008.4925831

[EP95] Jun-guang, Z., & Zhen-chao, X. (2010, October). Notice of Retraction: Method study of software project risk management. In 2010 International Conference on Computer Application and System Modeling (ICCASM 2010) (Vol. 8, pp. V8-9). IEEE.

[EP96] Sharma M., Trivedi P., Dubey A., Toshniwal A., & Swarnkar H. (2013). Pioneering an automated risk removal tools in software engineering. Information Systems and Computer Networks, (págs. 104-107). doi:10.1109/ICISCON.2013.6524183

[EP97] Islam, S. (2009, August). Software development risk management model: a goal driven approach. In Proceedings of the doctoral symposium for ESEC/FSE on Doctoral symposium (pp. 5-8).

[EP98] Zhou C., Wang Y., & Huang H. (2016). Sensitivity analysis of software project risk assessment model. Advanced Information Management, Communicates, Electronic and Automation Control Conference, (págs. 569-572). doi:10.1109/IMCEC.2016.7867274

[EP99] Arena M., Arnaboldi M., Azzone G. (2010). The organizational dynamics of Enterprise Risk Management. Organizations and Society, 35, no 7, págs. 659-675. https://doi.org/10.1016/j.aos.2010.07.003

[EP100] Lobato L., Neto P., & Do Carmo Machado I. (2012). A study on risk management for software engineering. doi:10.1049/ic.2012.0006

APÉNDICE B. PRODUCCION CIENTÍFICA

Durante el desarrollo de esta tesis se han divulgado y comunicado resultados parciales de la investigación a través de las siguientes publicaciones:

- Ortiz F, Davila M., Panizzi M. y Bertone R. State of the art determination of risk management in the implantation process of computing systems. En las Actas del I Congreso Internacional sobre Avances en Nuevas Tendencias y Tecnologías (ICAETT 2019). Ecuador, Guayaquil Ecuador, 29 al 31 de mayo, pp- 23-32 (2019). ISNB 978-3-030-32022-5. Este articulo ha sido seleccionado para el libro "ICAETT: The International Conference on Advances in Emerging Trends and Technologies" de la serie Communications in Advances in Emerging Trends and Technologies" (AISC) de la editorial Springer.
- Ortiz F., Panizzi M., y Bertone R. Risk determination for the implantation process of software systems. En las Actas del XXV Congreso Argentino de Ciencias de la Computación - CACIC 2019. Universidad Nacional de Río Cuarto, 14 al 18 de octubre, pp- 817- 825 (2019). ISBN 978-987-688-377-1. Este articulo obtuvo el premio a la mejor exposición del día martes 15 de octubre en el Workshop de Ingeniería de Software (WIS), esta mención se encuentra en: https://cacic2019.exa.unrc.edu.ar/workshops/ingenieria_de_software
- Ortiz F., Panizzi M., y Bertone R. Risk refinement in the deployment process of software systems: a case study. En las Actas del XXVI Congreso Argentino de Ciencias de la Computación - CACIC 2020. Universidad Nacional de La Matanza, 5 al 9 de octubre. ISBN en trámite. Este articulo ha sido seleccionado para el libro "Computer Science – CACIC 2020" de la serie Communications in Computer and Information Science" (CCIS) de la editorial Springer.

Adicionalmente, en el marco de los proyectos de investigación "ESTUDIO DEL PROCESO DE IMPLANTACIÓN DE SISTEMAS INFORMÁTICOS EN EL CONTEXTO INDUSTRIAL DE LA REPÚBLICA ARGENTINA" (SIUTNBA0006576) y "EL IMPACTO DEL FACTOR PEOPLEWARE EN LA IMPLANTACIÓN DE SISTEMAS INFORMATICOS" (EIUTNBA0004347) se lograron las siguientes contribuciones:

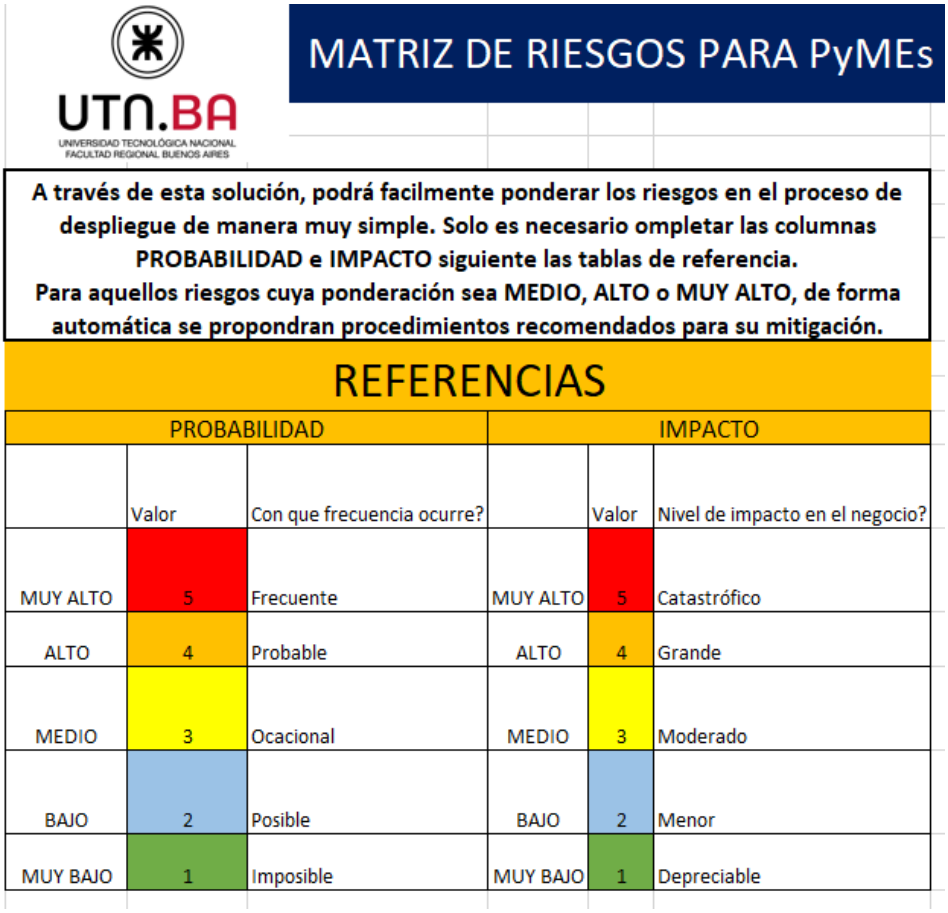
- Panizzi M., Dávila M., Hodes A., Vázquez P., Ortiz F., Bertone R., Hossian A. (2019). APORTACIONES AL PROCESO DE IMPLANTACIÓN DE SISTEMAS INFORMÁTICOS. Workshop. XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019). San Juan, 25 y 26 de abril. ISBN: 978-987-3619-27-4.

- Panizzi M., Davila M., Hodes A., Vázquez P., Ortiz F., Arana F., Bertone R. Desafíos para la implantación de sistemas de software. En las Actas del XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020), El Calafate, Argentina 7 y 8 de mayo de 2020. ISBN en trámite.

APÉNDICE C. HERRAMIENTA PARA EL DIMENSIONAMIENTO DE LOS RIESGOS

A continuación, se presentan algunas imágenes de la herramienta diseñada para el dimensionamiento de los riesgos en el proceso de despliegue de software:

La utilización de esta herramienta es muy simple dado que solo es necesario completar el valor de probabilidad e impacto de cada uno de los riesgos propuestas en base a la escala de referencia para el dimensionamiento de riesgos que se presenta en la Figura C.1



MATRIZ DE RIESGOS PARA PyMEs

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

A través de esta solución, podrá facilmente ponderar los riesgos en el proceso de despliegue de manera muy simple. Solo es necesario ompletar las columnas PROBABILIDAD e IMPACTO siguiente las tablas de referencia.
Para aquellos riesgos cuya ponderación sea MEDIO, ALTO o MUY ALTO, de forma automática se propondran procedimientos recomendados para su mitigación.

REFERENCIAS

PROBABILIDAD			IMPACTO		
	Valor	Con que frecuencia ocurre?		Valor	Nivel de impacto en el negocio?
MUY ALTO	5	Frecuente	MUY ALTO	5	Catastrófico
ALTO	4	Probable	ALTO	4	Grande
MEDIO	3	Ocacional	MEDIO	3	Moderado
BAJO	2	Posible	BAJO	2	Menor
MUY BAJO	1	Imposible	MUY BAJO	1	Depreciable

Figura C.1 Escalas de referencia para el dimensionamiento de riesgos

En base a los valores ingresados, de forma automática la herramienta pondera los riesgos para cada una de las dimensiones y según el valor obtenido, lo clasifica de acuerdo a la escala que se presenta en la Figura C2.

PONDERACION	DESCRIPCIÓN
MUY ALTO	Riesgo que puede afectar muy gravemente al proceso de despliegue. No se debe iniciar el proyecto sin la la mitigación de los mismos.
ALTO	Riesgo que puede afectar de forma importante al proceso de despliegue. Requiere medidas preventivas inmediatas.
MEDIO	Riesgo que puede o no afectar al proceso de despliegue. Se recomienda tener un plan de mitigación.
BAJO	Riesgo cuyo impacto es mínimo en el proceso de despliegue.
MUY BAJO	Riesgo marginal. Se puede asumir durante el proceso de despliegue.

Figura C.2 Escalas de ponderación

Para todos aquellos riesgos cuya ponderación tenga valores Medio, Alto y/o Muy Alto, la herramienta presentará el procedimiento propuesto que permita prevenir, mitigar y/o transferir los inconvenientes que se presentan en el proceso (Figura C.3).

	RIESGOS		PONDERACION			PROCEDIMIENTO RECOMENDADO
	Riesgo	Descripción	Probabilidad	Impacto	Ponderación	Procedimiento recomendado
RIESGOS PARA LA DIMENSION PROCESO (RPROC)	RProc1 Poca inversión en tecnología.	Más allá de la necesidad de cumplir normativas, la pobre inversión en tecnología es endémica a la industria y afecta a empresas de todos los tamaños.	5	3	15	PProc1 Las mediciones de software precisas son el mejor método de prevención para este tipo de riesgo. La metodología se basa en gestionar adecuadamente los costos, plazos, y otros factores cuantitativos y cualitativos asociados con los proyectos de despliegue.
	RProc2 Fricción entre la gestión de software y los altos ejecutivos.	La fricción entre los ejecutivos ocurre en la mayoría de las grandes empresas debido a objetivos y/o necesidades opuestas.	3	5	15	PProc2 Una vez que se genera fricción entre los ejecutivos principales y la gestión del software, no es fácil continuar con el proyecto correctamente. Algunos de los enfoques para el control introducen cambios radicales, como la externalización de la gestión del software y la reducción de tamaño de entregables durante el despliegue.
	RProc3 Nula o inexistente normativa corporativa.	La falta de políticas corporativas genera confusión y reglas poco claras que podrían generar graves inconvenientes durante el Proyecto.	4	2	8	FALSO
	RProc4 Planes de prueba mal elaborados.	La errónea documentación de los planes de prueba puede generar que durante las mismas no se tenga en cuenta todos los casos de uso necesarios.	5	3	15	PProc4 Uno de los métodos para prevenir este tipo de riesgo es elaborar el plan de pruebas de despliegue durante la fase de análisis y diseño, para anticipar de esta forma los requerimientos necesarios. La metodología de pruebas de software dependerá de la que se esté utilizando para la gestión del proyecto.
	RProc5 Agenda o plan de trabajo reducido.	Hitos anticipados o entregables tangibles que ocurren significativamente después de sus fechas planificadas y comprometidas.	4	5	20	PProc5 Existen varios métodos de estimación de proyectos de despliegue con el objetivo de mitigar este tipo de riesgos como por ejemplo la utilización de juicio de expertos, la utilización de modelos de estimación, la descomposición del plan de trabajo y la comparación por analogía con otros proyectos similares entre otros.
	RProc6 Cancelación del proceso de despliegue.	Diversas restricciones o inconvenientes en la estrategia pueden generar la cancelación o suspensión de las actividades de despliegue.	3	5	15	PProc6 La prevención más efectiva es la planificación y la estimación del proyecto de despliegue. Esto es, metas bien definidas y tareas asignadas de forma adecuada. Se debe mantener además una comunicación fluida entre todos los participantes.
	RProc7 Fricción entre el cliente y la empresa proveedora de software.	Enemistad o antagonismo personal que ocurre entre el cliente y los contratistas de software como resultado de malentendidos pueden generar retrasos o incluso la cancelación del proyecto de despliegue.	3	5	15	PProc7 A fin de minimizar la probabilidad de fricción entre clientes y contratistas y las consecuencias que esto puede traer al proyecto de despliegue, es recomendable contar con personal de legales capacitado en el dominio del software, a fin de que pueda ejecutar las cláusulas contractuales de ser necesario.
	RProc8 Métricas inadecuadas.	Métricas de software comunes que violan el estándar o que se comportan de manera paradójica, contra intuitiva o impredecible.	3	3	9	PProc8 Las analogías con el uso de métricas en otros proyectos es uno de los métodos más efectivos para prevenir métricas incorrectas durante el despliegue. Cuanto más importante sea la cantidad de proyectos analógicos (no menor a 25), más efectivo será el resultado.
	RProc9 Sobrecostos.	Las consecuencias de los errores de estimación de costos y recursos suelen ser más graves cuando las estimaciones son bajas y se subestiman los recursos que realmente se necesitan.	4	5	20	PProc9 A medida que el proyecto avanza es más difícultoso el control de los costos asociados. Los sobrecostos se pueden producir por varios motivos. La mejor forma de mitigación consiste en un seguimiento detallado del proyecto de despliegue. Cualquier exceso de tiempos o recursos utilizados puede generar sobrecostos. En particular la utilización de horas extras de trabajo para el personal puede ser un factor que desencadene el riesgo.
	RProc10 Planes de capacitación inadecuados.	Una de las causas más importantes detrás de los fracasos en los despliegues es que algunos empleados no conozcan los beneficios del nuevo sistema.	2	5	10	PProc10 Se deben cubrir con la suficiente antelación, todos y cada uno de los aspectos necesarios de formación y capacitación para todos los integrantes del proyecto de despliegue incluyendo técnicos y usuarios finales. Se debe registrar cada una de las capacitaciones realizadas y evaluar su nivel de cumplimiento de acuerdo con las necesidades del proyecto.
	RProc11 Herramientas y métodos de gestión de despliegue inadecuados.	Métodos automatizados de documentación sumados a enfoques metodológicos inadecuados, podrían generar errores en el registro de los resultados y las anomalías encontradas.	1	3	3	FALSO
	RProc12 Repositorios inadecuados.	Falta de capacidades formales y automatizadas para lidiar con la sincronización, referencias cruzadas, integración y actualización de software.	4	5	20	PProc12 Uno de los pasos preventivos más efectivos para un control de configuración inadecuado de los repositorios a utilizar durante el despliegue del producto software es llevar a cabo un análisis completo de todos los tipos de componentes que se fueron producidos, cómo se conectan y con qué frecuencia se actualizan.
	RProc13 Dimensionamiento inexacto de los entregables.	Fallas en la estimación del tamaño de los componentes principales de software pueden generar inconvenientes en el despliegue de estos.	2	1	2	FALSO
	RProc14 Baja satisfacción del usuario.	Los usuarios pueden no estar satisfechos con la facilidad de uso, capacitación, funcionalidad, calidad y confiabilidad del software entregado. Esto puede derivar en poca o nula utilización de este.	3	5	15	PProc14 La satisfacción del usuario es un tema complejo y multifacético. Algunos de los pasos preventivos que parecen efectivos incluyen especialistas en experiencia de usuario. Además, las encuestas de satisfacción del usuario son el mecanismo de control básico para garantizarla durante el despliegue.
	RProc15 Objetivos de mejora ambiguos.	Objetivos para mejorar la productividad o la calidad del software que son abstractos o tan ambiguos que no hay forma de interpretarlos con precisión.	4	3	12	PProc15 El establecimiento de un programa formal de medición de software y la adopción de métricas funcionales son medidas preventivas efectivas para eliminar objetivos ambiguos durante el despliegue de software.
						PRes1 Un método de prevención y/o mitigación de este tipo de riesgo es crear un inventario de las habilidades de los empleados

Figura C.3 Herramienta de dimensionamiento de riesgos