



*Universidad Tecnológica Nacional
Facultad Regional Buenos Aires*



UTN.BA
ESCUELA DE
POSGRADO

TESIS DE MAESTRÍA

Ingeniería en Sistemas de Información

DISEÑO DE MÉTRICAS PARA EL PROCESO DE DESPLIEGUE DE SISTEMAS DE SOFTWARE

Alumno: Esp. Pablo Daniel Vazquez

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

CABA, 2022

RESUMEN

El despliegue es la fase del ciclo de vida de desarrollo de software en la cual se transfiere el producto software al cliente. Como sucede en otros procesos, el despliegue requiere de planificación para estimar su esfuerzo, costo y además medir diferentes aspectos del proceso para garantizar la culminación del mismo de forma exitosa.

En el despliegue se llevan a cabo actividades en las que se pueden producir problemas, como lo son la falta de componentes, descargas incompletas o despliegues erróneos que puede introducir demoras, ser desorganizado y muy costoso. Los problemas que ocurren en la fase de despliegue se transfieren y eventualmente resuelven como parte de la fase de mantenimiento. Algunas empresas pueden tardar meses y hasta años en lograr finalizar el despliegue de un sistema de software en su totalidad. Por causa de estos problemas, un despliegue de software eficiente y controlado ahorrará considerablemente recursos en términos de costo y esfuerzo.

En la actualidad, para que las empresas mantengan su competitividad en la industria de software estas deben ofrecer productos de calidad, que satisfagan las demandas y exigencias del cliente. Para lograr esto, existen diversos enfoques de desarrollo de software que priorizan la calidad en el proceso y en el producto obtenido. 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 y esto las constituye un elemento fundamental en la economía del país, pero a las mismas se le hace muy difícil de implementar este tipo de enfoques de desarrollo de software debido a que implica una gran inversión de tiempo y recursos.

Dada la relevancia del proceso de despliegue y la necesidad de las PyMES de disponer de procesos estabilizados y controlados, el presente trabajo de tesis tiene como objetivo, aportar a los Ingenieros de Software involucrados en el despliegue de sistemas de software, un conjunto de métricas para dicho proceso. Estas métricas cubren tres aspectos del proceso, el “Proceso” en sí, el cual incluye las actividades y tareas que componen el proceso de despliegue. El segundo aspecto es el “Producto”, que contempla características tales como tamaño, complejidad, características de diseño, rendimiento y nivel de calidad del producto de software. Y el último aspecto es la “Persona”, que involucra tanto a profesionales informáticos como usuarios del sistema.

Las métricas diseñadas para el proceso de despliegue permiten fortalecer el proceso de despliegue de sistemas de software que, por considerarse la última fase del ciclo de vida de desarrollo de software, en muchas ocasiones no se lo toma en consideración con la importancia que amerita.

Por último, se desarrollan dos estudios de caso en los que se aplica el conjunto de métricas diseñadas para el proceso de despliegue de sistemas de software. Estos casos se llevaron a cabo en dos PyMEs de desarrollo de software de Argentina. Estos estudios de casos permitieron evidenciar falencias en el uso de métricas para las actividades y tareas del proceso de despliegue, así como confirmar la viabilidad de las métricas propuestas en esta tesis.

Palabras clave: Métricas, despliegue de sistemas de software, PyMEs de Argentina.

ABSTRACT

Deployment is the phase of the software development life cycle in which the software product is transferred to the customer. As in other processes, the deployment requires planning to estimate its effort, cost and also measure different aspects of the process to guarantee its successful completion.

Deployment involves activities that can cause problems, such as missing components, incomplete downloads, or failed deployments that can be time consuming, disorganized, and very costly. Problems that occur in the deployment phase are transferred and eventually resolved as part of the maintenance phase. Some companies can take months or even years to complete a full software system deployment. Because of these issues, an efficient and controlled software deployment will save considerable resources in terms of cost and effort.

Currently, for companies to maintain their competitiveness in the software industry, they must offer quality products that meet the demands and demands of the client. To achieve this, there are various software development approaches that prioritize quality in the process and in the product obtained. However, in Argentina, the software industry is made up mostly of small and medium-sized enterprises (SMEs), which represent almost 80% of the sector and this constitutes a fundamental element in the country's economy, but they are makes it very difficult to implement this type of software development approach because it involves a large investment of time and resources.

Given the relevance of the deployment process and the need for SMEs to have stabilized and controlled processes, this thesis work aims to provide Software Engineers involved in the deployment of software systems with a set of metrics to said process. These metrics cover three aspects of the process, the "Process" itself, which includes the activities and tasks that make up the deployment process. The second aspect is the "Product", which includes characteristics such as size, complexity, design features, performance and quality level of the software product. And the last aspect is the "Persona", which involves both IT professionals and system users.

The metrics designed for the deployment process make it possible to strengthen the software system deployment process which, because it is considered the last phase of the software development life cycle, is often not taken into consideration with the importance it deserves.

Finally, two case studies are developed in which the set of metrics designed for the software system deployment process is applied. These cases were carried out in two software development SMEs in

Argentina. These case studies made it possible to show shortcomings in the use of metrics for the activities and tasks of the deployment process, as well as to confirm the viability of the metrics proposed in this thesis.

Keywords: Metrics, deployment of software systems, SMEs.

DEDICATORIAS

A mi mujer y mi hijo, por existir en mi vida.

AGRADECIMIENTOS

A mi directora, la Dra. Marisa Daniela Panizzi, por creer en mí, permitirme formar parte de su grupo de investigación, motivarme mientras transitaba este camino y guiarme 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. María Florencia Pollo Cattaneo por su continuo acompañamiento y apoyo.

A mis viejos por haberme dado las herramientas para ser lo que soy hoy.

A mis hermanos por estar siempre.

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN.....	10
1.1. MOTIVACIÓN.....	11
1.2. PLANTEAMIENTO DEL PROBLEMA.....	12
1.3. OBJETIVOS DE LA TESIS.....	16
1.3.1 GENERAL.....	16
1.3.2 ESPECÍFICOS.....	16
1.4. METODOLOGÍA DE INVESTIGACIÓN.....	17
1.4.1. MÉTODOS DE INVESTIGACIÓN.....	17
1.4.1.1. MAPEO SISTEMÁTICO DE LA LITERATURA.....	17
1.4.1.2. PROTOTIPADO EVOLUTIVO EXPERIMENTAL.....	17
1.4.1.3. ESTUDIOS DE CASOS.....	17
1.4.2. ABORDAJE METODOLÓGICO.....	18
1.5. CONTEXTO DE LA INVESTIGACIÓN.....	18
1.6. ESTRUCTURA DE LA TESIS.....	19
CAPÍTULO 2. ESTADO DEL ARTE.....	20
2.1 PLANIFICACIÓN DEL SMS.....	21
2.2. EJECUCIÓN DEL SMS.....	25
2.3. RESULTADOS DEL SMS.....	27
2.3.1. RESPUESTA A LAS PREGUNTAS DE INVESTIGACIÓN.....	27
2.3.2. HALLAZGOS ADICIONALES.....	30
2.4. AMENAZAS A LA VALIDEZ.....	32
2.5. CONCLUSIONES DEL SMS.....	33
CAPÍTULO 3. DEFINICIÓN DE MÉTRICAS.....	35
3.1 ACTIVIDADES Y TAREAS DE LA ISO/IEC/IEEE 12207.....	36
3.2 TIPIFICACIÓN DE LAS MÉTRICAS SEGÚN LA ISO 9126.....	39
3.3 TAXONOMÍA DE LAS MÉTRICAS.....	39
3.4 MÉTRICAS DEFINIDAS PARA EL PROCESO DE DESPLIEGUE.....	40
3.4.1 MÉTRICAS PARA LA DIMENSIÓN “PROCESO”.....	40
3.4.2 MÉTRICAS PARA LA DIMENSIÓN “PRODUCTO”.....	43
3.4.3 MÉTRICAS PARA LA DIMENSIÓN “PERSONA”.....	43
CAPÍTULO 4. VALIDACIÓN DE LA SOLUCIÓN.....	45
4.1. ESTUDIO DE CASO 1.....	46
4.1.1 DISEÑO DEL ESTUDIO DE CASO 1.....	46
4.1.2. PREGUNTAS DE INVESTIGACIÓN.....	46
4.1.3. CASO Y UNIDAD DE ANALISIS.....	47
4.1.4. PREPARACION PARA LA RECOLECCIÓN DE DATOS.....	48
4.1.5. ANÁLISIS E INTERPRETACION DE LOS RESULTADOS.....	49
4.1.6. AMENAZAS A LA VALIDEZ.....	51
4.1.7. LECCIONES APRENDIDAS.....	52
4.1.8. CONCLUSIONES DEL ESTUDIO DE CASO 1.....	52
4.2. ESTUDIO DE CASO 2.....	53
4.2.1 DISEÑO DEL ESTUDIO DE CASO 2.....	53
4.2.2. PREGUNTAS DE INVESTIGACIÓN.....	53
4.2.3. CASO Y UNIDAD DE ANALISIS.....	54
4.2.4. PREPARACION PARA LA RECOLECCIÓN DE DATOS.....	55
4.2.5. ANÁLISIS E INTERPRETACION DE LOS RESULTADOS.....	56
4.2.6. AMENAZAS A LA VALIDEZ.....	58

4.2.7. LECCIONES APRENDIDAS	59
4.2.8. CONCLUSIONES DEL ESTUDIO DE CASO 2	59
CAPÍTULO 5. CONCLUSIONES Y TRABAJOS FUTUROS	61
5.1. CONTRIBUCIONES	62
5.2. CONSECUCCIÓN DE LOS OBJETIVOS DE LA TESIS.....	63
5.3. TRABAJOS FUTUROS.....	65
5.4. PUBLICACIONES	65
5.4.1. CONGRESOS NACIONALES	65
5.4.2. CONGRESOS INTERNACIONALES.....	66
5.4.3. REVISTA.....	66
6. BIBLIOGRAFÍA.....	68

ÍNDICE DE FIGURAS

Figura 2.1. Cantidad de artículos por año.	30
Figura 2.2. Cantidad de artículos por tipo de publicación.	30
Figura 3.1. Procesos del estándar ISO/IEC/IEEE 12207:2017.	36
Figura 4.1. Clasificación de estudios de caso basada en la definición de Yin.	48
Figura 4.2. Clasificación de estudios de caso basada en la definición de Yin.	55

ÍNDICE DE TABLAS

Tabla 1.1 Clasificación de PyMEs.	11
Tabla 2.1 Preguntas de investigación (PI) y motivación (MO).	21
Tabla 2.2 Bibliotecas y repositorios digitales utilizadas.	22
Tabla 2.3 Cadena de búsqueda piloto.	22
Tabla 2.4 Cadenas de búsqueda definitiva.	23
Tabla 2.5 Criterios de inclusión y exclusión.	24
Tabla 2.6 Esquema de clasificación de estudios primarios.	25
Tabla 2.7 Formulario de extracción de datos.	25
Tabla 2.8 Listado de estudios primarios.	26
Tabla 2.9 Resultados por preguntas de investigación (PIs).	27
Tabla 2.10 Estudios primarios por fuente de búsqueda, título y tipo de publicación, y año.	31
Tabla 2.11 Síntesis por país.	31
Tabla 2.12 Síntesis por contexto.	32
Tabla 3.1 Actividades del “Proceso de Transición”.	38
Tabla 3.2 Métricas de la Dimensión “Proceso” para cada una de las actividades.	40
Tabla 3.3 Descripción de las Métricas de la Dimensión “Proceso”.	41
Tabla 3.4 Métricas de la Dimensión “Producto” para cada una de las actividades.	43
Tabla 3.5 Descripción de las Métricas de la Dimensión “Producto”.	43
Tabla 3.6 Métricas de la Dimensión “Persona” para cada una de las actividades.	44
Tabla 3.7 Descripción de las Métricas de la Dimensión “Persona”.	44
Tabla 4.1. Resumen de los documentos analizados para el estudio de caso y métricas.	48
Tabla 4.2 Resumen de los documentos analizados para el estudio de caso y métricas.	55

CAPÍTULO 1. INTRODUCCIÓN

El presente capítulo presenta en la sección 1.1. lo que nos motivó a llevar a cabo esta tesis de Maestría. A continuación, en la sección 1.2, se presenta la problemática actual respecto al proceso de despliegue de sistemas de software. En la sección 1.3 se presenta el objetivo de la tesis. La metodología de investigación aplicada para el desarrollado de esta tesis de Maestría se detalla en la sección 1.4. En la sección 1.45 se describe el contexto sobre el cual se llevó a cabo la tesis. Por último, para finalizar este capítulo, en la sección 1.6, se presenta la estructura del resto del documento.

1.1. MOTIVACIÓN

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.

En este trabajo de tesis se utilizará la clasificación de PyMEs según la cantidad de empleados. La última clasificación de PyMEs que ha sido propuesta por la Secretaría de Emprendedores y PyMEs del Ministerio de Desarrollo Productivo (Ministerio de Desarrollo Productivo, 2018) de Argentina, se presenta en la Tabla 1.1.

Tabla 1.1 Clasificación de PyMEs.

Categoría	Actividad				
	Construcción	Servicios	Comercio	Industria y minería	Agropecuario
Micro	12	7	7	15	5
Pequeña	45	30	35	60	10
Mediana tramo 1	200	165	125	235	50
Mediana tramo 2	590	535	345	655	215

Las PyMES para subsistir y crecer, necesitan soluciones de Ingeniería de Software eficientes y efectivas. Pero la implementación adecuada de las técnicas y buenas prácticas de Ingeniería de Software es una tarea difícil para las PyMES, ya que a menudo operan con recursos limitados y con estrictas limitaciones de tiempo (Mishra & Mishra, 2009).

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 y es un proceso crucial, porque su resultado determina la aceptación exitosa o no del sistema de software por parte del cliente (Subramanian, 2017). En consecuencia, dada la relevancia del proceso de despliegue y la necesidad de las PyMES de disponer de procesos estabilizados y controlados para la entrega de sistemas de software que satisfagan las necesidades de sus clientes, se plantean un conjunto de métricas que permitan mejorar la calidad del proceso de despliegue de sistemas de software.

En la actualidad, en el contexto de las metodologías ágiles existen prácticas que permiten transferir el producto software desde el desarrollo a operación, tales como DevOps (Desarrollo y Operaciones, su acrónimo en inglés de *Development and Operations*) y el Despliegue Continuo (en inglés, *Continuous Deployment*). DevOps es un 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, 2017). Y 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 (Díaz *et al.*, 2019), que cuentan con recursos económicos y tecnológicos para poder aplicarlas. Por este motivo, se realiza la propuesta de métricas para el proceso de despliegue de sistemas de software orientada a las PyMES de Argentina que requieren fortalecer sus procesos de software con el propósito de ser competitivas a pesar de contar con recursos tecnológicos y recursos humanos insuficientes (Ianzen *et al.*, 2013).

1.2. PLANTEAMIENTO DEL PROBLEMA

La necesidad de las PyMES por sistematizar y estabilizar sus procesos de software para sostener su competitividad en el sector de software llevó a centrar la investigación en la construcción de un conjunto de métricas para el proceso de despliegue de sistemas de software.

Como se mencionó anteriormente, el despliegue de sistemas de software tiene importancia en el ciclo de vida de desarrollo del software porque se logra la aceptación por parte del cliente y la culminación del proyecto de software.

El proceso de despliegue contiene prácticas que generan problemas, como lo son la falta de componentes, descargas incompletas o despliegues erróneos (Jansen *et al.*, 2006), que puede llevar mucho tiempo, ser desorganizada y ser muy costosa (Tyndall, 2012). Los problemas que ocurren 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. Por causa de estos problemas, un despliegue de software eficiente ahorrará considerablemente recursos en términos de costo y esfuerzo (Subramanian, 2017).

Otras de las cuestiones con las que se debe lidiar en el proceso de despliegue, es la limitación de recursos tecnológicos, en particular su distribución, heterogeneidad y dinámica. En algunas ocasiones, se debe reaccionar ante fallas y variaciones en la calidad de los recursos sin olvidar satisfacer los requisitos de los interesados. Por lo tanto, el despliegue debe ser un proceso adaptable, pero sin perder el foco de satisfacer los requisitos del proyecto (Arcangeli *et al.*, 2015).

Yim menciona que muchas de las fallas que se producen durante el proceso de despliegue son causadas por errores humanos. Estas las clasifica según su origen en cinco tipos: nueva información, supervisión, comunicación, educación y desatención. Y el grado en que cada una de estas afectan al proceso de despliegue, también se ve afectado en gran medida por las personas, el procedimiento y la infraestructura de cada equipo de ingeniería (Yim, 2014).

En un gran número de empresas este proceso se lleva a cabo de manera informal y la falta de precisión provoca fallas durante el funcionamiento del sistema que, según la gravedad de las mismas, llevan a la eliminación del software (Carrizo *et al.*, 2017). Estas dificultades se relacionan con el entorno en el que se distribuyen o simplemente debido a la cultura organizacional (Cyrus *et al.*, 2014).

Reascos *et al.* (Reascos *et al.*, año) consideran tres contextos para reconocer las causas de la alta tasa de fallas en los proyectos de despliegue de aplicaciones de software en pequeñas y medianas empresas: tecnológico, organizacional y medioambiental. Entre las principales causas tecnológicas se encuentran: la infraestructura heterogénea e incompatible, las pocas capacidades y competencias tecnológicas de las PyME, la complejidad de estos sistemas, su ajuste y personalización en la empresa y la mala calidad y seguridad de los datos. En cuanto a las causas organizativas se mencionan: el liderazgo deficiente, planificación estratégica baja, los costos directos e indirectos son poco

estimados, los errores en las etapas iniciales escalan a lo siguiente, una deficiencia en la estructura de la organización y los procesos informales, falta de recursos necesarios, bajos niveles de gestión (proyectos, cambio, riesgo), la selección del paquete de software y la implantación son difíciles, 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: cambiar las regulaciones gubernamentales, las constantes presiones del mercado, la dificultad de acceder a paquetes de software y consultores que se ajustan a la organización (Paredes *et al.*, 2017).

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 *et al.*, 2019).

La creciente demanda de productos software de calidad, exige a las organizaciones que desarrollan software, el uso de estándares y metodologías que aseguren la entrega de dichos productos software con la calidad requerida. La obtención de software de calidad implica realizar controles y mediciones de manera constante durante todas las fases de su desarrollo, con el propósito de conocer su nivel de calidad en cada momento. En este contexto, las métricas ayudan a entender tanto al proceso que se utiliza para desarrollar un producto software, como el propio producto. Las métricas, cumplen un rol importante en la obtención de un producto software de alta calidad. (Basso, 2014).

En la Ingeniería de Software, existen diversos enfoques de desarrollo de software que priorizan la calidad en el proceso y en el producto obtenido. Estos modelos de calidad presentan factores y métricas que describen las características del software y sus relaciones, las cuales nos permiten identificar fortalezas y debilidades, así como ayudar a controlar y reducir causas significativas de baja calidad, costo o rendimiento (López *et al.*, 2016).

En las últimas décadas, las métricas del software y su proceso de medición asociado han captado la atención de la comunidad de Ingeniería de Software como medio para cuantificar y controlar la calidad del software. De esto resulta importante resaltar en el desarrollo de software, que la calidad no solo se ve reflejada en actividades o tareas. Además, la forma de trabajo del equipo (Carrizo D. *et al.*, 2018), el uso de modelos de calidad y metodologías adecuadas que permitan llevar el control de todas las fases del proceso, buscan reducir costos, evitar retrasos en la planificación, mejorar el tiempo de dedicación y aumentar la calidad en el desarrollo de software, permitiendo mejorar la competitividad de las organizaciones (Chacón Luna *et al.*, 2015).

Algunos autores mencionan que, la calidad es una meta genérica y necesaria que debe buscar la Ingeniería de Software, donde las métricas son una herramienta primordial. También contribuyen a perfeccionar, cuantificar y pronosticar los diferentes comportamientos y controles a los procesos de diseño, desarrollo e implementación de sistemas de software, y a una posterior evaluación del nivel de satisfacción del usuario. Esto permite que el producto software pueda medirse con el fin de evaluar variables como la productividad, la calidad, entre otros factores (Basso, 2014; Morillas *et al.*, 2007; Ruiz *et al.*, 2006).

Otros autores, expresan que para poder evaluar la calidad de un producto software se requiere de un modelo de calidad, las métricas que se van a utilizar, el proceso de evaluación y las herramientas de soporte que automaticen todo lo posible el proceso de desarrollo del software (Rodríguez *et al.*, 2014).

Por otro lado, otros autores entienden que, la calidad de software se puede medir en términos de densidad de defectos (Fenton *et al.*, 2014). Este concepto lo reafirman Fernández *et al.* (Fernández *et al.*, 2015). F. y Ángel M., cuando definen que la calidad de los sistemas software depende en gran medida, del resultado obtenido de las pruebas de software que hayan sido realizadas durante su desarrollo. Por ello, las empresas productoras de software deben establecer los mecanismos de control que permitan determinar que su producto software cumple con normas, procesos y estándares de calidad, que garanticen que su producto se encuentra libre de errores (Puello *et al.*, 2016).

Por su parte, Pressman. (Pressman, 2015) se refiere a la calidad del software como “la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”. También hace mención acerca de las métricas expresando que son escalas de unidades sobre las cuales puede medirse un atributo cuantificable.

Actualmente, las organizaciones de desarrollo y los softwares que son creados diariamente demandan la incorporación de normas que certifiquen un estimado de calidad. De esta forma, se logra que los productos y procesos que se llevan a cabo en ellas estén vinculados estrechamente, ya que el proceso que se toma en la realización del software permite un resultado de calidad (Redrován Castillo *et al.*, 2017).

Es muy importante comprender cómo y especialmente por qué las métricas son efectivas. Estas métricas son una parte fundamental en los modelos de predicción, los cuales nos ayudan a garantizar la calidad de los productos (Rahman *et al.*, 2013).

Las métricas pueden ser utilizadas en todas las fases del ciclo de vida del software, debido a que, con ellas, se puede alcanzar la eficacia del proyecto, la optimización del producto y la calidad de este. Para hacer un uso adecuado de las métricas, no es suficiente con medir los atributos cuantitativamente, sino que es necesario tener en cuenta consideraciones como las unidades que se aplican, el tipo de software al que es aplicable y las condiciones en que se deben recoger los datos (López *et al.*, 2016).

Eisty *et al.* (Eisty *et al.*, 2018). en su investigación concluyen que la comunidad de desarrollo de software tiene un conocimiento general de las métricas. Sin embargo, su conocimiento de métricas específicas es deficiente y su uso aún más limitado. También obtuvieron como resultado de su trabajo, que las métricas más utilizadas están relacionadas con el rendimiento y las pruebas.

Si bien contamos con metodologías, estándares y prácticas que custodian el proceso de despliegue, la calidad del proceso se ve afectada por una serie de dificultades que se producen durante su ejecución. Las PyMEs para ser competitivas en el sector del software, requieren de procesos estables y controlados y como se mencionó anteriormente, en este trabajo de tesis se busca fortalecer el proceso de despliegue de sistemas de software y para esto se diseña un conjunto de métricas para el proceso de despliegue de sistemas de software que permita a las PyMEs incrementar la calidad de dicho proceso.

1.3. OBJETIVOS DE LA TESIS

1.3.1 GENERAL

El objetivo general de la presente tesis consiste en el:

Diseño de un conjunto de métricas que permitan robustecer el proceso de despliegue de sistemas de software para PyMEs de la República Argentina.

1.3.2 ESPECÍFICOS

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

- O1. Construir el estado del arte sobre métricas para el proceso de despliegue.
- O2. Diseñar un esquema de clasificación para las métricas a construir.
- O3. Construir un conjunto de métricas para el proceso de despliegue de sistemas de software.
- O4. Validar el conjunto de métricas en diferentes estudios de casos.

1.4. METODOLOGÍA DE INVESTIGACIÓN

Para el desarrollo de la investigación de la tesis de Maestría, se siguió un enfoque de investigación clásico (Riveros *et al.*, 1985; Creswell., 2002) con énfasis en la producción de tecnologías (Sábato *et al.*, 1982); identificando los métodos y materiales necesarios para desarrollar el proyecto.

1.4.1. MÉTODOS DE INVESTIGACIÓN

A continuación, se describen los métodos de investigación que se utilizarán en el desarrollo de la tesis.

1.4.1.1. MAPEO SISTEMÁTICO DE LA LITERATURA

Los mapeos sistemáticos de la literatura (en inglés, *Systematic Literature Mapping* o SMS) son estudios secundarios cuyo principal objetivo es 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. Esto permite identificar temas en los que la evidencia empírica sea escasa y sea necesario realizar más estudios empíricos (Genero *et al.*, 2014).

1.4.1.2. PROTOTIPADO EVOLUTIVO EXPERIMENTAL

El prototipado evolutivo experimental (Basili, 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.

1.4.1.3. ESTUDIOS DE CASOS

Un estudio de caso en ingeniería del software es una investigación empírica que hace uso de múltiples evidencias para investigar una instancia (o un pequeño número de instancias) de un fenómeno contemporáneo relacionado con la ingeniería del software dentro de su contexto real, específicamente cuando las fronteras entre el fenómeno y su contexto no pueden definirse claramente (Runeson *et al.*, 2012).

Los estudios de casos no dan lugar, como resultado, a relaciones causales como ocurre con los experimentos, sino que permiten comprender más en profundidad el fenómeno que se está estudiando en su contexto real. Precisamente, ahí radica la principal funcionalidad de los estudios de casos, en la

capacidad de proporcionar resultados de investigación a partir de proyectos del mundo real (Genero *et al.*, 2014).

Los estudios de casos se caracterizan por:

- 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 aplican de la siguiente manera:

- Para el O1 se desarrolla el SMS para la construcción del estado del arte con el propósito de descubrir cotejar y sistematizar la evidencia empírica para identificar (determinar) el uso de métricas en el proceso de despliegue de sistemas de software.
- Para el O2 se define la clasificación de las métricas aplicable al proceso de despliegue.
- Para el O3 se diseña un conjunto de métricas para el proceso de despliegue, mediante el abordaje del prototipado evolutivo experimental.
- Finalmente, para el O4 se validó el conjunto de métricas para el proceso de despliegue de sistemas de software mediante 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 enmarca en los proyectos de investigación titulado 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 Informáticos” (EIUTNBA0004347). Ambos proyectos, a cargo de la Dra. Marisa Daniela Panizzi y con el asesoramiento científico y tecnológico del Mg. Rodolfo Alfredo Bertone del Grupo de investigación

en Ingeniería de Software el Instituto de Investigación en Informática (III-LIDI) de la Facultad de Informática de la Universidad Nacional de La Plata.

1.6. ESTRUCTURA DE LA TESIS

El resto del documento se estructura como se detalla a continuación:

- **Capítulo 2: Estado del arte.** En el segundo capítulo, se presenta la construcción del estado del arte mediante el desarrollo del mapeo sistemático de literatura con el propósito de determinar el uso de métricas para el proceso de despliegue de sistemas de software. Por último, se describen las conclusiones del estado del arte.
- **Capítulo 3: Definición de las métricas.** Presenta el desarrollo de la definición de un conjunto de métricas para el proceso de despliegue de sistemas de software para PyMEs de Argentina. Se explica la visión tridimensional del proceso “Proceso, Producto, Persona”, la clasificación considerada, así como también las actividades y tareas de los estándares en los cuales se basa la propuesta de métricas.
- **Capítulo 4: Validación de la solución.** En cuarto capítulo se describen 2 estudios de casos con el propósito de examinar la viabilidad de la aplicación del conjunto de métricas en PyMEs desarrolladoras de software de Argentina. El primer estudio de caso se trata del despliegue de un Portal de Recursos Humanos realizado en una entidad bancaria de la República Argentina. El segundo estudio consiste en el despliegue de un Sistema de Gestión para una agencia publicitaria de Argentina, también realizado por una PyME de desarrollo de software.
- **Capítulo 5: Conclusiones, trabajos futuros y publicaciones.** En quinto capítulo se detallan las principales contribuciones de la tesis, las líneas de trabajo que se pueden seguir a futuro como continuación de esta tesis y las publicaciones obtenidas.
- **Bibliografía.** En esta sección se listan las referencias bibliográficas utilizadas en el desarrollo de esta tesis.

CAPÍTULO 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) el uso de métricas en el despliegue de sistemas de software.

El desarrollo del SMS se encuentra organizado de la siguiente manera: la sección 2.1 describe la actividad de planificación, la sección 2.2 detalla la ejecución de la revisión, en la sección 2.3 se presentan los resultados del SMS, en la sección 2.4 se exponen las amenazas a la validez del SMS y finalmente en la sección 2.5 se presenta la conclusión en base a los resultados obtenidos.

2.1 PLANIFICACIÓN DEL SMS

El SMS se desarrolla de acuerdo con las directrices propuestas por Kitchenham *et al.* (2015) y por Petersen *et al.* (2008).

En esta sección se presenta la definición del protocolo de revisión que se compone de: las preguntas de investigación (PI), la estrategia de búsqueda, los criterios y proceso de selección de los estudios, la estrategia de extracción de datos y el proceso de síntesis.

El objetivo de este SMS es responder la siguiente pregunta de investigación (PI): *¿Se hace uso de métricas durante el proceso de despliegue de sistemas de software?* Esta pregunta principal se descompone en un conjunto de sub-preguntas (PI1-3), las cuales se presentan en la Tabla 1 junto a la motivación de cada una de ellas (MO1-3).

Tabla 2.1 Preguntas de investigación (PI) y motivación (MO).

Ref.	Preguntas	Ref.	Motivación
PI1	¿Qué contribuciones existen para la mejora del proceso de despliegue?	MO1	Determinar que contribuciones existen para la mejora del proceso de despliegue.
PI2	¿Cuáles son los atributos o factores de calidad más estudiados para el proceso de despliegue?	MO2	Descubrir sobre qué atributos o factores se centran las métricas propuestas.
PI3	¿Qué tipos de investigaciones se utilizan?	MO3	Identificar los tipos de investigación de los estudios primarios según la clasificación de Wieringa (2005), en concordancia con los tipos de investigación recomendados por Petersen (Petersen <i>et al.</i> , 2008; Petersen <i>et al.</i> , 2015).

Se decidió realizar una búsqueda automática en las librerías y plataformas digitales descritas en la Tabla 2.2, se considerando artículos de congresos y artículos de revistas. La búsqueda se realizó en el período comprendido entre el año 2010 hasta diciembre del año 2020. Se decidió para el año de inicio de la búsqueda, el 2010 por tratarse de un año relevante dado que la aparición de tendencias y desafíos en ese momento tuvieron un impacto en los procesos de software. Entre estos, podemos mencionar el Desarrollo Global de Software e Internet como entorno de desarrollo, una infraestructura arquitectónica y de ejecución y una infraestructura básica de distribución y negocios, lo que implica un cambio en los procesos de construcción de software y en el modo de operación de los sistemas software, con la aparición del software móvil (Fuggetta *et al.*, 2014).

Tabla 2.2 Bibliotecas y repositorios digitales utilizadas.

Bibliotecas/Repositorios/Plataformas	Opciones
Google Academic	Artículos de congresos, artículos de revistas
SCOPUS	Artículos de congresos, artículos de revistas
Biblioteca digital de ACM	Artículos de congresos, artículos de revistas
<i>IEEE Xplore</i>	Artículos de congresos, artículos de revistas
ScienceDirect	Artículos de congresos, artículos de revistas

En una primera instancia y como mecanismo de prueba de las librerías y plataformas digitales, se realizó una búsqueda piloto con la cadena que se encuentra en la Tabla 2.3. Esta cadena se debió adaptar para algunas librerías, dado que su formato y longitud no permitían realizar búsquedas, además de devolver artículos que no resultaron relevantes para la investigación. Los términos se emplearon en el idioma inglés. La Tabla 2.4 presenta las cadenas definitivas para la búsqueda de acuerdo con la librería o repositorio utilizado.

Tabla 2.3 Cadena de búsqueda piloto.

Cadena de la búsqueda piloto
<i>("software system" OR "software" OR "software application" OR "computer software") AND ("deployment" OR "transition") AND ("metric" OR "metrics" OR "measurement" OR "Quality metric" OR "indicator")</i>

Tabla 2.4 Cadenas de búsqueda definitiva.

Biblioteca/Repositorio	Cadena
Google Academic	<p>Con todas las palabras: deployment metrics</p> <ol style="list-style-type: none"> 1. Con al menos una de las palabras: "software system" OR software OR "software application" OR "computer software" OR deployment OR transition OR metric OR metrics OR measurement OR "Quality metric" 2. Con al menos una de las palabras: "software system" OR software OR "software application" OR "computer software" OR deployment OR transition OR metric OR metrics OR measurement OR "Indicator" <p>Artículos comprendidos entre enero del 2010 y diciembre del 2020</p> <p>Cadenas finales:</p> <p>allintitle: deployment metrics "software system" OR software OR "software application" OR "computer software" OR deployment OR transition OR metric OR metrics OR measurement OR "Indicator"</p> <p>allintitle: deployment metrics "software system" OR software OR "software application" OR "computer software" OR deployment OR transition OR metric OR metrics OR measurement OR "Quality metric"</p> <p>Debido a una limitación del buscador en relación a la cantidad de conectores admitidos dentro de una cadena de búsqueda, la cadena original se debió ejecutar en dos partes.</p>
SCOPUS	(TITLE ("software system" OR "software" OR "software application" OR "computer software") AND TITLE-ABS-KEY ("deployment" OR "transition") AND TITLE-ABS-KEY ("metric" OR "metrics" OR "measurement" OR "Quality metric" OR "indicator")) AND DOCTYPE (ar OR cp) AND PUBYEAR > 2009 AND PUBYEAR < 2021 AND (LIMIT-TO (SUBJAREA , "COMP"))
Biblioteca digital de ACM	[[Abstract: "deployment"] OR [Abstract: "transition"]] AND [[Abstract: "software system"] OR [Abstract: "software"]] OR [Abstract: "software application"] OR [Abstract: "computer software"]] AND [[Abstract: "metric"] OR [Abstract: "metrics"] OR [Abstract: "measurement"] OR [Abstract: "quality metric"] OR [Abstract: "indicator"]] AND [Publication Date: (01/01/2010 TO 12/31/2020)]
<i>IEEE Xplore</i>	(("Title": "software system" OR "software" OR "software application" OR "computer software") AND ("Title": "deployment" OR "delivery" OR "transition" OR "implantation") AND ("Title": "metric" OR "metrics" OR "measurement" OR "Quality metric" OR "indicator"))
ScienceDirect	("software system" OR "software" OR "software application" OR "computer software") AND ("deployment" OR "transition") AND ("metric" OR "metrics" OR "measurement" OR "Quality metric") Date: 2010 to 2020 Article type: Research article Subject areas: Engineering

Biblioteca/Repositorio	Cadena
	Publication Title: Journal of Network and Computer Applications, Procedia Engineering, Computer Networks.

Los criterios de inclusión y exclusión utilizados para el proceso de selección se presentan en la Tabla 2.5.

Tabla 2.5 Criterios de inclusión y exclusión.

Criterios de inclusión:
<ul style="list-style-type: none"> • Artículos desde enero del 2010 hasta diciembre del 2020. • Artículos en el idioma inglés. • Artículos que contengan cadenas candidatas en el título, palabras clave y/o en el resumen. • Artículos duplicados: si hay varios artículos de un mismo autor que contemple la misma investigación, se considerara el más completo.
Criterios de exclusión:
<ul style="list-style-type: none"> • Artículos que no cumplan los criterios de inclusión. • Estudios que no cuenten con resultados claros. • Literatura gris, tesis, presentaciones en power point, artículos que cuenten solamente con el resumen. • Artículos a los cuales no se tiene acceso.

Es importante mencionar que si bien el método del SMS propuesto en (Kitchenham *et al.*, 2015) (Fuggetta *et al.*, 2014) sugiere que la búsqueda debe realizarse sobre metadatos homogéneos en las librerías digitales, dada la diversidad de librerías digitales y plataformas empleadas (en abierto o por suscripción), se debió realizar la búsqueda combinando los metadatos considerando en algunos casos el título, las palabras clave y en otros el título y/o resumen.

El proceso de selección de los estudios consistió en los siguientes pasos:

- 1) Realizar la búsqueda en las fuentes definidas aplicando la cadena en el título y/o en el resumen.
- 2) Eliminar los artículos duplicados.
- 3) Aplicar los criterios de inclusión y exclusión en el título, resumen y palabras clave.
- 4) Aplicar los criterios de inclusión y exclusión al texto completo. Este proceso permitió la selección de los estudios primarios que se analizaron para dar respuesta a las preguntas de investigación (PI) formuladas.

Para dar respuesta a cada una de las preguntas de investigación (PI) se definió un esquema de clasificación. Algunas de las categorías del esquema de clasificación se definieron durante la

planificación del SMS, otras surgieron a partir de la lectura del artículo completo. La descripción de las categorías utilizadas en el esquema de clasificación se presenta en la Tabla 2.6.

Tabla 2.6 Esquema de clasificación de estudios primarios.

Dimensión	Categorías
Contribución.	Métricas, algoritmos, métodos, modelos, frameworks, otros, no informa.
Atributos o factores de calidad	Costo, esfuerzo, confiabilidad, eficiencia, Otros, No informa.
Tipos de Investigación.	Investigación de evaluación, artículo filosófico, propuesta de solución, investigación de validación, reporte de experiencia personal, artículo de opinión. Se utilizó la clasificación de tipos de investigación propuesta por Wieringa <i>et al.</i> en (Wieringa, <i>et al.</i> , 2005).

El formulario de extracción de datos que se presenta en la Tabla 2.7, se compone de dos partes; la primera se refiere a los metadatos de cada uno de los estudios primarios y la segunda se refiere a cada una de las preguntas de investigación (PI).

Tabla 2.7 Formulario de extracción de datos.

Metadatos	Id Artículo, Año, Título, Autor/es, Nombre de la publicación, Tipo de publicación (revista o congreso), Fuente, Origen de búsqueda, Palabras clave.
PI/Dimensión	Categorías
PI1/Contribución.	Métrica, algoritmo, método, modelo, framework, otros, no informa.
PI2/ Atributos o factores de calidad.	Costo, esfuerzo, confiabilidad, eficiencia, Otros, No informa.
PI3/ Tipos de investigación.	Investigación de evaluación, artículo filosófico, propuesta de solución, investigación de validación, reporte de experiencia personal, artículo de opinión. Se utilizó la clasificación de tipos de investigación propuesta por Wieringa <i>et al.</i> en (Wieringa <i>et al.</i> , 2005).

Para dar respuesta a las preguntas de investigación (PI) definidas en el protocolo, se utiliza una síntesis temática basada en el esquema de clasificación que se representará a través de tablas y gráficos.

2.2. EJECUCIÓN DEL SMS

En esta sección, se presenta la búsqueda realizada en las librerías y plataformas digitales, la selección de estudios primarios de acuerdo con lo definido en el protocolo de revisión del SMS.

Se aplicó la cadena de búsqueda en las librerías definidas en la estrategia de búsqueda (Tabla 2.2 de la sección 2.1.). De un total de 3137 artículos obtenidos tras aplicar los criterios de selección, se detectaron 29 artículos relevantes y de estos, finalmente se analizaron 7 artículos primarios.

La selección de los estudios primarios ha sido realizada por el tesista en formación respetando el proceso de selección definido en el protocolo de revisión (sección 2.1.). En paralelo, la directora de tesis replicó el proceso de selección para la obtención de estudios primarios. Los dos conjuntos de estudios primarios se validaron por el tesista y los directores, se discutieron las discrepancias con el propósito de determinar que artículos eran apropiados de incluir o no en el estudio. Finalmente, luego de aplicar el proceso de selección, se consideraron 7 estudios primarios para su análisis con el propósito de dar respuesta a las preguntas de investigación definidas (sección 2.1.).

Con el propósito de incrementar la cantidad de artículos relevantes, se aplicó el método “bola de nieve” (en inglés, “*snowballing*”) de acuerdo con los lineamientos sugeridos por Wohlin (Wohlin, 2014). Este especifica que, una vez que se identifica un estudio primario en una de las fuentes de búsqueda, se explora de manera recursiva las referencias del estudio utilizando los mismos criterios definidos para la búsqueda. Finalmente, luego de aplicar el método “bola de nieve” no se encontró ningún artículo para ser considerado en el análisis. La lista completa de los estudios primarios analizados se presenta en la Tabla 2.8.

Tabla 2.1 Listado de estudios primarios.

Id	Estudio Primario
[EP1]	Soner, S., Jain, A., Saxena, D. (2010). Metrics calculation for deployment process. 2010 2nd International Conference on Software Technology and Engineering. DOI: 10.1109/ICSTE.2010.5608760.
[EP2]	Xia, T., Bhardwaj, S., Dmitriev, P., & Fabijan, A. (2019). Safe Velocity: A Practical Guide to Software Deployment at Scale using Controlled Rollout. 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). DOI:10.1109/icse-seip.2019.00010.
[EP3]	Dubois, D. J., Trubiani, C., Casale, G. (2016). Model-driven application refactoring to minimize deployment costs in preemptible cloud resources. In 2016 IEEE 9th International Conference on Cloud Computing (CLOUD) (pp. 335-342). DOI 10.1109/CLOUD.2016.50.
[EP4]	Su, X., Liu, H., Wu, Z., Zuo, D., Yang, X. (2010). SA based software deployment reliability estimation: Problem space, challenges and strategies. IEEE 2010 International Conference on Educational and Information Technology (ICEIT). DOI:10.1109/iceit.2010.5607510.
[EP5]	Lehtonen, T., Suonsyrjä, S., Kilamo, T., Mikkonen, T. (2015). Defining metrics for continuous delivery and deployment pipeline. Proceedings of the 14th Symposium on Programming Languages and Software Tools (SPLST'15) Tampere, Finland, Oct 9-10, 2015. Published on CEUR-WS: 14-Dec-20.15. ONLINE: http://ceur-ws.org/Vol-1525/ URN: urn:nbn:de:0074-1525-1. (pp 16-30).
[EP6]	Wan, X., Guan, X., Wang, T., Bai, G., Choi, B. (2018). Application deployment using Microservice and Docker containers: Framework and optimization. Journal of Network and Computer Applications, 97–109. DOI:10.1016/j.jnca.2018.07.003

[EP7] Rodrigues, G., Guimaraes, F., Rodrigues, G., Knauss, A., Joao Paulo C. de Araújo, J., Andrade, H., Ali, R., (2019). GoalD: A Goal-Driven Deployment Framework for Dynamic and Heterogeneous Computing Environments. Information and Software Technology. DOI: <https://doi.org/10.1016/j.infsof.2019.04.003>.

2.3. RESULTADOS DEL SMS

En esta sección se realiza un análisis de los estudios primarios obtenidos para dar respuesta a cada una de las preguntas de investigación (sección 2.3.1.) y además exhiben algunos hallazgos adicionales mediante el uso de gráficos y tablas (sección 2.3.2.).

2.3.1. RESPUESTA A LAS PREGUNTAS DE INVESTIGACIÓN

En la Tabla 2.9 se presenta una síntesis de los resultados obtenidos del análisis de los estudios primarios presentados en la Tabla 2.8, de acuerdo con el esquema de clasificación definido en la Tabla 2.6 de la sección 2.2

Tabla 2.1 Resultados por preguntas de investigación (PIs).

Referencia	Resultados por cada PI		
	Contribución (PI1)	Atributo o factor de calidad (PI2)	Tipos de investigación (PI3)
[EP1]	Métricas.	Esfuerzo.	Propuesta de solución.
[EP2]	Modelo.	Esfuerzo. Confiabilidad.	Evaluación.
[EP3]	Modelo.	Costo.	Validación.
[EP4]	Modelo.	Confiabilidad.	Propuesta de solución.
[EP5]	Métricas.	Esfuerzo.	Evaluación.
[EP6]	Algoritmo. Framework.	Costo.	Validación.
[EP7]	Framework.	Esfuerzo.	Validación.

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

PI1: ¿Qué contribuciones existen para la mejora del proceso de despliegue?

Soner *et al.* [EP1] en su investigación, exponen factores que afectan al proceso de despliegue, como la cobertura del proceso, capacidad de cambio del proceso, coordinación entre procesos y abstracción de políticas de despliegue. Para minimizar estas dificultades presenta tres métodos para el cálculo del costo y el esfuerzo del proceso de despliegue. El primero se basa en un porcentaje fijo del esfuerzo del proceso de desarrollo. En el segundo método, el cálculo del esfuerzo se basa en la garantía de

calidad, la prueba de aceptación del usuario y la producción. Finalmente, el último método que presenta se basa en la experiencia.

Xia *et al.* [EP2] con el propósito de ayudar a las empresas a optimizar sus prácticas de despliegue de software, presentan un modelo híbrido entre experimentación controlada y lanzamiento por fases, a las cuales llama anillos. A este modelo lo denomina despliegue controlado. En su trabajo enuncia las estrategias para diseñar y organizar métricas de despliegue, para decidir la duración de la implementación y las estrategias para el diseño y la toma de decisiones en anillos individuales.

Dubois *et al.* [EP3] en su modelo, hacen uso de la evaluación de modelos de Redes de colas como métrica de costos que se relaciona con el tiempo de procesamiento para optimizar los costos de ejecución de aplicaciones en la nube, con el fin de minimizar el costo de su despliegue requiriendo menos capacidad computacional y menos recursos.

Su *et al.* [EP4] basan su investigación en los desafíos para la estimación de la confiabilidad en el despliegue de software basado en arquitectura de software. Como propuesta, presentan y discuten los 17 parámetros propuestos para el cálculo de la confiabilidad.

Lehtonen *et al.* [EP5] realizan la propuesta de métricas para un proceso de despliegue continuo. El objetivo de las métricas que propone es proporcionar información valiosa al equipo que les permita mejorar el proceso de automatización del despliegue de software.

En Wan *et al.* [EP6] se propone un framework de comunicación y un algoritmo de asignación de recursos escalable para minimizar el costo del despliegue de la aplicación en entornos de computación en la nube.

El modelo que proponen Rodrigues *et al.* [EP7] propone un framework para abordar sistemáticamente el despliegue autónomo en entornos informáticos dinámicos y muy heterogéneos.

PI2: ¿Cuáles son los atributos o factores de calidad más estudiados para el proceso de despliegue?

A través de la propuesta de métricas que realizan Soner *et al.* [EP1] buscan calcular el esfuerzo del proceso de despliegue de software.

Xia *et al.* [EP2], con su propuesta de despliegue controlado, no solo buscan optimizar cada una de las fases de despliegue, sino también maximizar la confiabilidad del producto final frente al usuario final.

El modelo de Dubois *et al.* [EP3] permite optimizar los costos de ejecución de aplicaciones en la nube, con el fin de minimizar el costo de su despliegue.

Su *et al.* [EP4] presentan y discuten los 17 parámetros propuestos para el cálculo de la confiabilidad en el despliegue de software basado en arquitectura de software.

Lehtonen *et al.* [EP5] realiza la propuesta de métricas con el propósito de optimizar la automatización del proceso de despliegue de software.

Wan *et al.* [EP6] proponen un framework y un algoritmo con el objetivo de optimizar los costos del despliegue de las aplicaciones.

Rodrigues *et al.* [EP7] proponen un framework que busca optimizar los costos de despliegue de las aplicaciones.

PI3: ¿Qué tipos de investigaciones se utilizan?

De los estudios primarios analizados, dos corresponden a la categoría “propuesta de solución”, tres a la categoría “validación” y dos corresponden a la categoría “evaluación”.

Soner *et al.* [EP1] en su trabajo proponen una idea sobre el procedimiento de despliegue, los factores que deben seguirse y algunos métodos mediante los cuales se puede calcular fácilmente el esfuerzo de despliegue. Su *et al.* [EP4] describen qué información constituye el modelo de confiabilidad de implementación de software basado en arquitectura de software, detallan los desafíos para obtener la información disponible para el modelo y enuncian cómo afrontar esos desafíos.

Dentro de los estudios en los que se realizan evaluaciones en el contexto industrial, se encuentra el de Xia *et al.* [EP2], en su investigación, presentan el despliegue controlado, que aplica la experimentación controlada, experimentos simultáneos con diferentes variaciones del mismo producto a cada etapa de un despliegue por fases tradicionales. Este consiste en despliegues para audiencias más pequeñas antes de implementarlos de forma generalizada. Para evaluar la efectividad del despliegue controlado se presenta un estudio de caso en Microsoft Corporation en los EE. UU. entre julio de 2017 y agosto de 2018. Durante este período se examinaron cientos de experimentos controlados. Lehtonen *et al.* [EP5] que llevaron a cabo un estudio de caso único industrial para identificar qué datos son creados automáticamente por las diversas herramientas en la automatización del despliegue. Los resultados obtenidos mostraron que muchas métricas útiles se generan automáticamente. Basándose en esos datos, definieron nuevas métricas para describir las propiedades de la canalización del despliegue las cuales fueron discutidas con el equipo de desarrollo del proyecto de caso.

Dentro de los artículos en los que se realizan validaciones de las soluciones, se encuentran Dubois *et al.* [EP3] que realizan la validación de su propuesta de modelo bajo diferentes escenarios inspirados en un sistema real. Los resultados que obtuvieron manifiestan que reduce los costos de implementación, hasta en un 60% en comparación con el mismo modelo, pero sin refactorización de aplicaciones impulsada por modelos. Wan *et al.* [EP6] se propone una validación de su propuesta mediante comparaciones con tres estrategias existentes en Docker Swarm utilizando trazas reales de Google Cluster. Demuestran que el framework y el algoritmo propuesto proporcionan más flexibilidad y ahorran más costos que las estrategias existentes. Y, por último, para validar la escalabilidad de su propuesta, Rodrigues *et al.* [EP7] diseñaron escenarios que reflejan las incertidumbres inherentes de los sistemas autoadaptativos. Los resultados de la validación demuestran la utilidad y escalabilidad de GoalD en la planificación de la implementación de un escenario razonablemente grande en solo unos pocos milisegundos.

2.3.2. HALLAZGOS ADICIONALES

Dentro de los hallazgos adicionales, se presentan diferentes aspectos como, por ejemplo, la cantidad de artículo encontrados por año de publicación, los foros de publicación, los países de donde proviene los artículos y el contexto donde se realizaron las investigaciones (académico o en la industria).

La Figura 1 presenta la cantidad de artículos por año. Aun observando que los años 2010 y 2019 presentan los picos más altos de publicaciones, ambos años con dos artículos cada uno, no es posible determinar una tendencia clara sobre el tipo de contribución de los estudios. La mayoría de los artículos encontrados fueron publicados en congresos, lo que representa el 71% (5 estudios) del total, y las publicaciones en revistas el 29% (2 estudios), como se muestra en la Figura 2. Respecto al evento donde los artículos fueron publicados, se encontró que existe una gran diversidad de congresos y revistas.

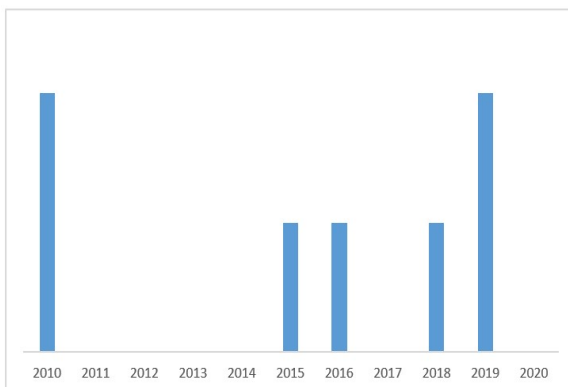


Figura 2.1. Cantidad de artículos por año.

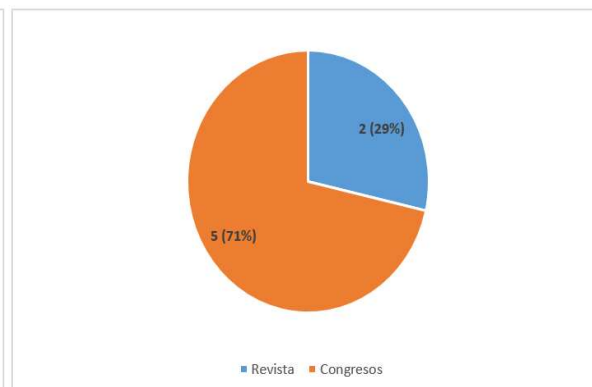


Figura 2.2. Cantidad de artículos por tipo de publicación.

La Tabla 2.10 sintetiza los estudios primarios por librería digital utilizada (Scopus, *IEEE Xplore* o Biblioteca digital de ACM) o plataformas abiertas (Google Academic y ScienceDirect), título de publicación, tipo de publicación (conferencia o revista) y año. El mayor número de estudios se encontró en *IEEE Xplore*.

Tabla 2.2 Estudios primarios por fuente de búsqueda, título y tipo de publicación, y año.

ID Artículo	Fuente	Título de publicación	Tipo de Publicación	Año
[EP1]	<i>IEEE Xplore</i>	2nd International Conference on Software Technology and Engineering	Congreso	2010
[EP2]	<i>IEEE Xplore</i>	IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)	Congreso	2019
[EP3]	<i>IEEE Xplore</i>	IEEE 9th International Conference on Cloud Computing (CLOUD)	Congreso	2016
[EP4]	<i>IEEE Xplore</i>	International Conference on Educational and Information Technology	Congreso	2010
[EP5]	Google Academic	14th Symposium on Programming Languages and Software Tools.	Congreso	2015
[EP6]	ScienceDirect	Information and Software Technology	Revista	2018
[EP7]	ScienceDirect	Journal of Network and Computer Applications	Revista	2019

Otro análisis realizado corresponde al país donde se originaron las publicaciones. Se identificó que los orígenes se encuentran muy dispersos, como se muestra en la Tabla 2.11.

Tabla 2.3 Síntesis por país.

ID Artículo	País	Cantidad de artículos
[EP6][EP7]	United Kindom	2
[EP1][EP3]	USA	2
[EP2]	Canada	1
[EP4]	China	1
[EP5]	Finlandia	1

En la Tabla 2.12 se puede observar que cinco de estos estudios pertenecen al campo académico, mientras que un estudio es desarrollado por la academia y la industria, así como también uno solo por la industria.

Tabla 2.4 Síntesis por contexto.

Contexto de estudio	Cantidad de artículos	ID Artículo
Académico	5	[EP1][EP3][EP4][EP6][EP7]
Industria	1	[EP2]
Académico e Industria	1	[EP5]

2.4. AMENAZAS A LA VALIDEZ

En este apartado analizamos las potenciales amenazas a la validez que podrían afectar a nuestro estudio, con respecto a las cuatro categorías sugeridas por Wohlin *et al.* (2014).

- **Construir Validez del constructo.** En este SMS, con el fin de mitigar estas amenazas, describimos el significado que le hemos dado al proceso de despliegue del sistema de software, basado en estándares y metodologías reconocidas internacionalmente (IEEE ISO/IEC/IEEE 12207, 2017); (Jacobson *et. al.*, 1999); (Agile Business Consortium, 2016).
- **Validez interna.** Para mitigar las preocupaciones sobre la validez interna, el primer autor creó un protocolo de revisión como parte de la investigación de su tesis de Maestría, y esto fue revisado por los directores de la tesis.
- **Validez externa.** Se tomó la decisión de utilizar tres motores de búsqueda en nuestra búsqueda de las revistas y actas de congresos que son relevantes y recomendados para el campo de la Ingeniería de Software y complementar con dos plataformas abiertas reconocidas por la comunidad científica. No se consideró la literatura gris, como los artículos disponibles solo en forma de resúmenes, presentaciones en PowerPoint, tesis doctorales, porque incluirlos podría haber afectado la validez de nuestros resultados.
- **Fiabilidad.** Se intentó mitigar el sesgo de las publicaciones definiendo cuidadosamente (a) los criterios de inclusión y exclusión para poder seleccionar estudios primarios y (b) los criterios de exclusión específicamente, con el fin de seleccionar reglas basadas en las preguntas de investigación predefinidas en el trabajo. Para aumentar la confiabilidad, el tesista aplicó los criterios con la ayuda de los directores cada vez que se presentaba algún desacuerdo con respecto a la inclusión o exclusión de una regla. Paralelamente, los directores replicaron el proceso de selección y se obtuvo un segundo conjunto de estudios primarios. Los dos

conjuntos de estudios primarios fueron validados por el tesista y los directores; se discutieron las discrepancias entre ellos, con el propósito de determinar si era apropiado incluir un artículo en particular o no, y de ese modo se obtuvo el listado final de estudios primarios. Además, se diseñó un formulario para la registración de los datos con Excel y se mapearon las preguntas de investigación de acuerdo con el esquema de clasificación definido para cumplir con los objetivos de este estudio. Se considera que el efecto potencial de este sesgo tiene menos importancia en estudios de mapeos sistemáticos que en las revisiones sistemáticas de literatura. Para corroborar aún más el proceso de búsqueda, se aplicó el método “*snowballing*”, pero desafortunadamente no permitió encontrar más estudios primarios que los hallados en la búsqueda inicial.

2.5. CONCLUSIONES DEL SMS.

El SMS presentado permitió analizar el estado del arte y descubrir si existen métricas para el proceso de despliegue de sistemas de software. En este estudio se seleccionaron 7 estudios primarios relevantes de un conjunto inicial de 3137 artículos. Estos se obtuvieron una vez realizada la búsqueda en las fuentes SCOPUS, *IEEE Xplore*, Biblioteca digital de ACM, Google Academic y ScienceDirect en el período comprendido entre enero del 2010 y diciembre del 2020. Una vez analizados los estudios primarios, se concluye que:

- Todos tienen un punto en común que es la optimización del esfuerzo y costos del proceso de despliegue.
- El 29% de los estudios primarios analizados proponen métricas que intervienen en el proceso de despliegue. Por otro lado, el 29% de las publicaciones proponen un framework para el proceso de despliegue. Mientras que el 43%, presenta un modelo para el proceso de despliegue de sistemas de software. Y el 14%, realiza la propuesta de un algoritmo para dicho proceso.
- El 57% de las publicaciones enfoca su propuesta a la optimización del esfuerzo del proceso de despliegue, el 29% se centran en una reducción de los costos y el 29% restante aborda su propuesta con el fin de incrementar la confiabilidad.
- El 42 % de las publicaciones realizan una validación de la propuesta de solución, el 29 % de los estudios presentan una evaluación de la propuesta de solución y el 29% corresponden a propuestas de solución. No se identificaron artículos que informen una experiencia u opinión.

Dada la vacancia del uso de métricas para el proceso de despliegue de sistemas de software, que se logró evidenciar mediante el desarrollo del SMS y la necesidad de las PyMEs de Argentina de fortalecer el proceso de despliegue de sistemas de software, se plantea el diseño de un conjunto de métricas.

CAPÍTULO 3. DEFINICIÓN DE MÉTRICAS

En este capítulo se describe el proceso realizado para la definición del conjunto de métricas para el proceso de despliegue de sistemas de software, así como también los estándares en los cuales se basa la propuesta junto con 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. En la sección 3.2 se presenta una clasificación de métricas que se ha considerado como base para esta propuesta. A continuación, en la sección 3.3 explica la visión tridimensional del proceso de despliegue “Proceso, Producto y Persona” sobre la cual se definen las métricas para cada una de estas dimensiones. Finalmente, en la sección 3.4 se presentan las métricas propuestas.

3.1 ACTIVIDADES Y TAREAS DE LA ISO/IEC/IEEE 12207

Para la definición de las métricas, se contemplaron las actividades y tareas del “proceso de transición” (en inglés, *Transition Process*) del estándar ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE 12207, 2017) por ser un estándar reconocido internacionalmente. La Figura 3.1. presenta los procesos que conforman el estándar ISO/IEC/IEEE/12207 y donde se inserta el “proceso de transición”.

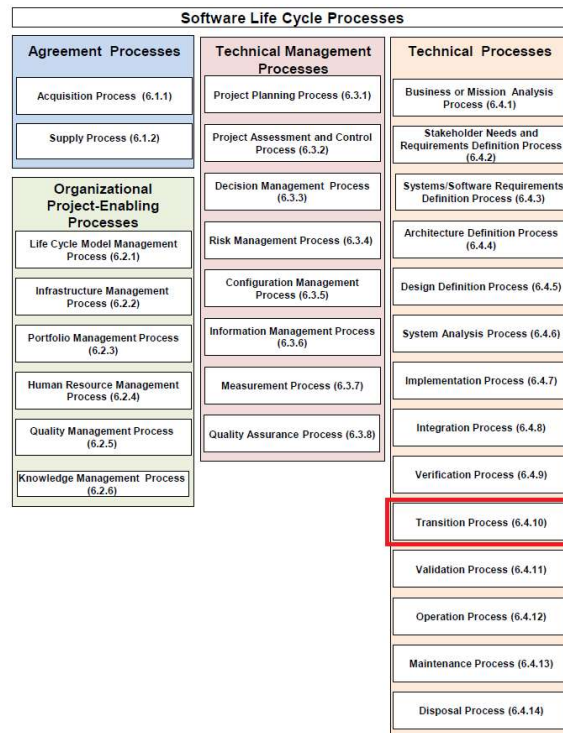


Figura 3.1. Procesos del estándar ISO/IEC/IEEE 12207:2017.

El “proceso de transición” forma parte del grupo de Procesos técnicos de la ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE 12027, 2017) y se compone de las siguientes actividades: 1) Preparación del despliegue, 2) Realización del despliegue y 3) Gestión de los resultados del despliegue. A continuación, se detallan las actividades con sus correspondientes tareas:

1) Preparación del despliegue:

- Definir una estrategia para administrar las versiones de software.
- Identificar y definir los cambios en las instalaciones, el sitio, la red de comunicaciones o el entorno de destino necesarios para el despliegue del sistema de software.
- Identificar las necesidades de información y organizar la documentación del usuario y la capacitación de los operadores, usuarios y otras partes interesadas necesarias para la utilización y el soporte del sistema.
- Preparar información detallada sobre el proceso, como planes, cronogramas y procedimientos.
- Identificar las restricciones del sistema desde la transición para incorporarse en los requisitos, la arquitectura o el diseño del sistema de software.
- Identificar y planificar los sistemas o servicios necesarios para apoyar al proceso.
- Adquirir el acceso a los sistemas o servicios a utilizar.

2) Ejecución del despliegue:

- Preparar el sitio de operación o el entorno virtual de acuerdo con los requisitos de instalación.
- Entregar el sistema o elemento software para su instalación en el lugar y tiempo correctos.
- Instalar el producto en su ubicación operativa física o virtual e interactúe con su entorno.
- Proporcionar documentación de usuario y capacitación para los operadores, usuarios y otras partes interesadas necesarias para la utilización y el soporte del producto.
- Realizar la activación y la verificación.

3) Gestión de los resultados del despliegue:

- Registrar los resultados del despliegue.
- Registrar incidentes y problemas del despliegue y realizar un seguimiento de su resolución.
- Mantener la trazabilidad de los elementos del sistema de software desplegado.

- Proporcionar artefactos clave y elementos de información que se hayan seleccionado para las líneas base.

Para una mejor estructuración de la solución, así como también para que su aplicación en la industria del software se realice de manera sistemática en diferentes proyectos de despliegue se define una codificación para el proceso en estudio.

Para ello, se utiliza la propuesta de Runeson *et al.* (Runeson. *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. A continuación, se detallan estos lineamientos:

- 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 basados en agrupaciones de códigos y en este punto de la solución permite codificar los tres grupos de actividades que conforman el proceso de transición del estándar ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE 12027, 2017).

La codificación se aplica de la siguiente manera:

- 1) Código primario: Actividad (A)

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

Tabla 3.1 Actividades del “Proceso de Transición”.

Actividad
A1 Preparación del despliegue.
A2 Realización del despliegue.
A3 Gestión de los resultados del despliegue.

3.2 TIPIFICACIÓN DE LAS MÉTRICAS SEGÚN LA ISO 9126

Para tipificar las métricas se usó la clasificación propuesta por la norma ISO/IEC 9126 (ISO/IEC 9126-2, 2003; ISO/IEC 9126-3, 2003; ISO/IEC 9126-4, 2004) la cual divide las métricas en tres categorías, según su naturaleza:

- **Métricas básicas:** son métricas que se obtienen directamente del análisis del código o la ejecución del software. No involucra ningún otro atributo ni depende de otras métricas. Pressman denomina a estas métricas, directas (Pressman, 2015). Ejemplos de estas métricas son: la cantidad de líneas de código del programa o de cada módulo, la cantidad de horas de desarrollo, la cantidad de fuentes de datos o tablas a utilizar, la cantidad de atributos y registros de una tabla, entre otras.
- **Métricas de agregación:** son métricas compuestas a partir de un conjunto definido de métricas básicas (o directas), generalmente mediante una suma ponderada.
- **Métricas derivadas:** son métricas compuestas por una función de cálculo matemático, que utiliza como variables de entrada el valor de otras métricas. Pressman denomina a estas métricas indirectas (Pressman, 2015). Ejemplos de estas métricas son: la cantidad de líneas de código producidas por hora y por persona, el porcentaje de completitud del proyecto, el tamaño promedio de los módulos del software, el tiempo promedio que una persona dedica a corregir los defectos de un módulo, entre otras.

3.3 TAXONOMÍA DE LAS MÉTRICAS

Para organizar el conjunto de métricas definidos en esta tesis, por algún tipo de clasificación, se adoptó la visión de métricas basada en tres enfoques, también denominada “visión tridimensional” del proceso de despliegue propuesta por Vazquez *et al.* (Vazquez P., *et al.*, 2018). Esta visión 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”, 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 y el tamaño, entre otras. Y finalmente, la última dimensión denominada “Persona”, se la considera debido a la existencia del

peopleware y su impacto en el proceso de implantación de sistemas informáticos (Panizzi *et al.*, 2017).

3.4 MÉTRICAS DEFINIDAS PARA EL PROCESO DE DESPLIEGUE

Para cada una de las actividades del proceso de transición del estándar ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE, 2017) descritas en la sección 3.1 y de acuerdo con la visión tridimensional del proceso de despliegue “Proceso, Producto y Persona”, se definieron las métricas que se describen en las secciones siguientes.

3.4.1 MÉTRICAS PARA LA DIMENSIÓN “PROCESO”

En la Tabla 3.2, se presenta para cada una de las actividades del proceso, la clasificación basada en la tipificación mencionada en la sección 3.2, y el acrónimo de las métricas propuestas para la dimensión “Proceso” y en la Tabla 3.3 se presenta una descripción de cada una de las métricas propuestas para esta dimensión.

Tabla 3.1 Métricas de la Dimensión “Proceso” para cada una de las actividades.

Actividades	Clasificación	Métrica
A1 Preparación del despliegue.	Básica.	Capacitaciones (CAP). Horas de entrenamiento planificada (HEP). Pruebas de aceptación planificada (PAP). Personas a entrenar planificada (PEP). Esfuerzo estimado de la tarea (ET). Tareas planificadas (TP).
	Derivadas.	Esfuerzo (ESF).
A2 Realización del despliegue. A3 Gestión de los resultados del despliegue.	Básica.	Capacitaciones realizadas (CAPR). Horas de entrenamiento ejecutadas (HEE) Pruebas de aceptación ejecutadas (PAE). Personas entrenadas (PE). Esfuerzo real de tarea (ERT). Tareas realizadas (TR).
	Agregación.	Esfuerzo real del proceso (ESFR).
	Derivadas.	% de desvío de capacitaciones (%DesvíoCap). % de desvío de las horas de entrenamiento (%DesvíoHE). % de desvío de las personas entrenadas (%DesvíoPEnt). % de avance del entrenamiento (%AvanceE).

Actividades	Clasificación	Métrica
		% de avance de pruebas de aceptación (%AvancePA).
		% de desvío de las pruebas de aceptación (%DesvíoPA).
		% de desvío del esfuerzo (%DesvíoEsf).
		% de avance de las tareas (%AvanceTR).
		% de desvío de las tareas (%DesvíoTR).
		Productividad del proceso (PRD).
		Productividad de la tarea (PRDT).

Tabla 3.2 Descripción de las Métricas de la Dimensión “Proceso”.

Acrónimo de la métrica	Descripción	Fórmula
ESF	<p>Esfuerzo. Indica al esfuerzo Total requerido para completar el proceso. Este valor se considera una métrica híbrida (Trendowicz y Jeffery, 2014). La unidad de medida de la métrica es horas.</p> <p>EEK: Esfuerzo estimado Karner. Esfuerzo calculado utilizando la estimación por punto de caso de Uso de Karner (Karner, 1993).</p> <p>EEPP: Esfuerzo estimado por Planning Poker. Calculado haciendo uso del método de estimación de esfuerzo de Planing Poker (Grenning, 2002).</p>	$ESF = (EEK+EEPP)/2$
%DesvíoCAP	<p>% Desvío capacitaciones. Indica el desvío entre las capacitaciones planificadas (CAP) y las capacitaciones realizadas (CAPR). La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.</p> <p>CAP: indica la cantidad de capacitaciones que se realizarán. La unidad de medida es un valor numérico y toma valores mayores a 0.</p> <p>CAPR: cantidad de capacitaciones realizadas. La unidad de medida es un valor numérico y toma valores mayores o iguales a 0.</p>	$\%DesvíoCap = 100 - (CAPR / CAP)$
%DesvíoHE	<p>% Desvío entrenamiento. Indica el desvío entre las horas de entrenamiento planificadas (HEP) y las horas entrenamiento ejecutadas (HEE). La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.</p> <p>HEP: indica el total de horas de entrenamiento planificadas. La unidad de medida es horas y toma valores mayores a 0.</p> <p>HEE: indica el total de horas de entrenamiento ejecutadas. La unidad de medida es horas y toma valores mayores a 0.</p>	$\%DesvíoHE = 100 - (HEE / HEP)$
%DesvíoPA	<p>% Desvío prueba aceptación. Indica el desvío entre las pruebas de aceptación planificadas (PAP) y las pruebas de aceptación ejecutadas (PAE). La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.</p> <p>PAP: indica el total de pruebas de aceptación planificadas. La unidad de medida es un valor numérico y toma valores mayores a 0.</p> <p>PAE: indica el total de pruebas de aceptación ejecutadas. La unidad de medida es un valor numérico y toma valores mayores a 0.</p>	$\%DesvíoPA = 100 - (PAE / PAP)$
%DesvíoPEnt	<p>% Desvío personas entrenadas. Indica el desvío entre la cantidad planificada de personas para su entrenamiento (PEP) y la cantidad de personas entrenadas (PE). La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.</p> <p>PEP: indica la cantidad planificada de personas para su entrenamiento. La unidad de medida es un valor numérico y toma valores mayores a 0.</p>	$\%DesvíoPEnt = 100 - (PE / PEP)$

Acrónimo de la métrica	Descripción	Fórmula
	PE: indica la cantidad de personas entrenadas. La unidad de medida es un valor numérico y toma valores mayores a 0.	
%AvanceE	<p>% Avance entrenamiento. Indica el porcentaje de avance del entrenamiento. La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.</p> <p>HEP: indica el total de horas de entrenamiento planificadas. La unidad de medida es horas y toma valores mayores a 0.</p> <p>HEE: indica el total de horas de entrenamiento ejecutadas. La unidad de medida es horas y toma valores mayores a 0.</p>	$\%AvanceE = HEE / HEP$
%AvancePA	<p>% Avance pruebas aceptación. Indica el avance de las pruebas de aceptación. Este valor se puede obtener de forma parcial en la medida que avanza el proceso. La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.</p> <p>PAP: indica el total de pruebas de aceptación planificadas. La unidad de medida es un valor numérico y toma valores mayores a 0.</p> <p>PAE: indica el total de pruebas de aceptación ejecutadas. La unidad de medida es un valor numérico y toma valores mayores a 0.</p>	$\%AvancePA = PAE / PAP$
%DesvíoEsf	<p>%Desvío esfuerzo. Desfasaje entre el esfuerzo estimado y el real. La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.</p> <p>ESF: indica el esfuerzo calculado en la planificación. La unidad de medida es horas y toma valores mayores a 0.</p> <p>ESFR: indica el esfuerzo real consumido en el proceso de despliegue. La unidad de medida es horas y toma valores mayores a 0.</p>	$\%DesvíoEsf = 100 - (ESFR / ESF)$
%DesvíoTR	<p>% Desvío tareas. Indica el desvío entre las tareas planificadas y las tareas realizadas. La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.</p> <p>TP: indica las tareas planificadas. La unidad de medida es un valor numérico y toma valores mayores a 0.</p> <p>TR: indica las tareas completadas en el período analizado. La unidad de medida es un valor numérico y toma valores mayores a 0.</p>	$\%DesvíoTR = 100 - (TR / TP)$
%AvanceTR	<p>%Avance tareas. Indica el avance de las tareas. La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.</p> <p>TP: indica las tareas planificadas. La unidad de medida es un valor numérico y toma valores mayores a 0.</p> <p>TR: indica las tareas completadas en el periodo analizado. La unidad de medida es un valor numérico y toma valores mayores a 0.</p>	$\%AvanceTR = TR / TP$
PRD	<p>Productividad del proceso. Indica el nivel de productividad del proceso una vez que este ha finalizado. La unidad de medida de esta métrica es un % y puede tomar valores mayores a 0.</p> <p>ESF: indica el esfuerzo calculado en la planificación. La unidad de medida es horas y toma valores mayores a 0.</p> <p>ESFR: indica el esfuerzo real consumido en el proceso de despliegue. La unidad de medida es horas y toma valores mayores a 0.</p>	$PRD = ESF/ESFR$
PRDT	<p>Productividad de la tarea. Indica el nivel de productividad de la tarea realizada. Al igual que la productividad del proceso, esta métrica se puede obtener una vez finalizada la tarea. La unidad de medida de la métrica es un % y toma valores mayores a 0.</p> <p>ET: indica el esfuerzo estimado de la tarea planificada. La unidad de medida es horas y toma valores mayores a 0.</p>	$PRDT = ET/ERT$

ERT: indica el esfuerzo real de la tarea. La unidad de medida es horas y toma valores mayores a 0.

3.4.2 MÉTRICAS PARA LA DIMENSIÓN “PRODUCTO”

En la Tabla 3.4, se presenta para cada una de las actividades del proceso, la clasificación basada en la tipificación mencionada en la sección 3.2, y el acrónimo de las métricas propuestas para la dimensión “Producto” y en la Tabla 3.5 se presenta una descripción de cada una de las métricas propuestas para esta dimensión.

Tabla 3.3 Métricas de la Dimensión “Producto” para cada una de las actividades.

Actividades	Clasificación	Métrica
A2 Realización del despliegue. A3 Gestión de los resultados del despliegue.	Básica.	Defectos de Configuración (DC). Defectos de aplicación (DA).
	Agregación.	Cantidad de defectos (CantDef).
	Derivadas.	Densidad de defectos (%DensidadD).

Tabla 3.4 Descripción de las Métricas de la Dimensión “Producto”.

Acrónimo de la métrica	Descripción	Fórmula
DC	Indica la cantidad de defectos de configuración detectados. La unidad de medida es un valor numérico y toma valores mayores a 0.	
DA	Indica la cantidad de defectos de aplicación detectados. La unidad de medida es un valor numérico y toma valores mayores a 0.	
CantDef	Indica la cantidad de defectos obtenidos durante el proceso de despliegue y así como las pruebas de aceptación ejecutadas. La unidad de medida es un valor numérico entero y toma valores mayores a 0.	$CantDef = DC + DA$
%DensidadD	% Densidad de defectos. Indica el porcentaje de defectos (CDE) respecto a las pruebas de aceptación efectuadas (PAE). La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.	$\%DensidadD = CDE / PAE$

3.4.3 MÉTRICAS PARA LA DIMENSIÓN “PERSONA”

En la Tabla 3.6, se presenta para cada una de las actividades del proceso, la clasificación basada en la tipificación mencionada en la sección 3.2, y el acrónimo de las métricas propuestas para la dimensión “Persona” y en la Tabla 3.7 se presenta una descripción de cada una de las métricas propuestas para esta dimensión.

Tabla 3.5 Métricas de la Dimensión “Persona” para cada una de las actividades.

Actividades	Clasificación	Métrica
A1 Preparación del despliegue.	Básica.	Velocidad Histórica (VELH).
A2 Realización del despliegue.	Básica.	Velocidad (VEL).
A3 Gestión de los resultados del despliegue.	Derivadas.	%Variación velocidad (%VVEL).

Tabla 3.6 Descripción de las Métricas de la Dimensión “Persona”.

Acrónimo de la métrica	Descripción	Fórmula
VELH	Velocidad Histórica. Indica la cantidad de tareas realizadas por el equipo en un periodo de tiempo según despliegues históricos. La unidad de medida es un valor numérico y toma valores mayores a 0.	
VEL	Velocidad. Indica la cantidad de tareas realizadas por el equipo en un periodo de tiempo. La unidad de medida es un valor numérico y toma valores mayores a 0.	
%VVEL	% Variación de velocidad. Indica el porcentaje de variación de la velocidad del equipo en relación a la velocidad histórica. La unidad de medida de la métrica es un % y toma valores mayores o iguales a 0.	$\%VVEL = VEL / VELH$

CAPÍTULO 4.

VALIDACIÓN DE LA

SOLUCIÓN

En este capítulo se describen dos estudios de casos desarrollados en PyMEs desarrolladoras de software de Argentina con el propósito de examinar la viabilidad de la aplicación del conjunto de métricas definidas en el capítulo 3. El primer estudio de caso se desarrolló en una PyME Mediana de tramo 2 de Argentina 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 Mediana tramo 1 que se dedica al desarrollo de sistemas de información a medida para clientes de diferentes rubros. En ambos estudios de casos se presentan las métricas propuestas para el 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. Se tuvo acceso a la documentación de cada uno de los proyectos sujeto a un acuerdo de no revelar el nombre de las empresas, así como el compromiso de informar sobre los hallazgos y recomendaciones a considerar en la propuesta de métricas para el proceso de despliegue.

4.1. ESTUDIO DE CASO 1

4.1.1 DISEÑO DEL ESTUDIO DE CASO 1

En esta sección, se describe el estudio de caso, siguiendo los lineamientos propuestos de Runeson *et al.* (Runeson *et al.*, 2012). El objetivo principal del estudio de caso consiste en examinar la viabilidad de la aplicación del conjunto de métricas propuestas para el proceso de despliegue de sistemas de software en un entorno real con el propósito de refinarlas (si fuese necesario). Según la clasificación de Robson (Robson, 2002) se enmarca en los estudios exploratorios. Después de mantener una reunión con el Gerente de sistemas y el líder de proyectos de una PyME de sistemas de la República Argentina a la cual se le proporcionó el conjunto de métricas propuestas (sección 3.4), se trabajó con documentación del despliegue de funcionalidades, para un Portal de Recursos Humanos de una entidad bancaria realizado por la PyME.

4.1.2. PREGUNTAS DE INVESTIGACIÓN

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

PI1: ¿Que mediciones se llevaron a cabo durante las actividades del proceso de despliegue de sistemas de software?

A través de esta pregunta, se busca obtener la información del uso de las métricas propuestas durante la ejecución del proceso de despliegue y el uso dado por la consultora para validar su aplicabilidad.

PI2: ¿De qué manera se puede fortalecer el proceso de despliegue de sistemas de 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 un conjunto de métricas identificando su objetivo y la forma de cálculo.

4.1.3. CASO Y UNIDAD DE ANALISIS

En esta sección se describe el contexto, el caso y la unidad de análisis del estudio de caso. Según la clasificación de Yin (Yin, 2014), es un caso único holístico y sus elementos se presentan en la Figura 4.1.

Contexto: el estudio de caso se realizó en una PyME Mediana de tramo 2 desarrolladora de software sistemas radicada en la Ciudad Autónoma de Buenos Aires, la misma cuenta con un plantel de 430 empleados. Esta empresa desarrolla sistemas de información a medida para clientes de diversos rubros, entre ellos financiero, automotriz, farmacéutico y banca. En los proyectos de software combinan prácticas ágiles con metodologías de desarrollo con ciclo de vida iterativo. Se tuvo acceso a la documentación del despliegue del proyecto sujeto a un acuerdo de no revelar el nombre de la compañía, así como el compromiso de informar sobre los hallazgos y recomendaciones a considerar en la propuesta de métricas para el proceso de despliegue.

Caso: despliegue de un Portal de Recursos Humanos realizado en una entidad bancaria de la República Argentina. Este 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), integración con portal de capacitación a distancia, modificación a la interfase 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.

Unidad de análisis: documentación del despliegue de un Portal de Recursos Humanos.

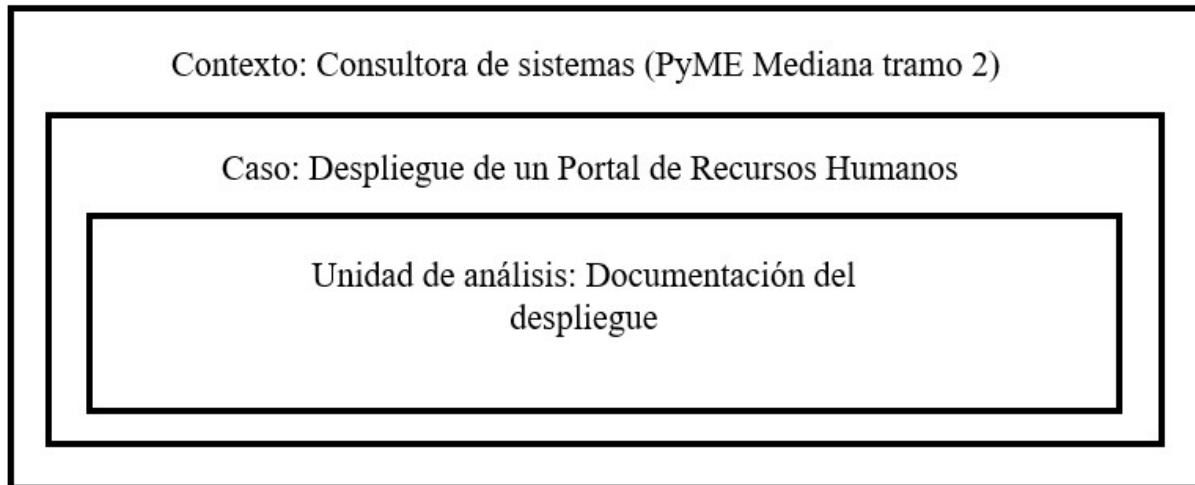


Figura 4.1. Clasificación de estudios de caso basada en la definición de Yin (Yin, R., 2014).

4.1.4. PREPARACION PARA LA RECOLECCIÓN DE DATOS

Se utilizó una técnica de tercer grado combinada con un método independiente según la clasificación propuesta en (Lethbridge *et al.*, 2005). Se utilizó una plantilla con un esquema de codificación compuesto de 3 grupos, cada uno de ellos coinciden con las 3 actividades del proceso técnico “Transición” del estándar ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE 12027, 2017). Para recolectar la información sobre las métricas utilizadas, se utilizó la codificación propuesta en las secciones 3.1 y 3.4.

En la Tabla 4.1, se presenta el resumen de los documentos analizados y las métricas relacionadas a cada una de las actividades del proceso de despliegue en el estudio de caso. La definición de las métricas se describe en la sección 3.4.

Tabla 4.1. Resumen de los documentos analizados para el estudio de caso y métricas.

Documentos analizados	Actividades del despliegue		
	Métricas para A1	Métricas para A2	Métricas para A3
Plan de trabajo	CAP HEP PAP PEP ET TP ESF		
Informe de avance		%DesvíoCap %DesvíoHE %DesvíoPEnt %AvanceE %AvancePA %DesvíoPA %DesvíoEsf	

Documentos analizados	Actividades del despliegue		
	Métricas para A1	Métricas para A2	Métricas para A3
		%AvanceTR %DesvíoTR %DensidadD	
Informe de cierre			%DesvíoCap %DesvíoHE %DesvíoPEnt %AvanceE %AvancePA %DesvíoPA %DesvíoEsf %AvanceTR %DesvíoTR %DensidadD
Casos de Prueba del despliegue		CantDef %DensidadD	

4.1.5. ANÁLISIS E INTERPRETACION DE LOS RESULTADOS

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

PI1: ¿Que mediciones se llevaron a cabo durante las actividades del proceso de despliegue de sistemas de software?

En base a la documentación analizada, se logró evidenciar falencias en el uso de métricas para las actividades del proceso de despliegue:

Actividad 1 (A1) - Preparación del despliegue. De acuerdo al plan de trabajo analizado, se pudo identificar que se cuenta con el detalle de tareas planificadas, así como el esfuerzo estimado para cada una de ellas, dando como resultado el esfuerzo total del proceso de despliegue.

Del análisis documental realizado también se pudo reconocer que se cuenta con los casos de test a ejecutar, los cuales incluyen las pruebas de aceptación.

Actividad 2 (A2) - Realización del despliegue. Según los informes de avance del proceso de despliegue, en éste se presenta el avance de las tareas ejecutadas. También se registró el esfuerzo consumido en cada una de las tareas que se fueron completando.

Durante el análisis documental se evidenciaron planes de prueba incompletos y pruebas de testing poco eficientes. Aun así, se realizaron cálculos de densidad de defectos basados en los defectos detectados resultado de las pruebas ejecutadas.

De este mismo análisis se pudo identificar que los técnicos (empleados del banco) no disponían de los conocimientos y aptitudes necesarias para llevar a cabo un despliegue de manera correcta, esto se observó en el despliegue de scripts y el informe de seguimiento proporcionado por la consultora. Esta situación provocó la necesidad de incorporar recursos técnicos de la consultora para cubrir la falta de estos conocimientos de los empleados del banco a quienes se debió capacitar para realizar los futuros despliegues.

Actividad 3 (A3) - Gestión de los resultados del despliegue. De acuerdo con el informe de cierre del proyecto de despliegue, la consultora debió afrontar inconvenientes con los repositorios de software (falta de permisos necesarios, versiones anteriores, falta de componentes, etc.). Además del bajo compromiso y desconocimiento de los técnicos del banco se generaron múltiples inconvenientes durante el despliegue. Estos inconvenientes técnicos, sumado a fallas en las funcionalidades de la aplicación, impactaron fuertemente en la satisfacción de los usuarios durante y una vez finalizado el despliegue.

En este informe también se evidenció que hubo un dimensionamiento erróneo de los entregables, la ejecución de tareas que no fueron planificadas, lo cual se vio reflejado en el valor final del esfuerzo real registrado y el cumplimiento de pruebas poco eficientes.

En la documentación analizada se identificó el uso de las métricas referidas al avance y desvíos de las tareas ejecutadas, los desvíos del esfuerzo y densidad de defectos, no obstante, las métricas referidas al entrenamiento realizado no se calcularon.

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

Se presentó un informe a la consultora en el cual se recomienda continuar con el uso de métricas con el propósito de fortalecer el proceso de despliegue. Estas métricas contribuyen a la toma temprana de decisiones y control del despliegue. Además, se sugirió el uso de las métricas para otros procesos del ciclo de vida de desarrollo de software.

La empresa encontró útil la aplicación de métricas porque le permitió comenzar a comprender y organizar el proceso de despliegue. Además, el tesista proporcionó un informe con un conjunto de recomendaciones para continuar con la mejora su proceso de despliegue para proyectos futuros, así como sugerencias de buenas prácticas de Ingeniería de Software en general. El conjunto de recomendaciones que se le presentaron a la PyME se detalla a continuación:

- Las mediciones de software son el mejor método de control y contribuyen a la toma temprana de decisiones.
- Se debe planificar y estimar cada una de las fases del ciclo de vida de desarrollo del software.
- Identificar las mediciones relevantes para cada fase del ciclo de vida de desarrollo del software.
- Elaborar el plan de pruebas de despliegue durante la fase de análisis y diseño. La metodología de pruebas de software estará relacionada a la que se esté utilizando para el ciclo de desarrollo o la gestión del proyecto.
- Identificar las capacidades necesarias para llevar adelante cada una de las actividades del despliegue.
- Identificar los riesgos del proceso para lograr una adecuada gestión de estos que permita evitar o disponer de procedimientos para su mitigación.

4.1.6. AMENAZAS A LA VALIDEZ

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

- **Validez de constructo.** Los resultados se obtuvieron en base al análisis documental realizado de la aplicación del conjunto de métricas propuestas 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 un Portal de Recursos Humanos realizado en una entidad bancaria de 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. En este caso se considera necesario informar sobre estos hallazgos, ya que sirve como un incentivo para que otros investigadores repitan nuestro estudio 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 la validación de un conjunto de métricas 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 métricas definidas 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 utilizaron las métricas propuestas; 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 las métricas.
- **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 1

Se presentaron los resultados de un estudio de caso para determinar la viabilidad de la aplicación del conjunto de métricas diseñado para el proceso de despliegue de sistemas de software en un entorno real. Este consistió en el análisis de las métricas utilizadas en el despliegue 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, la aplicación de las métricas propuestas, de las cuales se hizo uso de las métricas para determinar el desvío del

esfuerzo, avance y desvíos de las tareas planificadas y densidad de defectos. Esto nos permitió validar la pertinencia e idoneidad de las métricas propuestas.

- La segunda pregunta nos permitió diseñar un conjunto de métricas recomendadas (sección 3.4) para que la empresa implemente y mejore su proceso de despliegue, así como para introducir el uso de métricas para futuros despliegues de sistemas de software. Además, se sugirió el uso de métricas para otros procesos del ciclo de vida de desarrollo de software.

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

4.2. ESTUDIO DE CASO 2

4.2.1 DISEÑO DEL ESTUDIO DE CASO 2

En esta sección, se describe el estudio de caso, siguiendo los lineamientos propuestos de Runeson *et al.* (Runeson *et al.*, 2012). El objetivo principal del estudio de caso consiste en examinar la viabilidad de la aplicación del conjunto de métricas propuestas para el proceso de despliegue de sistemas de software en un entorno real con el propósito de refinarlas (si fuese necesario). Según la clasificación de Robson (Robson, 2002) se enmarca en los estudios exploratorios. Tras mantener una reunión con uno de los líderes de proyectos de una PyME de sistemas de la República Argentina a la cual se le proporcionó el conjunto de métricas propuestas (sección 3.4), se trabajó con documentación del despliegue de un Sistema de Gestión para una agencia publicitaria de Argentina realizado por la PyME.

4.2.2. PREGUNTAS DE INVESTIGACIÓN

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

PI1: ¿Que mediciones se llevaron a cabo durante las actividades del proceso de despliegue de sistemas de software?

A través de esta pregunta, se busca obtener información del uso de las métricas propuestas en la ejecución del proceso de despliegue y el uso dado por la consultora para validar su aplicabilidad.

PI2: ¿De qué manera se puede fortalecer el proceso de despliegue de sistemas de 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 un conjunto de métricas identificando su objetivo y la forma de cálculo.

4.2.3. CASO Y UNIDAD DE ANALISIS

En esta sección se describe el contexto, el caso y la unidad de análisis del estudio de caso. Según la clasificación de Yin (Yin, 2014), es un caso único holístico.

Contexto: el estudio de caso se realizó en una PyME Mediana tramo 1 de sistemas radicada en la Ciudad Autónoma de Buenos Aires, la misma cuenta con un plantel de 60 empleados. Esta empresa desarrolla sistemas de información a medida para clientes de diferentes rubros. Salvo sea a pedido del cliente, en sus proyectos de software combinan prácticas ágiles con metodologías de desarrollo con ciclo de vida iterativo. Se tuvo acceso a la documentación del despliegue del proyecto sujeto a un acuerdo de no revelar el nombre de la compañía, así como el compromiso de informar sobre los hallazgos y recomendaciones a considerar en la propuesta de métricas para el proceso de despliegue.

Caso: despliegue de un Sistema de Gestión para una agencia de publicidad de la República Argentina. Este consistió en llevar a cabo las siguientes tareas. Estas son:

- Instalación y puesta en marcha de un nuevo servidor en el centro de datos.
- Instalación del Motor de base de datos.
- Migración de la información del sistema predecesor al nuevo sistema.
- Instalación del sistema de software en los equipos de los usuarios.
- Configuración y parametrización del sistema, alertas y notificaciones.
- Capacitación a los usuarios del Sistema de Gestión.

Unidad de análisis: documentación del despliegue de un Sistema de Gestión.

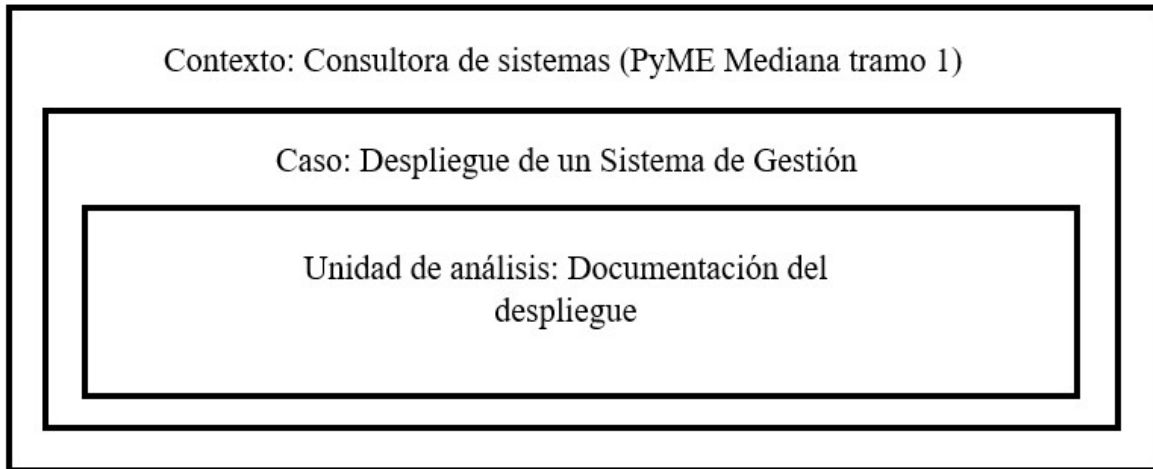


Figura 4.1. Clasificación de estudios de caso basada en la definición de Yin (Yin, R., 2014).

4.2.4. PREPARACION PARA LA RECOLECCIÓN DE DATOS

Se utilizó una técnica de tercer grado combinada con un método independiente según la clasificación propuesta en (Lethbridge *et al.*, 2005). Se utilizó una plantilla con un esquema de codificación compuesto de 3 grupos, cada uno de ellos coinciden con las 3 actividades del proceso técnico “Transición” del estándar ISO/IEC/IEEE 12207:2017 (ISO/IEC/IEEE 12027, 2017). Para recolectar la información sobre las métricas aplicadas, se utilizó la codificación propuesta en las secciones 3.1 y 3.4.

En la Tabla 4.2, se presenta el resumen de los documentos analizados y las métricas relacionadas a cada una de las actividades del proceso de despliegue en el estudio de caso. La definición de las métricas se describe en la sección 3.4.

Tabla 4.1 Resumen de los documentos analizados para el estudio de caso y métricas.

Documentos analizados	Actividades del despliegue		
	Métricas para A1	Métricas para A2	Métricas para A3
Plan de despliegue	CAP HEP PAP PEP ET TP ESF E VELH		
Informe de avance		%DesvíoCap %DesvíoHE %DesvíoPEnt %AvanceE	

		%AvancePA	
		%DesvíoPA	
		%DesvíoEsf	
		%AvanceTR	
		%DesvíoTR	
		%DensidadD	
		PRD	
		PRDT	
		CAPR	
Cierre del despliegue			%DesvíoCap %DesvíoHE %DesvíoPEÍnt %AvanceE %AvancePA %DesvíoPA %DesvíoEsf %AvanceTR %DesvíoTR%Den sidadD PRD PRDT VEL
Plan de Capacitación	CAP		
Casos de prueba	PA	PAE	PAE
Informe de ejecución de pruebas del despliegue		CantDef %DensidadD	

4.2.5. ANÁLISIS E INTERPRETACION DE LOS RESULTADOS

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

PI1: ¿Que mediciones se llevaron a cabo durante las actividades del proceso de despliegue de sistemas de software (identificación y análisis)?

En base al análisis documental realizado, se logró evidenciar falencias en el uso de métricas para las actividades del proceso de despliegue:

Actividad 1 (A1) - Preparación del despliegue. El plan de despliegue permitió evidenciar que cuentan con el valor que corresponde al esfuerzo que se dedicará al proceso y a las tareas que se llevarán a cabo durante el proceso de despliegue, las pruebas a realizar, ya que se cuenta con la definición de los casos de prueba a ejecutar y finalmente, se dispone de la planificación de la capacitación a realizar y la cantidad de usuarios a capacitar. También se encuentra planificado la cantidad de encuentros para dictar las capacitaciones.

Actividad 2 (A2) - Realización del despliegue. El informe de avance del proceso de despliegue expone el avance y desvío de las tareas ejecutadas, así como las replanificadas. Un ejemplo de esto último fue la reprogramación de algunas capacitaciones debido a la ausencia de personal de la aseguradora por encontrarse en un periodo vacacional.

El informe de ejecución de los casos de prueba evidenció que las pruebas de testing realizadas no se correspondían por completo con los casos de prueba diseñados, y el valor obtenido a través de la métrica densidad de defecto mostró un valor superior al esperado a causa de que el número de defectos fue mayor al supuesto.

Actividad 3 (A3) - Gestión de los resultados del despliegue. De acuerdo con el informe de cierre del proceso de despliegue, la empresa debió afrontar desvíos con las fechas pactadas para finalizar el despliegue, así como el registro del esfuerzo real denota un incremento en el esfuerzo insumido debido a errores en el software, lo cual también tuvo un impacto en el plan de capacitación debido a que el proceso de capacitación debió postergarse ya que, debido al corrimiento de fechas a causa de los defectos encontrados, el personal a capacitar entraba en un periodo vacacional. Este informe también evidenció que algunos de los errores detectados durante el despliegue del software se debían a las inconsistencias entre los ambientes de aseguramiento de calidad (en inglés *Quality Assurance* o QA) y producción.

De la documentación analizada se pudo identificar el uso de las métricas referidas al avance de las tareas ejecutadas, el esfuerzo y desvíos del proceso, así como la productividad de este. También se hizo uso de métricas como ser densidad de defectos y métricas relacionadas al entrenamiento realizado, como el avance de este, pero no hicieron uso de las métricas referidas al cumplimiento de las horas de entrenamiento ni de los usuarios que debían ser entrenados.

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

Se presentó un informe a la empresa en el cual se recomienda el uso de métricas con el propósito de fortalecer el proceso de despliegue. Estas métricas complementan a las que ya calculaban en proyectos anteriores y contribuyen al control y a la toma temprana de decisiones en el despliegue. Asimismo, se sugirió el uso de las métricas para otros procesos del ciclo de vida de desarrollo de software.

Además, el tesista proporcionó un informe con un conjunto de recomendaciones para continuar con la mejora de su proceso de despliegue para proyectos futuros, así como sugerencias de buenas

prácticas de Ingeniería de Software en general. El conjunto de recomendaciones que se le presentaron a la Aseguradora se detalla a continuación:

- Las mediciones de software son el mejor método de control y contribuyen a la toma temprana de decisiones.
- Definir de manera clara y precisa las actividades a realizar desde el inicio y el tiempo para desarrollar dichas actividades.
- Disponer de ambientes de calidad parecidos a los de producción, para que las pruebas realizadas en dicho ambiente sean lo más real posible y minimizar errores al momento realizar el despliegue en producción.
- Planificar y estimar para mostrar avances constantes que den valor al cliente.
- Identificar las mediciones relevantes para cada fase del ciclo de vida de desarrollo del software.
- Identificar los riesgos del proceso para lograr una adecuada gestión de los mismos que permita evitar o disponer de procedimientos para su mitigación.

4.2.6. AMENAZAS A LA VALIDEZ

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

- **Validez de constructo.** Los resultados se obtuvieron en base al análisis documental realizado y de la aplicación del conjunto de métricas 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 un Sistema de Gestión realizado en una agencia publicitaria de 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 realizó un estudio preliminar en (Vazquez *et al.*, 2021) y en este caso se considera necesario informar sobre estos hallazgos, ya que sirve como un incentivo para que otros investigadores repitan nuestro estudio 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 requería la validación de un conjunto de métricas 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 métricas definidas 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 utilizaron las métricas propuestas; 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 las métricas.
- **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 ESTUDIO DE CASO 2

Se presentaron los resultados de un estudio de caso para determinar la viabilidad de la aplicación del conjunto de métricas diseñados para el proceso de despliegue de sistemas de software en un entorno real. Este consistió en el análisis de las métricas aplicadas en el despliegue de un Sistema de Gestión realizado en una agencia publicitaria 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, que se logró calcular el esfuerzo estimado para las tareas a ejecutarse durante el proceso de despliegue, lo

cual permitió obtener el esfuerzo estimado total del mismo, también lograron aplicar métricas para determinar el desvío del esfuerzo y productividad del proceso, desvío y avance de las tareas planificadas y densidad de defectos. Esto nos permitió validar la pertinencia e idoneidad de las métricas propuestas.

- La segunda pregunta nos permitió diseñar un conjunto de métricas recomendadas (sección 3.4) para que la empresa implemente y mejore su proceso de despliegue, así como para introducir el uso de métricas para futuros despliegues de sistemas de software. Además, se propuso una serie de recomendaciones para mejorar su proceso, así como otros procesos del ciclo de vida de desarrollo de software.

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

CAPÍTULO 5.

CONCLUSIONES Y

TRABAJOS

FUTUROS

Este capítulo presenta las conclusiones obtenidas a través de la investigación desarrollada en esta tesis de Maestría y esboza el trabajo futuro. En la sección 5.1 se presentan las principales contribuciones de esta tesis. La sección 5.2 presenta la consecución de los objetivos de esta tesis. A continuación, en la sección 5.3 se introducen diversos aspectos que se plantean como trabajo futuro. Por último, en la sección 5.4 se enumeran las publicaciones obtenidas para difundir los resultados de esta tesis.

5.1. CONTRIBUCIONES

Esta tesis ha abordado la diseño y validación de un conjunto de métricas para el proceso de despliegue de sistemas de software. Por tanto, los principales aportes de esta tesis son:

- **Métricas.**

El conjunto de métricas diseñadas para el proceso de despliegue de sistemas de software le permita a las PyMES de Argentina fortalecer el despliegue. Estas métricas, le permitirán a las PyMEs llevar a cabo el seguimiento y control de la evolución del proceso para mejorar las actividades del despliegue. De esta manera, las PyMES cuentan con un proceso de despliegue maduro y esto finalmente terminará impactando en mejorar su competitividad en la industria del desarrollo de software. Para la construcción de las métricas, se siguió la visión tridimensional del proceso de despliegue definida en esta línea de investigación. Esta visión se compone de las dimensiones: “Proceso”, “Producto”, “Persona”. Para cada una de las dimensiones, 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. Finalmente, para cada actividad del mencionado estándar se realizó la propuesta de métricas según la clasificación propuesta por la norma ISO/IEC 9126 (ISO/IEC 9126-2, 2003; ISO/IEC 9126-3, 2003; ISO/IEC 9126-4, 2004). Las métricas diseñadas se evaluaron por diferentes profesionales de la industria del software para asegurar que fueran viables y útiles. La propuesta realizada también es útil para los investigadores interesados en ampliar la investigación presentada en esta tesis de Maestría. Las métricas fueron descriptas detalladamente en el Capítulo 3.

- **Conocimiento generado a través de los estudios empíricos.**

El conocimiento creado a través de los estudios empíricos realizados durante el desarrollo de la tesis de Maestría. El SMS permitió sistematizar el estado del arte respecto al uso de métricas en el despliegue de sistemas de software. Se logró identificar en los estudios, un punto en común que consiste, en la optimización del esfuerzo y el costo del proceso a través de la propuesta de modelos o frameworks. Además, se logró evidenciar, el uso de métricas con impacto en el cálculo del esfuerzo del proceso de despliegue de sistemas de software. El SMS se encuentra descrito detalladamente en el Capítulo 2.

Además, se llevaron a cabo dos estudios de casos, el primer estudio de caso (EC1), se llevó a cabo en una PyME Mediana de tramo 2 de desarrollo software de la República Argentina. El segundo estudio de caso (EC2), se desarrolló en una PyME Mediana tramo 1 de desarrollo de sistemas de la República Argentina. En ambos estudios de casos (EC1 y EC2) se pudo identificar el uso de las métricas propuestas lo cual permitió examinar la viabilidad de aplicación de la propuesta de métricas para refinarlas y completarlas de forma que las empresas PyMEs las adopten en futuros despliegues de sistemas de software. Los estudios de casos se describen en el Capítulo 4.

5.2. CONSECUSIÓN DE LOS OBJETIVOS DE LA TESIS

El objetivo general de esta tesis sirvió como punto de partida para definir los objetivos específicos de la investigación y proponer una solución para contribuir a la consecución de dicho objetivo general.

A continuación, se explica la consecución de los objetivos específicos de esta tesis de Maestría:

- O1: *Construir el estado del arte sobre métricas para el proceso de despliegue.*

Para lograr este objetivo se realizó un SMS que permitió identificar que los estudios centraban su propuesta en la optimización del esfuerzo y los costos del proceso de despliegue. Además, se logró confirmar la existencia de la propuesta de métricas para el proceso de despliegue. Estas propuestas tienen como limitación que se enfocan en el cálculo del esfuerzo del proceso y la mejora del proceso de automatización del despliegue. Las propuestas de métricas halladas en el SMS resultan insuficientes o inadecuadas para

las PyMEs quienes requieren mejorar la calidad de sus procesos para incrementar su competitividad dentro de la industria de desarrollo de software. El desarrollo del SMS se presenta detalladamente en el Capítulo 2.

- O2. *Diseñar un esquema de clasificación para las métricas a construir.*

Para lograr este objetivo se tuvo en cuenta la visión tridimensional del proceso de despliegue definida en esta línea de investigación (Proceso, Producto, Persona). Para cada una de las dimensiones 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. Y finalmente para cada una de las actividades del mencionado estándar, se adoptó la clasificación propuesta por la norma ISO/IEC 9126 (ISO/IEC 9126-2, 2003; ISO/IEC 9126-3, 2003; ISO/IEC 9126-4, 2004) la cual divide las métricas en tres categorías, según su naturaleza en: métricas básicas, métricas agregadas y métricas derivadas.

- O3. *Construir un conjunto de métricas para el proceso de despliegue de sistemas de software.*

Para lograr este objetivo y dado que los hallazgos del SMS confirmaron la vacancia del uso de métricas para el proceso de despliegue de sistemas de software, se diseñó un conjunto de métricas. Para su diseño se tuvo en cuenta la visión tridimensional del proceso de despliegue definida en esta línea de investigación, (Proceso, Producto, Persona). Además, para cada una de las dimensiones, se consideraron las actividades y tareas del proceso técnico “Transición” del estándar ISO/IEC/IEEE 12207 (ISO/IEC/IEEE, 2017). El conjunto de métricas diseñado y su descripción completa del modelo se encuentra en el Capítulo 3.

- O4. *Validar el conjunto de métricas en diferentes estudios de casos.*

Para cumplir con este objetivo se realizaron dos estudios de casos con el propósito de examinar la viabilidad de la aplicación de las métricas propuestas sobre un entorno real. El primer estudio de caso (EC1), se llevó a cabo en una PyME Mediana de tramo 2 de desarrollo software de la República Argentina en el cual se aplicaron algunas de las métricas propuestas durante el despliegue de entregas de funcionalidades de un Portal de Recursos Humanos de una entidad bancaria. En el segundo estudio de caso (EC2) se desarrolló en una PyME Mediana tramo 1 de desarrollo de sistemas de la República

Argentina donde también se aplicaron algunas de las métricas propuestas durante el despliegue de un Sistema de Gestión para una agencia publicitaria de Argentina.

En ambos casos, la aplicación de las métricas de parte de las PyMEs, permitió validar la aplicabilidad e idoneidad de las métricas propuestas de manera que las empresas PyMEs las adopten para futuros despliegues de sistemas de software.

5.3. TRABAJOS FUTUROS

Los resultados de la investigación presentados en esta tesis de Maestría han abierto una serie de trabajos futuros, tanto desde el punto de vista de la investigación como desde el punto de vista práctico, para permitir la transferencia de las métricas diseñadas hacia los profesionales de la industria del software. A continuación, se muestran algunas posibles líneas de trabajo futuras:

- Incrementar el número de estudios de casos, aplicando la solución a otros tipos de sistemas de información y recolectar información mediante una encuesta basadas en TAM (*Technology Acceptance Model*), con el objetivo de analizar el nivel de aceptación y de utilidad de la aplicación del conjunto de métricas diseñado para el proceso de despliegue de sistemas de software.
- Evaluar la viabilidad de poder aplicar las métricas en otros tipos de empresas, como por ejemplo empresas emergentes (Startups) o empresas de mayor tamaño para recolectar feedback por parte de los usuarios y contrastarlo con los resultados obtenidos en PyMES.
- Proponer nuevas métricas basadas en indicadores obtenidos a partir del diseño de encuestas de satisfacción.

5.4. PUBLICACIONES

En esta sección se presentan las publicaciones producidas durante el desarrollo de esta tesis de Maestría.

5.4.1. CONGRESOS NACIONALES

- Vázquez P., Marisa Panizzi M., Bertone R. (2018). Estimación del esfuerzo del proceso de implantación de software basada en el método de puntos de caso de uso. Mar del Plata, 29 y 30 de noviembre de 2018 Universidad Tecnológica Nacional. 6to. Congreso Nacional de

Ingeniería Informática / Sistemas de Información (CoNaIISI 2018). Simposio de Ingeniería de Sistemas y de Software. (En línea) ISSN: 2347-0372.

- Vazquez, P., Panizzi, M., (2017) “Implantación de Sistemas Informáticos – Identificación de Elementos del Proceso”, En: 5º Congreso Nacional de Ingeniería Informática / Sistemas de Información Ing. de Sistemas, Ing. de Software y Gestión de Proyectos (CONAIISI 2017), ISSN: 2347-0372, 2017

5.4.2. CONGRESOS INTERNACIONALES

- Vázquez P., Marisa Panizzi M., Bertone R. (2018). Estimación del esfuerzo del proceso de implantación de software basada en el método de Puntos de Caso de Uso. Concepción del Uruguay, Entre Ríos, Argentina, 25 y 26 de octubre de 2018 Universidad Tecnológica Nacional. VI SEMINARIO ARGENTINA - BRASIL DE TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN (SABTIC 2018). (En línea) ISSN: 2337-2970.
- Vazquez P., Panizzi M., Bertone R. (2021). Refinamiento de métricas para el proceso de despliegue de sistemas de software: Estudio de caso. Proceedings of the IADIS Ibero American Conference Applied Computing (CIACA 2021). Del 18 al 19 de noviembre de 2021. ISBN 978-989-8704-35-1.
- Vazquez P., Panizzi M. y Bertone R. (2021). Métricas para el proceso de despliegue de sistemas software: Un mapeo sistemático. Congreso Laninoamericano de Ingeniería (Ingenio). Desarrollo e Innovación en Ingeniería Vol. II, ed. 6. Zenodo. <https://doi.org/10.5281/zenodo.5513920> ISBN: 978-958-53278-6-3.

5.4.3. REVISTA

- Vázquez P., Marisa Panizzi M., Bertone R. (2019). Estimación del esfuerzo del proceso de implantación de software basada en el método de Puntos de Caso de Uso. Brazilian Journal of Development, Curitiba, v. 5, n. 3, mar. 2019. (En línea) ISSN: 2525-8761.

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 informáticos” (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 978-987-3714-82-5.

6. BIBLIOGRAFÍA

Agile Alliance. Agile Glosarry. <https://www.agilealliance.org>. Último acceso el 02/11/2021.

Agile Business Consortium. DSDM. <https://www.agilebusiness.org>. Último acceso el 02/11/2021.

Ambler S. (2016). The Agile Unified Process (AUP), <http://www.ambyssoft.com/unifiedprocess/agileUP.html>. Último acceso el 05/07/2017.

Arcangeli J., Boujbel R., & Leriche S. (2015). Automatic deployment of distributed software systems: Definitions and state of the art. *Journal of Systems and Software*, 103, pp. 198–218. doi:10.1016/j.jss.2015.01.040.

Basili V. (1993). “The Experimental Paradigm in Software Engineering”. En *Experimental Software Engineering Issues: Critical Assessment and Future Directions* (Ed. Rombach, H., Basili, V., Selby, R.). *Lecture Notes in Computer Science*, Vol. 706. ISBN 978-3-540-57092-9.

Basso D. (2014). Propuesta de Métricas para Proyectos de Explotación de Información. <http://sistemas.unla.edu.ar/sistemas/redisla/ReLAIS/relais-v2-n4-157-218.pdf>. Último acceso el 14/12/2017.

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. Último acceso el 02/12/2020.

Cámara de Software y Servicios Informáticos - CESSI. Sector SSI / OPSSI Coyuntura 2019-2020, <https://www.cessi.org.ar>. Último acceso el 02/12/2020.

Carrizo D., Alfaro A., (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Ingeniare. Revista chilena de ingeniería*, vol. 26 N° 1, 2018, pp. 114-129.

Carrizo D. & Sanchez L. (2017). Benchmarking to adopt an asap-based methodological guideline for software systems deployment. 30th IBIMA Conference: 8-9 November 2017, Madrid, Spain. ISBN: 978-0-9860419-9-0.

Chacón Luna A., Rodas Silva J., Vinueza Morales M. (2015). Estándares que contribuyen al desarrollo y entrega de productos de Software de Calidad. *Revista Ciencia UNEMI Vol. 8 - N° 13*, pp. 90 – 99, ISSN: 1390 – 4272.

- Charette R. (2005). Why software fails [software failure]. *IEEE spectrum*, 42(9), pp. 42-49.
- Creswell J. (2002). "Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research". Prentice Hall. ISBN 10: 01-3613-550-1.
- Cyrus K.M. and Nejad A.V. (2014). "The impact of national culture on the different phases of ERP implementation" *Proceedings of 44th International Conference on Computers and Industrial Engineering and IMSS 2014 - 9th International Symposium on Intelligent Manufacturing and Service Systems, Joint International Symposium on "The Social Impacts of Developments in Information, Manufacturing and Service Systems"*, pp. 1662-1671.
- 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.
- Eisty N. U., Thiruvathukal G. K., & Carver J. C. (2018). A Survey of Software Metric Use in Research Software Development. 2018 IEEE 14th International Conference on e-Science (e-Science). DOI:10.1109/escience.2018.00036.
- Erich F., Amrit C., Daneva M. (2017). "A Qualitative Study of DevOps Usage in Practice". *Software: Evolution and Process*, 29, pp. 1-20.
- Fenton N. & Bieman J. (2014). *Software metrics: a rigorous and practical approach*, CRC Press.
- Fernández F., Ángel M. (2015). Aplicación de técnicas de pruebas automáticas basadas en propiedades a los diferentes niveles de prueba del software. Recuperado de <http://hdl.handle.net/2183/14814>.
- Fuggetta A., & Di Nitto E. (2014). Software process. In *Proceedings of the on Future of Software Engineering* (pp. 1-12). DOI: doi.org/10.1145/2593882.2593883.
- Genero Bocco M., Lemus J. A. C., Velthuis M. G. P. (2014). *Métodos de investigación en ingeniería del software*. Ediciones de la U.
- Grenning, J. (2002). Planning poker or how to avoid analysis paralysis while release planning. *Hawthorn Woods: Renaissance Software Consulting*, 3, pp. 22–23. <https://mail.renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf>. Último acceso el 27/02/2022.

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.

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

ISO/IEC 9126-1. (2001). *Software engineering - Product quality - Part 1 Quality model*. <http://www.iso.org/iso/home.html>.

ISO/IEC 9126-2. (2003). *Software engineering - Product quality - Part 2 External metrics*. <http://www.iso.org/iso/home.html>.

ISO/IEC 9126-3. (2003). *Software engineering - Product quality - Part 3 Internal metrics*. <http://www.iso.org/iso/home.html>.

ISO/IEC 9126-4. (2004). *Software engineering - Product quality - Part 4 Quality in use metrics*. <http://www.iso.org/iso/home.html>.

Jacobson I., Booch G., Rumbaugh J. (1999). *The Unified Software Development Process*. 1 st Edition. Addison Wesley, USA.

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.

Karner, G. (1993). *Metrics for Objectory*, Degree thesis, Universidad de Linkoping, Suecia.

Kitchenham B., Budgen D., Brereton P. (2015). *Evidence-Based Software Engineering and Systematic Reviews*, Chapman and Hall 1 st. Editon. Chapman and Hall/CRC. USA.

Kitchenham B. y Chartes S. (2007). *Guidelines for performing systematic literature reviews in Software engineering*, Keele University, EBSE-2007-01.

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

López A., Sánchez A., Montejano G. (2016). Definición de Métricas de Calidad para Productos de Software. XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016, Entre

Ríos, Argentina). Red de Universidades con Carreras en Informática (RedUNCI). ISBN: 978-950-698-377-2. <http://sedici.unlp.edu.ar/handle/10915/53444>. Último acceso el 16/12/2017.

Mishra D., & Mishra A. (2009). Software Process Improvement in SMEs. *Computer Science and Information Systems*, pp. 111-140.

Morillas A. y Díaz, B. (2007). Qualitative answering surveys and soft computing. *Fuzzy economic review*, vol. XII, No. 1, pp. 3-19.

PAe, Métrica versión.3, Portal de Administración Electrónica, Gobierno de España, 2001, https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.WcvDsZry00. Último acceso el 30/09/2017.

Panizzi, M., Bertone R., Hossian A. (2017). Proceso de Implantación de Sistemas Informáticos – Identificación de vacancias en Metodologías Usuales. Libro de Actas de la V Conferencia Iberoamericana de Computación Aplicada CIACA 2017, pp. 207 -215. ISBN 978-989-8533-70-8. Vilamoura, Algarve, Portugal.

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.

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

Petersen K., Feldt R., Mujtaba S., Mattsson M. (2008). “Systematic mapping studies in software engineering”, in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, pp. 68–77.

Petersen K., Vakkalanka S., Kuzniarz L. (2015). “Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update”, in *Information and Software Technology*, 64, pp. 1-18.

Pressman R. (2015). *Software Engineering: A practitioners’ approach*. Mc Graw Hill, Eighth edition.

Puello J., San Feliu T. & Calvo-Manzano J. (2016). Una aproximación basada en metamodelado del área de proceso de Validación del CMMI: Un caso de estudio. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (17), pp. 26–40.

Rahman F., & Devanbu P. (2013). How, and why, process metrics are better. In 2013 35th International Conference on Software Engineering (ICSE), pp. 432-441. IEEE.

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.

Redrován Castillo F.F., Loja Mora N.M., Correa Elizaldes K.D. y Piña Orozco J.I. (2017). Estado del arte: Métricas de calidad para el desarrollo de aplicaciones web. 3C Tecnología: glosas de innovación aplicadas a la pyme, 6(4), pp. 1-12. DOI: <http://dx.doi.org/10.17993/3ctecno.2017.v6n4e24.1-12>.

Riveros H. y Rosas L. (1985). “El Método Científico Aplicado a las Ciencias Experimentales”. México: Editorial Trillas. ISBN 96-8243-893-4.

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

Rodríguez M., Piattini M. (2014). Entorno para la Evaluación y Certificación de la Calidad del Producto Software. XIX Jornadas de Ingeniería del Software y Bases de Datos JISBD'2014, pp. 163-176.

Ruiz G., Pena A., Castro C., Alaguna A. et al. (2006). Modelo de evaluación de calidad de software basado en lógica difusa, aplicada a métricas de usabilidad de acuerdo con la norma ISO/IEC 9126. Avances en Sistemas e Informática, vol. 3, No. 2, pp. 25-29.

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

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

Savor T., Douglas M., Gentili M., Williams L., Beck K., & Stumm M. (2016). Continuous deployment at Facebook and OANDA. Proceedings of the 38th International Conference on Software Engineering Companion - ICSE '16. DOI:10.1145/2889160.2889223.

Sábato J, Mackenzie M. (1982). “La Producción de Tecnología: Autónoma o Transnacional”. Instituto Latinoamericano de Estudios Transnacionales - Technology & Engineering. ISBN 9789684293489.

Trendowicz, A., y Jeffery, R. (2014). *Software project effort estimation: Foundations and best practice guidelines for success*. Springer International Publishing. Recuperado de [//www.springer.com/gp/book/9783319036281](http://www.springer.com/gp/book/9783319036281).

Tyndall J. B. (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.

Vazquez P., Panizzi M., Bertone R. (2021). Refinamiento de métricas para el proceso de despliegue de sistemas de software: Estudio de caso. *Proceedings of the IADIS Ibero American Conference Applied Computing (CIACA 2021)*. Del 18 al 19 de noviembre de 2021. ISBN 978-989-8704-35-1.

Vazquez P., Panizzi M., Bertone R. Anexo – Refinamiento de métricas para el proceso de despliegue de sistemas de software: estudio de caso. Disponible en: <https://doi.org/10.6084/m9.figshare.15044025.v1>.

Wohlin C. (2014). “Guidelines for snowballing in systematic literature studies and a replication in software engineering”, in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, pp. 1–10.

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.

Yim K. S. (2014). Norming to Performing: Failure Analysis and Deployment Automation of Big Data Software Developed by Highly Iterative Models. 2014 IEEE 25th International Symposium on Software Reliability Engineering. Doi:10.1109/issre.2014.31.

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