



Universidad Nacional
Sistema de Estudios de Posgrado
Maestría en Administración de Tecnología
de la Información

Fábricas de Software:
Un esquema de prestación de servicios de
TI para la construcción de software

Wilson Marín Calderón

Heredia, Costa Rica, Abril, 2019

Universidad Nacional

Facultad de Ciencias Exactas y Naturales

Escuela de Informática

Posgrado en Gestión de la Tecnología de Información y Comunicación (ProGesTIC)

**FORMULARIO DE DEPÓSITO LEGAL, AUTORIZACIÓN DE USO DE DERECHOS
PATRIMONIALES DE AUTOR E INCORPORACIÓN A REPOSITARIOS
INSTITUCIONALES DE INFORMACIÓN DE ACCESO PÚBLICO**

La persona abajo firmante, en condición de estudiante de la maestría

Wilson Hendrik Marín Calderón

y autor del Trabajo final de graduación titulado:

**Fábricas de Software: Un esquema de prestación de servicios de TI para la
construcción de software**

para optar al grado académico de Máster en:

**Administración de tecnologías de Información con énfasis en administración
de proyectos**

de conformidad con lo establecido en el documento de “Lineamientos generales para la realización del trabajo final de graduación” y demás normativa universitaria relacionada con estos trabajos de graduación, DECLARO BAJO FE DE JURAMENTO conociendo la responsabilidad civil, penal o administrativa en que podría incurrir al no decir la verdad, lo siguiente:

1. El documento, producto, obra audiovisual, software, resultado del trabajo final de graduación referido anteriormente es original, inédito y ha cumplido con todo el proceso de aprobación académico que confiere el grado académico postulado con esta obra.

2. El trabajo final de graduación referido anteriormente constituye una producción intelectual propia de la persona abajo firmante y a esta fecha no ha sido divulgado a terceros(as) de forma pública, por ningún medio de difusión impreso o digital.

3. Autorizo el depósito de un ejemplar en formato impreso y otro en formato digital (entregado en soporte de disco compacto), en la colección de trabajos finales de graduación del ProGesTIC de la Universidad Nacional, así como la realización de copias electrónicas adicionales para fines exclusivos de seguridad y conservación de la información.

4. En caso de que el trabajo final de graduación haya sido elaborado como obra en colaboración -bien se trate de obras en las que los autores(as) tienen el mismo grado de participación o aquellas en las que existe una persona autora principal y una o varias personas autoras secundarias-, todos(as) ellos(as) han contribuido intelectualmente en la elaboración del documento y en este acto, libero de responsabilidad a las autoridades del posgrado y a los funcionarios que custodian la colección del ProGesTIC, en relación con el reconocimiento que se realiza respecto de los niveles de participación asignados por el propio autor del proyecto.

5. En caso de que el trabajo final de graduación haya sido elaborado como obras en colaboración (conforme a lo dispuesto en el punto 4), el autor abajo firmante designa _____ a

como encargado(a) de recibir comunicaciones y representar con autoridad suficiente a los suscritos, en condición de agente autorizado(a) de los demás autores(as).

6. Reconozco que la colección de trabajos finales del ProGesTIC no emite criterios ni valoraciones académicas sobre lo planteado en el producto final del trabajo de graduación y autorizo a esta dependencia para que proceda a poner a disposición del público la obra en mención, a través de los espacios físicos o virtuales que se posea, así como a través del Repositorio Institucional; a partir del cual los usuarios de dichas plataformas puedan acceder al documento y hacer uso de este en el marco de los fines académicos, no lucrativos y de respeto a la integridad del contenido del mismo así como la mención del autor o poseedor de sus derechos.

7. Manifiesto que todos los datos de citas dentro de texto y sus respectivas referencias bibliográficas, así como las tablas y figuras (ilustraciones, fotografías, dibujos, mapas, esquemas u otros) tienen la fuente y el crédito debidamente identificados y se han respetado los derechos de autor.

8. Autorizo la licencia gratuita no exclusiva de los derechos patrimoniales de autor para reproducir, traducir, distribuir y poner a disposición pública en formato electrónico, el documento depositado, para fines académicos, no lucrativos y por plazo indefinido en favor de la Universidad Nacional, que incluye además los siguientes actos:

a. La publicación y reproducción íntegra de la obra o parte de esta, tanto por medios impresos como electrónicos, incluyendo Internet y cualquier otra tecnología conocida o por conocer.

b. La traducción a cualquier idioma o dialecto de la obra o parte de esta.

c. La adaptación de la obra a formatos de lectura, sonido, voz y cualquier otra representación o mecanismo técnico disponible, que posibilite su acceso para personas no videntes parcial o totalmente, o con alguna otra forma de capacidades especiales que le impida su acceso a la lectura convencional del proyecto.

c. La distribución y puesta a disposición de la obra al público, de tal forma que el público pueda tener acceso a ella desde el momento y lugar que cada quien elija, a través de los mecanismos físicos o electrónicos de que disponga.

d. Cualquier otra forma de utilización, proceso o sistema conocido o por conocerse que se relacione con las actividades y fines académicos a los cuales se vincula la maestría, la colección de trabajos finales del ProGesTIC, la Escuela de Informática y la Universidad Nacional.

9. Reconozco que la colección de trabajos del ProGesTIC manifiesta actuar con diligencia para evitar la existencia en su sitio web de contenidos ilícitos y en caso de que tenga conocimiento efectivo de la existencia de infracciones a los derechos de propiedad intelectual, se reserva el derecho de proceder a bloquear el acceso

durante el trámite del debido proceso para comprobar el incumplimiento y en caso de verificarse la falta, retirar definitivamente el acceso al proyecto depositado.

10. Acepto que la publicación y puesta a disposición del público del trabajo final de graduación, así como la presente autorización de uso de la obra, se regirá por la normativa institucional de la Universidad Nacional y la legislación de la República de Costa Rica. Adicionalmente, en caso de cualquier eventual diferencia de criterio o disputa futura, acepto que esta se dirimirá de acuerdo con los mecanismos de Resolución Alternativa de Conflictos y la Jurisdicción Costarricense.

Autor Wilson H. María Calderón

Firma: WHM

Fecha de entrega: 23/04/2019

Correo: otrowilson@gmail.com

Resumen Ejecutivo

El paso de la producción artesanal de bienes a la industrialización a través de fábricas fue un paso natural. Esta transición surge de la necesidad de reducir costos e incrementar las ganancias mediante la producción en masa, la estandarización y reutilización de procedimientos, herramientas y diseños.

Sin embargo, el desarrollo de software no ha podido dar el paso a la industrialización de la misma manera en que las fábricas tradicionales lo hicieron, aún continúa dependiendo del artesano programador. Por lo que costos del desarrollo de software siguen siendo elevados y la probabilidad de cometer errores que atentan contra el éxito del proyecto se incrementa.

Casi paradójicamente, una de las principales razones por la que la transición a hacia la fábrica de software no se completa es por la misma creencia de los gerentes, académicos y desarrolladores de que el proceso de desarrollo de software tiene una parte de arte y que cada desarrollo es único; que el trabajo de la ingeniería de software es aún visto como de artesano, confiando mayormente en personas sumamente hábiles.

Bajo esta premisa, se realizó una revisión de literatura en 50 publicaciones, desde 1998 a 2018, sobre los componentes de fábricas de software (FS). El propósito fue desmitificar esa creencia a través de la materialización de los elementos que constituyen una FS, presentes en implementaciones documentadas.

El resultado fue un total de 64 elementos, agrupados en 11 categorías que muestran un panorama del conocimiento que se requiere para alcanzar un modelo funcional de Fabrica Software. Adicionalmente a estos elementos, se presenta un modelo de relaciones entre categorías y una propuesta de fábrica de software con procesos tomados de las publicaciones revisadas.

Índices

Contenido

1	Capítulo I: Introducción.....	11
1.1	Situación de Investigación	11
1.2	Justificación.....	13
1.3	Objetivos	13
1.3.1	Objetivo General.....	13
1.3.2	Objetivos Específicos.....	14
1.3.3	Metas	14
2	Capítulo II: Marco Teórico.....	15
2.1	Marco Conceptual	15
3	Capítulo III: Marco Metodológico.....	24
3.1	Paradigma de Investigación.....	24
3.1.1	Enfoque de Investigación.....	25
3.1.2	Tipo de Investigación	26
3.2	Sujetos y fuentes de información.....	28
3.3	Población, Muestra o espacio de investigación.....	29
3.3.1	Estrategia de búsqueda.....	29
3.3.2	Criterio de Inclusión.....	33
3.3.3	Criterio de Exclusión	34
3.3.4	Procedimiento de Selección	34
3.3.5	Evaluación de calidad.....	35
3.4	Definición de variables.....	36
3.5	Descripción de extracción de datos.....	37
3.5.1	Estrategia de Extracción.....	38
3.6	Instrumentos para la recolección de información.....	38
3.7	Proceso e instrumentos de tabulación de los datos	39
3.7.1	Síntesis de Datos.....	39
4	Capítulo IV: Análisis de resultados	41

4.1	Investigaciones Relevantes	41
4.2	Publicaciones primarias.....	41
4.3	Datos extraídos y Evaluación de calidad.....	42
4.4	Síntesis de Datos.....	43
4.5	Modelos de Fabrica de software	47
4.6	Modelo de Fabrica de software	49
4.7	Revisión de objetivos	64
5	Capítulo V: Conclusiones y Recomendaciones.....	67
5.1	Conclusiones.....	67
5.2	Recomendaciones.....	69
6	Capítulo VI: Análisis retrospectivo	71
7	Bibliografía.....	73
8	Anexos	78
8.1	Detalle de resultados de búsqueda:	78
8.2	Detalle de publicaciones primarias.....	81
8.3	Detalle de calificaciones de calidad.....	103
8.4	Detalle de datos recolectados	106
8.5	Total de los elementos de fábrica de software encontrados	117
8.6	Detalle de distribución de modelos de fábrica de software.....	121
8.7	Modelo de Componentes de Fabrica de Software.....	122
8.8	Bibliometría.....	123

Índice de Tablas

Tabla 1 : Metas de la Investigación	14
Tabla 2: Preguntas frecuentes sobre Ingeniería de Sistemas	18
Tabla 3: Estructura de Revisión Sistemática de Literatura	25
Tabla 4: Taxonomía de Cooper para una Revisión de Literatura	27
Tabla 5: Formato para registro de resultados de búsqueda.....	33
Tabla 6: Tabla de Registro de publicaciones primarias	35
Tabla 7: Variables de la investigación.....	37
Tabla 8: Evaluación de Calidad y Recolección de Datos.....	39
Tabla 9: Tabla de síntesis de datos.....	40
Tabla 10: Condensado de Calidad	42
Tabla 11: Cantidad de Elementos por Categoría	44
Tabla 12: Cantidad de Elementos por Tipo de componente	44
Tabla 13: Componentes más comunes en fábricas de software	45
Tabla 14: Modelos de Fábricas de Software:	47
Tabla 15: Objetivos, Preguntas y Metas	64
Tabla 16: Objetivos, Actividades, Instrumentos y Metas	65
Tabla 17: Validación de Objetivos	66
Tabla 18: Detalle de resultado de búsqueda	78
Tabla 19: Detalle de publicaciones primarias	81
Tabla 20: Detalle de evaluación de calidad.....	103
Tabla 21: Detalle de datos recolectados	106
Tabla 22: Elementos de fábrica de software encontrados.....	117
Tabla 23: Modelos de fábrica de software encontrados.....	121
Tabla 24: Cantidad de Publicaciones Excluidas.....	123
Tabla 25: Cantidad de Publicaciones por Base de Datos.....	124
Tabla 26: Cantidad de Publicaciones por Idioma.....	125
Tabla 27: Cantidad de Publicaciones por Año (1998-2018).....	125

Índice de Ilustraciones

Ilustración 1: Historia de los CMMI	20
Ilustración 2: Motor de búsqueda Primo	31
Ilustración 3: Búsqueda Avanzada de Google Académico	32
Ilustración 4: Elementos de Planificación	50
Ilustración 5: Elementos de Requerimientos.....	51
Ilustración 6: Elementos de Diseño	52
Ilustración 7: Elementos de Pruebas.....	53
Ilustración 8: Elementos de la Gestión del Conocimiento.....	54
Ilustración 9: Elementos de Desarrollo	55
Ilustración 10: Elementos de Mejora continúa.....	56
Ilustración 11: Elementos de Calidad.....	57
Ilustración 12: Elementos de Productividad	58
Ilustración 13: Elementos de Seguridad	59
Ilustración 14: Elementos de Operación y Mantenimiento.....	59
Ilustración 15: Elementos de Servicios	60
Ilustración 16: Modelo de relaciones entre categorías	61
Ilustración 17: Modelo de procesos de Fábrica de software	63
Ilustración 18: Modelo de elementos de Fábrica de Software	122

1 Capítulo I: Introducción

1.1 Situación de Investigación

El desarrollo de software ha sufrido una transformación desde sus inicios allá por los años 50. El conocimiento acumulado sobre la tecnología y los métodos de desarrollo han hecho que el proceso de la construcción de software se refine al punto que cada vez más se parece a la fabricación de productos en el sentido de que integra etapas bien definidas de planeación, diseño, construcción, pruebas y mantenimiento (Cusumano, 1987b).

Las compañías desarrolladoras de productos usualmente se enfrentan un dilema: satisfacer a los clientes en su demanda de una mayor variedad de productos o la búsqueda de reducción de costos y complejidad a través de la estandarización de herramientas, componentes y procedimientos (Cusumano, 1987). Los “talleres” (Job-shops) de productos usualmente se consideran más flexibles para ofrecer variedad, pero a un alto costo por no tener economía de escala, mientras que, por otro lado, manufacturas de producción en masa sacan mayor ganancia por unidad a cambio de reducida flexibilidad en términos de introducción de nuevos productos (Cusumano, 1987).

El movimiento desde los talleres hacia las fábricas de producción en masa es un proceso que sucede en mayor medida gracias a la iniciativa de los gerentes que utilizan el conocimiento acumulado sobre los productos y los procesos para estandarizar e integrar los procesos de producción de forma centralizada, desarrollar componentes intercambiables, dividir las labores, coordinar procesos, automatizar tareas y establecer controles (Cusumano, 1987).

En el desarrollo de software, la transición hacia la producción en masa ha generado avances tecnológicos como la programación orientada a objetos (POO) y la arquitectura orientada a servicios (SOA, por sus siglas en inglés) en términos de reutilización y optimización de los esfuerzos de desarrollo de software. Sin embargo,

la transición desde los “talleres de software” hacia las “fábricas de software” se encuentra aún en progreso (Stojanovski & Dzekov, 2009).

Uno de los principales problemas que impide el total movimiento hacia la fábrica de software (FS) es la idea que comparten gerentes, programadores y académicos de que el desarrollo de software tiene una parte de arte y que cada desarrollo es único; que el trabajo de la ingeniería de software aun es visto como de artesano, confiando mayormente en personas sumamente hábiles (Cusumano, 1987) (Stojanovski & Dzekov, 2009).

Los resultados del reporte CHAOS de 2015 publicado por un grupo de investigadores en tecnologías de información (The Standish Group International, Inc, 2015), parece apoyar esa conceptualización del problema. El reporte demuestra poco progreso en los proyectos de software desde el 2011. Donde solo el 29% del total evaluado es catalogado como exitoso, el 52% como desafiantes y 19% fallidos. Proyectos pequeños tienen una tasa de éxito del 61% mientras que los grandes y muy grandes tienen tasas de 11% y 6 % respectivamente. Particularmente llamativo es el hecho de que los proyectos catalogados como al por menor (“*retail*” en inglés), tiene la mayor tasa de éxito, con un 35% con respecto a otras categorías de industrias de mayor tamaño.

Basados en los datos anteriores, surge la inquietud de entrar en área de conocimiento de las fábricas de software para plasmar de manera concisa los componentes más comunes, de manera que permita el aprovechamiento de las ventajas de la estandarización de procesos, herramientas y procedimientos.

1.2 Justificación

Las fábricas de software (SF), como concepto, pretende sacar ventaja de la experiencia obtenida de las fabricas tradicionales en la creación de productos al aplicarla a la creación de software (Montequín, Pérez, Fernández , & Balsera, 2013). Algunas de estas ventajas son: incremento de la productividad, mayor rapidez de despliegue (*time to market*), una abstracción más alta, mejora en la reutilización y en la calidad. Algunos estudios han encontrado una mejora en el rendimiento de poco más del 30% en promedio, mencionan (Krajnc, Andrej, Gerlec, Goljat, & Polančič, 2012) citando a otros autores.

La investigación tiene como objetivo, describir los componentes de la fábrica de software de manera puntual, presentes en la industria para aprovechar las ventajas mencionadas anterior mente y las economías de gama lograda “*a través del manejo de múltiples diseños o la ingeniería de proyectos sistemática, en lugar de tratar cada proyecto como único.*” (Cusumano, 1991) . Como expectativa secundaria, se espera que sirva como un de material referencia, condensado y moderno para futuras investigaciones, de manera que colabore con el cambio de pensamiento popular de que el desarrollo de software es más un arte que una ciencia mencionado anteriormente.

1.3 Objetivos

1.3.1 Objetivo General

- Modelar los componentes más comunes de fábrica de software para prestación de servicios de TI en la construcción de software que sirva como material de referencia condensada y moderna, a través de una revisión de literatura.

1.3.2 Objetivos Específicos

- Identificar los elementos del esquema de fábrica de software más usados en la industria y sus variantes para enmarcar el concepto de fábrica de software
- Generar una clasificación de los elementos del esquema de fábrica de software que permita agruparlos según su relación
- Sintetizar un modelo de prestación de servicios de desarrollo de software basado en fábrica de software.

1.3.3 Metas

A continuación, se presenta una tabla asociativa que describe las metas de la investigación correspondientes a cada objetivo específico.

Tabla 1 : Metas de la Investigación

Objetivo específico	Metas
Identificar los elementos del esquema de fábrica de software más usados en la industria y sus variantes para enmarcar el concepto de fábrica de software	-Obtener los componentes presentes en modelos de fábrica de software
Generar una clasificación de los elementos del esquema de fábrica de software que permita agruparlos según su relación	-Organizar los componentes a través de una abstracción
Sintetizar un modelo de prestación de servicios de desarrollo de software basado en fábrica de software que concilie el funcionamiento de las fábricas de software con las necesidades de los clientes	-Crear un diagrama de fábrica de software -Descripción de los elementos del diagrama

Fuente: Elaboración propia

2 Capítulo II: Marco Teórico

2.1 Marco Conceptual

A continuación, se presenta la terminología de referencia con la definición de los términos más relevantes para el dominio popular.

Revisión de literatura es un procedimiento con múltiples propósitos. Justus J. Randolph (Randolph, 2009) menciona que puede servir como medio para demostrar conocimiento sobre un tema, comprendiendo aspectos como vocabulario, teorías, variables, métodos e historia. El procedimiento incluye identificar, localizar, sintetizar y analizar la literatura conceptual, así como como reportes de investigación, artículos, conferencias, libros tesis u otro material relacionado al tema de investigación (Williamson & Johanson, 2018).

La revisión sistemática de literatura en la ingeniería de software tiene sus orígenes en los trabajos de Kitchenham, y posteriormente de Dybå y Jørgensen, basados en revisiones preexistentes de otras áreas como los estándares médicos y revisiones sistemáticas propuestas por la sociología (Kitchenham & Brereton, 2013).

Tecnologías de Información, se refiere a un subconjunto o marco de trabajo dentro del área sistemas de información que comprende a los componentes tecnológicos específicamente, como comunicaciones, redes, realizar transacciones o gobernar la información, entre otros (Cornford & Shaikh, 2013). De forma más condensada (Owusu-Nyamekye, 2016) lo define como “(1) un componente de los Sistemas de Información que se encarga de las computadoras y las telecomunicaciones con el objetivo de recuperar, almacenar y transmitir información; y (2) como la parte de los Sistemas de Información que consiste de la red de todos los canales de comunicación usados dentro de la organización”.

Fábrica de software (SF) (*“Software Factory”* en inglés) fue acuñado como término por primera vez en el evento “NATO Science Conference” por R.W. Bemer de General Electric (Li, Li, & Li, 2001) de 1968 bajo las ideas de desarrollar procedimientos, herramientas y componentes de programas estandarizados (como las fábricas de producción en masa) para la creación de software; ideas que fueron criticadas por los asistentes debido a las limitaciones de la época (Cusumano, 1987). Una definición más elaborada sería: una línea de producción de software con herramientas, procesos y activos configurables, que pueden automatizar el desarrollo y mantenimiento de un tipo de software y sus variantes al adaptar, ensamblar y configurar elementos de un marco de trabajo (Greenfield, Short, Cook, & Kent, 2004). Los autores (Muños & Pelechano, 2005) lo ponen en pocas palabras diciendo que “...las Fábricas de Software se enfocan en el desarrollo de sistemas similares promoviendo la reutilización de arquitecturas, componentes y conocimiento (know-how)”. Los primeros intentos de fábricas de software se originaron en Japón, siendo Hitachi la primera compañía en el mundo en implementar una fábrica de software (Software Works) (Cusumano, 1987b).

Línea de Producción de software (*software product line, SPL*). Las líneas de producción son un concepto proveniente de la industria de manufacturas, que las utilizan para para satisfacer un segmento de mercado o propósito particular. Las líneas de producción de software, en cambio, son un concepto nuevo de rápida popularidad que apunta a convertirse en un paradigma de desarrollo de software. Se define como un conjunto de sistemas de software que comparten un conjunto de características que satisfacen necesidades específicas de un segmento de mercado en particular, contruidos a partir de un grupo esencial de activos (Northrop & Clements, 2012)

Ingeniería de software, es un concepto que fue mencionado por primera vez en la conferencia “*NATO Working Conference on Software Engineering*”, en 1968, llevada a cabo para discutir lo que en aquel entonces se conoció como “La crisis del software” generada por la introducción de circuitos integrados años antes, cuyo

poder habilitaba la construcción de sistemas de software inimaginable en términos de magnitud y complejidad (Sommerville, 2007) (Borg, 2012).

El desarrollo de estos nuevos sistemas grandes y complejos causó incrementos en los costos, atrasos e inestabilidad, así como dificultades para su mantenimiento. Ante esta situación nuevas técnicas y procedimientos fueron diseñados para controlar el desarrollo de esta forma de software y se convirtieron en parte de lo que hoy se llama ingeniería de software (Sommerville, 2007). Así fue como en 1969 se acuñó el término Ingeniería de Software para “denotar el desarrollo de software de acuerdo a un enfoque sistemático y organizado, apuntando a producir software de alta calidad con incertidumbre reducida” (Borg, 2012).

La IEEE posteriormente lo define como la aplicación de un enfoque sistemático, disciplinado, cuantificable al desarrollo, operación y mantenimiento del software, lo que es, la aplicación de ingeniería al software. (IEEE Standards Board, 1990). Mientras que (Laplante, 2007) lo describe como “un enfoque sistemático al análisis, al diseño, evaluación, implementación, pruebas, mantenimiento y reingeniería del software...” donde diferentes modelos para el ciclo de vida del software se definen, así como metodologías de especificación y evaluación. Esta última es la definición usada en el presente documento.

En la siguiente tabla, se presenta un extracto de las preguntas frecuentes sobre ingeniería de software elaborado por (Sommerville, 2007):

Tabla 2: Preguntas frecuentes sobre Ingeniería de Sistemas

Pregunta	Respuesta
¿Qué es el software?	Programas informáticos y documentación asociada. los productos de software pueden desarrollarse para un cliente en particular o desarrollarse para un mercado general
¿Qué es la ingeniería de software?	La ingeniería de software es una disciplina de ingeniería que se ocupa de todos los aspectos de la producción de software.
¿Cuál es la diferencia entre la ingeniería de software y la informática (computer science)?	La informática se ocupa de la teoría y los fundamentos; la ingeniería de software se ocupa de los aspectos prácticos de desarrollar y entregar software útil.
¿Cuál es la diferencia entre la ingeniería de software y la ingeniería de sistemas?	La ingeniería de sistemas se ocupa de todos los aspectos del desarrollo de sistemas informáticos, incluidos el hardware, el software y la ingeniería de procesos. La ingeniería de software es parte de este proceso.
¿Cuáles son los costos de la ingeniería de software?	Aproximadamente el 60% de los costos son costos de desarrollo, el 40% son costos de prueba. Para el software personalizado, los costos de evolución a menudo exceden los costos de desarrollo.
¿Qué son los métodos de ingeniería de software?	Enfoques estructurados para el desarrollo de software que incluyen modelos de sistema, notaciones, reglas, consejos de diseño y guía de procesos.

Fuente: (Sommerville, 2007)

Economía de escala, algunas veces llamada “economía de tamaño” se refiere a la reducción del promedio del costo por unidad a medida que la producción se incrementa (DeLuca , 2012). En la producción de software, así como en los productos, aparece cuando una implementación inicial o un prototipo diseñados por los ingenieros se reproduce mecánicamente (Greenfield & Short, 2003).

Economía de gama, usualmente llamada “costo complementario” consiste en la reducción de costos al ofrecer una variedad de productos a través de una sola compañía en lugar de varias especializadas. Se puede identificar una economía de gama cuando el consumo conjunto resulta más beneficioso o cuando se ofrecen múltiples productos en un mismo lugar (Barakat, 2003, pág. 24). En el software, se presenta en “la producción de múltiples diseños y su implementación inicial, cuando los mismos estilo, patrones y diseños se usan para crear múltiples diseños similares y los mismos leguajes, bibliotecas y herramientas son usados para la implementación inicial” (Greenfield & Short, 2003).

CMMI (*Capability Maturity Model Integration* o integración de modelos de madurez de capacidades) es una colección de buenas prácticas elaborado por un grupo de expertos en mejora, constituido por miembros del gobierno estadounidense, la industria y el SEI (*Software Engineering Institute* o Instituto de ingeniería de software), con objetivo de mejorar el modelo preexistente SW-CMM (*software capability maturity model* o modelo de madures de capacidad del software), lazado en 1991. Esta mejora consistió en expandir el conjunto de buenas a prácticas a otras áreas de la organización (CMMI Institute, 2013) . Actualmente CMMI cuenta con tres enfoques: CMMI Desarrollo, CMMI Adquisiciones y CMMI Servicios, todos en su versión 1.3. En la Ilustración 1 se puede observar el origen y la evolución CMMI hasta el día de hoy:

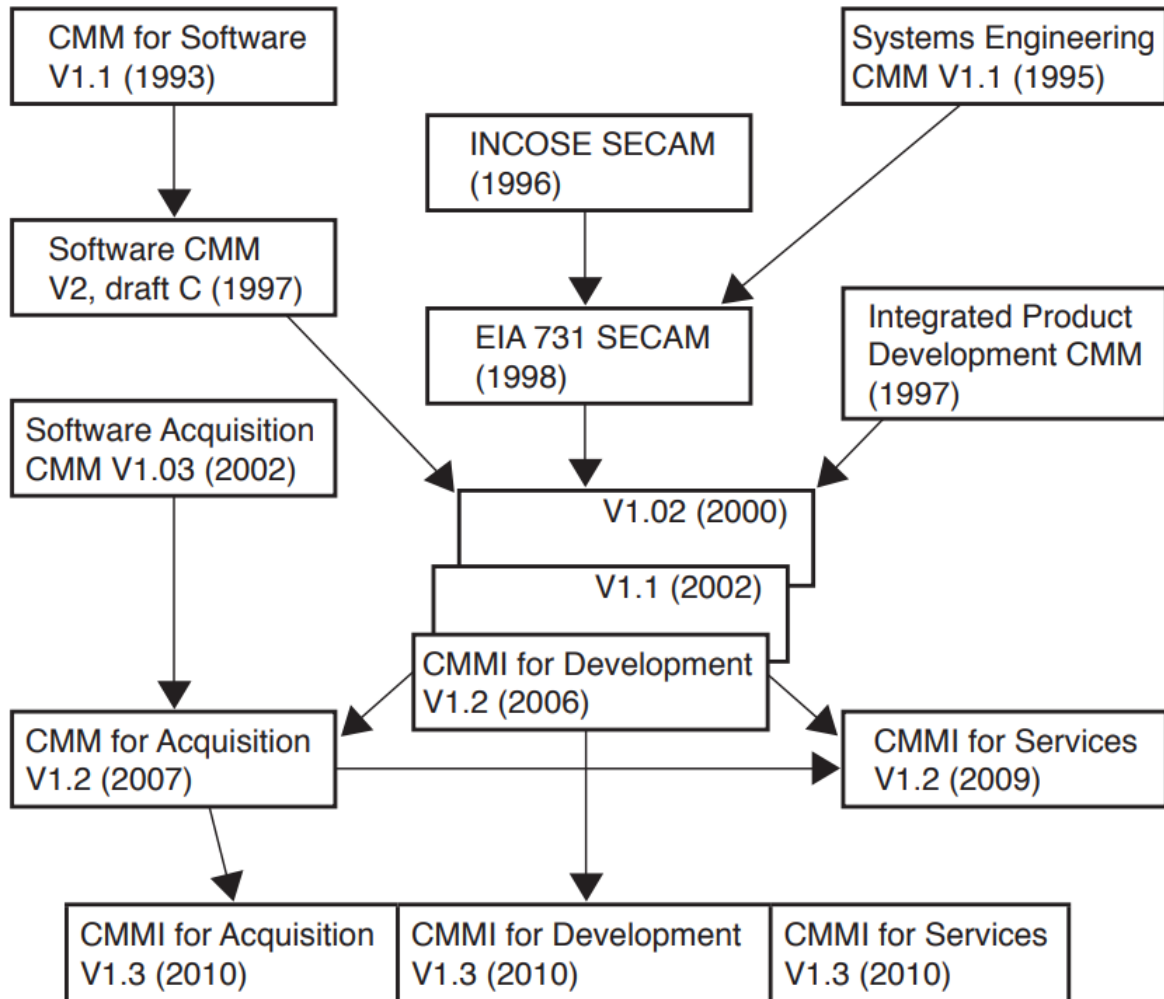


Ilustración 1: Historia de los CMMI

Fuente: (CMMI Institute, 2013, pág. 11)

Modelo y Meta Modelo. metamodelo puede ser considerado como la descripción de los conceptos usados para la construcción de modelos. Este se encuentra en un nivel de abstracción más arriba del modelo, de manera que, si se tiene un modelo “M”, el modelo usado para crear “M” es llamado metamodelo (Pop, 2005). En Ingeniería Dirigida por Modelos es tipificado y limitado por un lenguaje estándar (Lajmi, Cauvin, & Ziane, 2009), por ejemplo, el Eclipse Modeling Framework de Ecore.

Modelo, es definido de múltiples formas, aunque la Real Academia Española (RAE) lo define como “Arquetipo o punto de referencia para imitarlo o reproducirlo”. En

Ingeniería de software se utiliza para abstraer un conjunto de objetos o fenómeno y pueden ser de dos tipos, descriptivos o prescriptivos, donde los primeros son reflejo del fenómeno original mientras que los segundos se usan para especificar algo que va a ser creado (Staron, 2003). Similarmente (Pop, 2005) lo define como como “una abstracción de un sistema que refleja parte o todas las características de interés”.

En Ingeniería Dirigida por Modelos, modelo es un gráfico con objetos tipificados y delimitados por un metamodelo (Lajmi, Cauvin, & Ziane, 2009).

DSL (*Domain Specific Language*) o Lenguaje específico de dominio, es un lenguaje, usualmente pequeño y declarativo, que permite expresar problemas en un determinado dominio (como venta de libros, atención a pacientes, productos financieros, dibujo, diseño digital de circuitos), usando notaciones y abstracciones específicas para ese dominio. Algunos ejemplos populares de DSL son SQL, HTML, TeX (Brotto, 2006).

MDD (*Model Driven Development*) o desarrollo dirigido por modelos consiste en la utilización de los modelos utilizados en la especificación de sistemas de software para la generación automática de aplicaciones, a través de técnicas de generación de código y plantillas. De esta manera el desarrollo de software puede ser automatizado y la calidad puede ser estandarizada debido a que el código es generado conforme a cierta estructura, esquema o reglas; MDA y las fábricas de software representan dos modelos de este enfoque (Demir, 2006)

MDA (*Model Driven Architecture*) o arquitectura basa en modelos, es una solución propuesta por OMG (*Object Management Group*) que, permite el desarrollo de aplicaciones y la elaboración de especificaciones basados en una plataforma independiente de la tecnología en la cual será implementada (Object Management Group, 2019). MDA utiliza UML (también propiedad de OMG) para la definición de del modelo que se requiere implementar. Posteriormente ese modelo es transformado en código en tres pasos generales: creación de una Modelo Computacional Independiente, creación de un Modelo Independiente de Plataforma y creación de una Modelo Dependiente de Plataforma que convierte en código el modelo (Demir, 2006).

ISO es la Organización Internacional de Estándares, fue fundada en 1947 y ha publicado un total de 22568 estándares internacionales, cubriendo casi todos los aspectos de la tecnología y los negocios. Los estándares son elaborados por miembros expertos alrededor del mundo que los requieren, y establecidos por medio de un proceso de consenso (ISO, 2019).

DevOps es el acrónimo de “*Development*” y “*Operation*” (Desarrollo y Operación de software) y consiste en integrar las tareas de desarrollo, entrega y operación de software que usualmente están separadas. Con este enfoque se pretende tener equipos conformados por personas con diferentes funciones trabajando en el desarrollo y entrega continua de características nuevas de funciones operativas (Ebert, Gallardo, Hernantes, & Serrano, 2016).

KM (*Knowledge Management*) o Gestión del Conocimiento, tiene definiciones variadas entre los autores debido a que es un campo multidisciplinario, pero la definición de North, K. y Kumta, G. listada en la recopilación de definiciones de (Alwazae, 2015), incluye varios sujetos, acciones y propósitos:

“La gestión del conocimiento permite que individuos, equipos y organizaciones completas, así como redes, regiones y naciones, creen, compartan y apliquen de manera colectiva y sistemática el conocimiento para lograr sus objetivos estratégicos y operativos.”

Varios estudios citados en (Alavi, Smith, & Leidner, 1999) sobre la gestión del conocimiento coinciden en cuatro grandes procesos: el proceso crear, actualizar y mantener, el proceso de almacenar, el proceso de transferir (compartir) y el proceso de aplicar el conocimiento. Un estudio de la Universidad de Cranfiel, mencionado por los mismos autores, define los siguientes diez procesos para la gestión de conocimiento, y se consideró el más apto para la revisión de literatura:

“...crear nuevo conocimiento, encontrarlo internamente, adquirir conocimiento externamente, poseer el conocimiento, procesarlo, reutilizarlo, aplicar el conocimiento en algún beneficio, actualizarlo, compartirlo internamente y compartirlo fuera de la organización.”

CBSE (*Component Based Software Engineering*) o Ingeniería de Software Basado en Componentes consiste en el desarrollo de software a través de la utilización de componentes de software preexistentes, el desarrollo de componentes como entidades reusables y el intercambio de componentes para mejorar y personalizar los sistemas (Szyperski, Gruntz, & Murer, 2002).

En (Vale, et al., 2016), los autores realizaron una revisión de literatura de lo publicado entre 1984 a 2012 sobre CBSE. Exponen que CBSE es parte importante en el incremento de la productividad, ahorro de costos, mejora de la calidad, incremento de la reutilización, reducción de complejidad, entre otros. Además, es participe en múltiples dominios como SPL y MDD, sistemas empotrados, sistemas operativos, desarrollo de software basado en aspectos, sistemas orientados a servicios, etc., y que en la mayoría de los casos (alrededor de 68%, 820 publicaciones) es usado como componente esencial de una solución.

PSP/TSP (Personal Software Process / Team Software Process) o Proceso de Software Personal / Proceso de Software Equipo es un marco de trabajo para la mejora de la calidad de los productos de los ingenieros o grupos de ingenieros a través de la planificación y el seguimiento de su trabajo. El propósito es lograr compromisos que se puedan alcanzar respaldado por datos y la gestión de la calidad. Fue elaborado a partir de CMM en 1989 por Watts S. Humphrey. (Humphrey, Ferguson, Khajenoori, Macke, & Matvya, 1997). Desde la perspectiva del creador, POS mejora las habilidades en un enfoque a nivel personal, TSP mejora el desempeño de los equipos mediante un enfoque hacia el equipo y el producto, mientras que CMM mejora las capacidades de la organización mediante un enfoque de gestión (Humphrey, 2000).

3 Capítulo III: Marco Metodológico

En este apartado se describe el marco metodológico conformado por la naturaleza de la revisión de literatura, el proceso e instrumentos de selección de información, los sujetos de estudio y otros componentes de que guían la trayectoria de la revisión de literatura.

3.1 Paradigma de Investigación

Existe un número de paradigmas o marcos de trabajo interpretativos de investigación que definen las premisas epistemológicas del investigador, paradigmas como: positivismo, post-positivismo, interpretativo, constructivista, transformativo y pragmático, entre otros (Creswell, 2014) (Bekmamedova, y otros, 2018). La visión del Pragmatismo indica que no hay criterio filosófico mejor que otro y que de hecho practicar solo uno de ellos podría limitar la investigación, más bien se enfoca en la utilización de las herramientas que le permita interpretar la realidad del momento y las circunstancias (Olapurath, 1998). Por esta razón se considera adecuado para el desarrollo de la investigación en el marco del plan remedial de la MATI 2018-2019.

En términos más específicos a la ejecución de la investigación, se utilizó la revisión sistemática de literatura, cuyos orígenes en la ingeniería de sistemas están en los trabajos de Kitchenham, y posteriormente de Diva y Jørgensen, basados en revisiones preexistentes de otras áreas como los estándares médicos y revisiones sistemáticas propuestas por la sociología (Kitchenham & Brereton, 2013).

Para la elaboración de esta revisión se usó la estructura presentada por (Kitchenham & Charters, 2007b) como base para establecer la ruta de la investigación. La estructura definitiva se ilustra en la Tabla 4 a continuación.

Tabla 3: Estructura de Revisión Sistemática de Literatura

Fase	Actividades
Planificar la Revisión	Especificar Preguntas de investigación
	Elaborar el Protocolo de Investigación
Ejecutar la Revisión	Identificar Investigaciones Relevantes
	Seleccionar estudios Primarios
	Evaluar Calidad del Estudio
	Extraer Datos Requeridos
	Sintetizar los datos
Documentar la Revisión	Revisión de Objetivos
	Conclusiones y recomendaciones

Fuente: Elaboración propia, basado en (Kitchenham & Charters, 2007b)

La estructura de la tabla 4 difiere de lo propuesto por Kitchenham de la siguiente manera: la Planificación de la Revisión no contuvo la creación de una comisión de revisión ni la evaluación del protocolo, debido a que solo un investigador fue designado para esta revisión y para ambos procesos se requiere de más personas involucradas (Kitchenham & Charters, 2007a). Los apartados de la sección Documentar la Revisión, fueron substituidos para ajustar el proceso de revisión de literatura a las normas del plan remedial UNA 2018-2019, primeramente, y seguido por ser secciones desarrolladas en estudios de doctorado o como instrumento para publicación en una revista científica (Kitchenham & Charters, 2007a, págs. 39-43)

3.1.1 Enfoque de Investigación

En (Bekmamedova, y otros, 2018), los autores mencionan que los razonamientos deductivos o inductivos pueden usarse para analizar los datos y las extraer conclusiones. En particular se usó el razonamiento inductivo ya que se tomó como insumo los casos particulares encontrados en los objetos de estudio (publicaciones) para concluir principios y declaraciones generales sobre las fábricas de software.

Siguiendo con la estructura de la revisión sistemática de literatura establecida anteriormente en la Tabla 4, se debe establecer las preguntas de investigación

(P.I.). Para elaborar estas preguntas de investigación se tomó como guía los objetivos previamente definidos, dando como resultado el siguiente grupo de preguntas:

- **PI1:** ¿Cuáles son los elementos/componentes que definen una fábrica de software?
- **PI2:** ¿Cuáles son los elementos/componentes más comunes?
- **PI3:** ¿En cuales categorías predominan estos componentes?
- **PI4:** ¿Cuáles son los modelos de presentación de fábrica de software?

Se utilizó las especificaciones de (Kitchenham & Charters, 2007b) para la elaboración de preguntas de investigación, que consiste en 3 condiciones: que sea significativo e importante tanto para los profesionales como para los investigadores, que conduzca a cambios en la práctica actual de ingeniería de software o a una mayor confianza en el valor de la práctica actual, y que identifique discrepancias entre las creencias comunes y la realidad.

3.1.2 Tipo de Investigación

Se utilizó el tipo cuantitativo para la recolección y análisis de los datos con el fin de condensar la información de manera que resulte mecánico discriminar los elementos comunes en las fábricas de software.

Adicionalmente para el desarrollo de la revisión, se usó como modelo base la taxonomía de Harris M. Cooper, reunida por (Randolph, 2009) con el objetivo de especificar las características de las revisiones de literatura. La taxonomía establece seis características para las revisiones de literatura: enfoque, meta, perspectiva, cobertura, organización y audiencia, con sus respectivas categorías expuestas en la Tabla 4.

Tabla 4: Taxonomía de Cooper para una Revisión de Literatura

Característica	Categoría
Enfoque	Resultados de la investigación Métodos de búsqueda Teorías Prácticas o aplicaciones.
Meta	Integración: (a) Generalización (b) Resolución de conflictos. (c) Construcción de puentes lingüísticos. Crítica Identificación de temas centrales.
Perspectiva	Representación neutra Adopción de una posición
Cobertura	Exhaustivo Exhaustivo con cita selectiva. Representativa Central o pivote
Organización	Histórico Conceptual Metodológico
Audiencia	Académicos especializados Académicos generales Practicantes o formuladores de políticas Público en general

Fuente: (Randolph, 2009)

Según la taxonomía de Harris M. Cooper, el enfoque de la investigación puede tener cuatro categorías: resultado de investigaciones, metodologías de investigación, teorías y/o prácticas, o aplicaciones, que no son mutuamente excluyentes, pero típicamente hay un enfoque primario. El enfoque primario de esta investigación se encuentra dirigido al resultado de las investigaciones, seguido de un enfoque secundario hacia las prácticas y aplicaciones en las fábricas de software.

La meta es la integración, dirigida a la generalización, aunque no se discriminó la identificación de temas centrales encontrados en el proceso de exploración de entre varios artículos con enfoque empírico en la ejecución de fábricas de software. La perspectiva es esencialmente neutra, se tomó los hallazgos como hechos en la realidad de las publicaciones, sin embargo, esa veracidad fue evaluada más adelante con la evaluación de calidad de las publicaciones.

La cobertura es exhaustiva con cita selectiva, es decir, se consideró solo algunos tipos de publicaciones debido a la restricción de tiempo principalmente. La organización es metodológica pues tiene en general la estructura de Introducción, métodos, resultados y discusión, aunque se puede encontrar una parte histórica en la sección Situación de Investigación y otra conceptual en el Marco Teórico.

Finalmente, la audiencia, definida como académicos especializados, el tema está estrechamente relacionado a la práctica de desarrollo de software y esta encierra un cuerpo de conocimiento cuya adquisición queda fuera del alcance de este documento.

3.2 Sujetos y fuentes de información

Los sujetos centrales de la revisión de literatura son las publicaciones digitales con el tema “Fábrica de Software” como modelo de desarrollo de software. Las fuentes de información son las publicaciones que contengan información empírica de la utilización de fábricas de software, encontrados en el sistema Primo y en Google Académico. A estas publicaciones se les aplico un instrumento de extracción de

información descrito en la sección 3.6 Instrumentos para la recolección de información.

Estos sujetos pueden ser fuentes primarias en el caso de publicaciones con narración de la experiencia de los autores en la aplicación o participación en fábricas de software; o secundarias en caso de publicaciones con compilaciones de experiencias de terceros o nociones teóricas en la aplicación o participación de fábricas de software.

3.3 Población, Muestra o espacio de investigación

El conjunto universal de la población está enmarcado por las publicaciones digitales encontradas por Primo y Google Académico. Con el objetivo de seleccionar las publicaciones más significativas se diseñó el siguiente procedimiento dentro del Protocolo de Investigación definido anteriormente en la Tabla 4:

3.3.1 Estrategia de búsqueda

Para la búsqueda de publicaciones se definieron los siguientes criterios de búsqueda:

- Publicaciones entre 1998 y 2018
- Las siguientes palabras en inglés: *Software Factory, model, implementation, experience, components, elements, lessons, lessons learned, framework,*
- Las siguientes palabras en español: *fábrica de software, modelo, implementación, experiencia, componentes, elementos, lecciones, lecciones aprendidas, marco de trabajo, buenas prácticas.*
- La combinación de estas palabras con la siguiente lógica booleana:
 - a) (“*software factory*” OR “*Software factories*”) AND (*model* OR *implementation* OR *experience* OR *components*, OR *elements* OR *lessons* OR “*lessons learned*” OR *Framework* OR “*good practices*”)
 - b) “*software factory*” OR “*Software factories*”

- c) (*“fábrica de software” OR “Fabricas de Software”*) AND (*modelo OR implementación OR experiencia OR componentes, OR elementos OR lecciones OR “lecciones aprendidas” OR marco OR “marco de trabajo” OR “buenas practicas”*)
- d) *“fábrica de software” OR “Fabricas de Software”*

Esta lógica fue utilizada en la búsqueda avanzada de Primo y Google Académico, y adaptada debido a que cada motor de búsqueda tiene una forma particular de interpretar esa lógica. En seguida se describe el prendimiento de búsqueda en cada motor de búsqueda:

Primo:

Este buscador acepta cadenas de caracteres (palabras) de búsqueda en formato booleano, por lo que las cadenas definidas anteriormente no necesitaron de modificación para ser interpretados por el buscador. Por otro lado, permite escoger algunos de los campos en las publicaciones ser el objetivo para la búsqueda, i.e.: título, autor, tema, etiquetas de usuario, código de barras, clasificación, numero de sistema, descripción ISBN, ISSN o cualquier campo. De esta manera se puede escribir algo como *“fábrica de software”* aplicada solo al título con el fin de reducir los resultados. Por defecto Primo busca las palabras en cualquier campo.

En seguida una captura del motor de búsqueda Primo en la ilustración 2

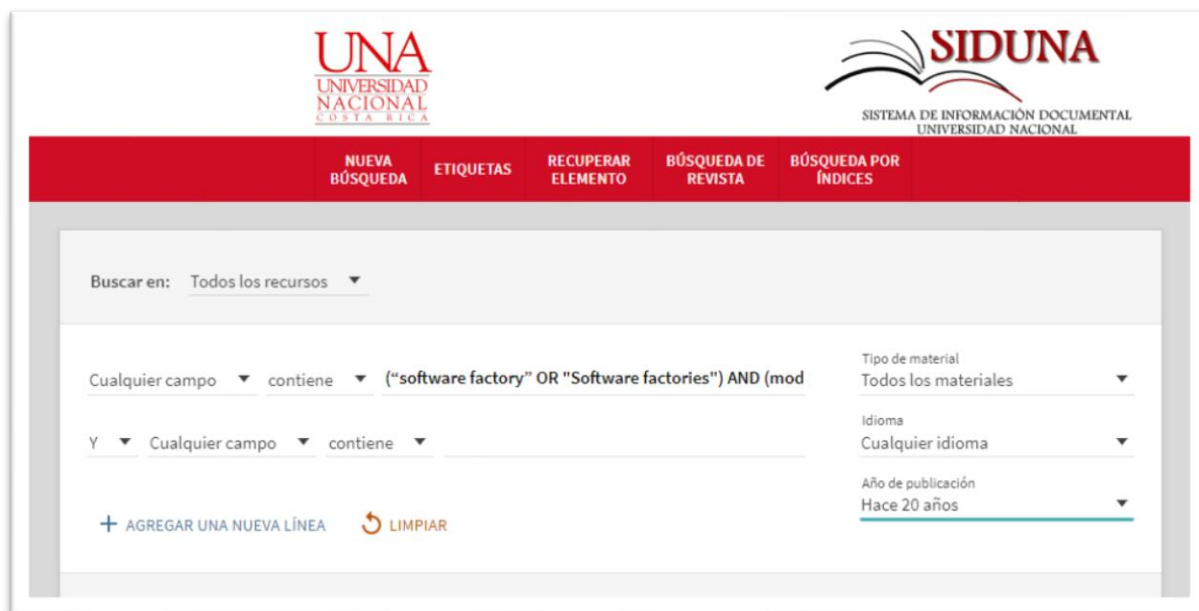


Ilustración 2: Motor de búsqueda Primo

Fuente: Elaboración propia

Con las palabras de finidas y con la lógica ya establecida anteriormente, se aplicó la combinación de palabras con los siguientes campos: cualquier campo, título, tema y descripción. Adicionalmente se registró los tipos de publicación y la cantidad de resultados, dando como resultado un total de 39 búsquedas.

Google Académico:

La lógica definida anteriormente para las búsquedas no es compatible por completo con el motor de Google Académico, por lo que se tuvo que modificar la presentación de las palabras para que la lógica se mantuviera consistente. La búsqueda avanzada solo permite una palabra obligatoria lo cual hizo que se distinguiera entre plurales y singulares. En la Ilustración 3 se muestra el buscador avanzado de Google Académico.

Ilustración 3: Búsqueda Avanzada de Google Académico

Google permite hacer la búsqueda con otros criterios como la búsqueda en los títulos o en todo el documento, así como agregar citas y patentes a los resultados de la búsqueda. Es importante mencionar que los resultados de las búsquedas son catalogados como artículos, aunque sean otro tipo de publicación, se mantuvo esta clasificación hasta la posterior lectura de los artículos seleccionados donde se pudo hacer una mejor distinción del formato.

Se aplicó la combinación de las nuevas cadenas y de los criterios de búsqueda definidos anterior mente, dando como resultado un total de 48 búsquedas.

Los resultados de cada una de las búsquedas fueron registrados en una tabla (i.e.: Tabla 6) con el siguiente formato:

Tabla 5: Formato para registro de resultados de búsqueda

Motor de Búsqueda	Cadena de Búsqueda	Aplicado a	Tipo de Publicación	Cantidad de Resultados

Fuente: Elaboración propia

En la columna “Criterio de Búsqueda” se introduce la combinación de palabras, en “Tipo de Publicación” se ingresa tipo de publicación (tesis, artículo, libro, conferencia, otro), “Motor de Búsqueda” será Primo o Google Académico, “Aplicado a” contiene en parte de la publicación se centró la búsqueda, mientras que “Cantidad de Resultados” almacena la cantidad de publicaciones encontradas.

Con la presente limitante de tres semanas para realizar la extracción y síntesis de información, se estimó que se podría evaluar un total de 50 publicaciones, por lo que se requirió de un acotamiento de los resultados, descrito más adelante en la sección de procedimiento de selección.

3.3.2 Criterio de Inclusión

Los criterios de inclusión que se utilizaron para determinar cuáles publicaciones (artículos, tesis, libros y conferencias) encontradas por los términos de búsqueda se utilizarían para la extracción de datos fueron los siguientes:

- Publicaciones de acceso libre
- Estudios que presenten experiencia empírica o datos de experiencias empíricas en la utilización de fábricas de software
- Publicaciones que colaboren con la respuesta a las preguntas de investigación

- Alguna otra publicación encontrada como referencia en los artículos seleccionados que colabore con la respuesta a las preguntas de investigación

3.3.3 Criterio de Exclusión

Los criterios de inclusión que se utilizarán para determinar cuáles publicaciones (artículos, tesis, libros y conferencias) encontradas por los términos de búsqueda se excluirán para la extracción de datos.

Los criterios se enumeran a continuación:

- Estudios que no presenten experiencia empírica explícita o implícita en las fábricas de software
- Artículos de acceso restringido o protegido por derechos de copia.

3.3.4 Procedimiento de Selección

El propósito de la selección fue obtener un número de publicaciones que fuera específico al tema de fábricas de software y que permitiera el estudio pertinente dentro de las restricciones de tiempo mencionadas anteriormente. Se realizó una serie de búsquedas para validar que los resultados fueran relevantes a la revisión y que la cantidad estuviera dentro de un rango que se aproximara a 50 publicaciones.

El procedimiento de selección se definió de la siguiente manera:

En Primo, las publicaciones que cumplieran con los criterios a) y c) (sesión 3.3.1) y en el título y cuyo acceso fuera público. En Google académico, las publicaciones que cumplieran con los criterios a) y c) aplicados al título y cuyo acceso fuera público.

Teniendo presentes los criterios de inclusión y exclusión, se realizó una revisión manual de los títulos, resumen y alguna otra sección de interés de los resultados de búsqueda con el fin de contextualizar más detalladamente la publicación.

Las publicaciones que cumplieran con los criterios de búsqueda y públicos fueron listadas en la tabla de registro de publicaciones primarias, cuyo formato se muestra

en la Tabla 6: Tabla de Registro de publicaciones primarias y marcados como excluidos si los criterios de inclusión no fueron satisfechos

Tabla 6: Tabla de Registro de publicaciones primarias

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido

Fuente: Elaboración propia

La columna “ID” corresponde a un número identificador de la publicación, mientras que “Tipo publicación” se refiere a tesis, artículo, libro, conferencia u otro. Las demás columnas son auto explicativas.

3.3.5 Evaluación de calidad

La evaluación de calidad se realiza con el objetivo de asegurar que las investigaciones sean relevantes para la investigación y que tengan un alto grado de imparcialidad (Stapić, de-Marcos, Strahonja, García-Cabot, & García López, 2016). Esto ayuda a centrar los esfuerzos de la investigación, la extracción de datos, así como los reportes finales en las publicaciones con la mejor calidad y menor sesgo.

Las preguntas que evaluaron la calidad de las publicaciones fueron:

Imparcialidad

- **C1:** ¿Está la publicación **libre** de mercadotecnia a favor del uso de una herramienta, modelo o marco de trabajo?
- **C2:** ¿Se percibe que los autores están **exentos** de motivaciones atenten contra la imparcialidad?

Relevancia

- **C3:** ¿Es la descripción de procedimientos, patrones, modelos, métodos, marcos de trabajo y/o herramientas **explícitas**?
- **C4:** ¿Contiene información que **ayuda a responder** las preguntas de investigación o los objetivos de esta investigación?

Las preguntas C1 y C2 fueron semánticamente modificadas para que la calificación fuera la misma para todas ellas y cumpliera con el estándar usado en otras publicaciones (da Silva, y otros, 2011), (i.e.: responder si tiene una connotación positiva y mejora la calidad de la publicación). Básicamente lo que se quiere saber es si la publicación está sujeta a algún producto o servicio, principalmente propietario en el caso de C1 y si se percibe otra motivación por parte de los autores que afectara la imparcialidad de su información en el caso de C2.

La calificación mencionada anteriormente es: S=1, P=0.5, N=0, siendo S = SI, P = Parcial y N = NO (Kitchenham, y otros, 2010). De esta manera el mayor puntaje de calidad es de 4, al contestar SI a todas las preguntas. La evaluación de calidad se realizó en paralelo con la extracción de los datos.

3.4 Definición de variables

A continuación, se definen las variables de la investigación y sus posibles valores, estos últimos fueron solo aproximaciones pues son una interpretación inicial de la situación de investigación. Se agregó el comodín “otro” con el fin de abarcar los valores que no fueron considerados desde el principio pero que resultaron relevantes para alcanzar los objetivos. Posteriormente, “otro” fue revaluado y signado un valor.

Tabla 7: Variables de la investigación

Objetivo específico	Variable	Valores
Identificar los elementos del esquema de fábrica de software y sus variantes para enmarcar el concepto de fábrica de software	Publicaciones	Tesis, artículo, libro, conferencia, otro.
	Componentes	Procesos, modelos, marcos de trabajo, herramientas, otro
Generar una clasificación de los elementos del esquema de fábrica de software que permita agruparlos según su relación	Categorías de componentes	Planificación Requerimientos Diseño y prototipo Desarrollo de software Pruebas Despliegue Operación y mantenimiento Otro
Sintetizar un modelo de prestación de servicios de desarrollo de software basado en fábrica de software	Relación entre Componentes	Diagrama Modelo

Fuente: Elaboración propia

3.5 Descripción de extracción de datos

Esta sección describe el procedimiento seguido para extraer la información de los artículos seleccionados después de pasar por la estrategia de búsqueda y un inicial proceso de aplicación de los criterios de inclusión y exclusión. Cabe destacar que los criterios de inclusión y exclusión se mantuvieron vigentes aun durante esta etapa.

3.5.1 Estrategia de Extracción

En esta fase, se utilizó un formulario de extracción de información para registrar la información obtenida de cada publicación. Con el objetivo de evitar un sesgo, los formularios fueron creados antes de empezar con la extracción de la información (Brereton, Kitchenham, Budgen, Turner, & Khalil, 2006). Este paso es particularmente importante debido a que la extracción fue realizada por un solo investigador y la posibilidad de caer en un sesgo era mayor.

Dado que durante la extracción de datos también se ejecutó la evaluación de calidad, se diseñó un instrumento que mantuvo el registro de la calificación de calidad y de los hallazgos en las publicaciones seleccionadas. El detalle del documento se encuentra en la sección 3.6 Instrumentos para la recolección de información.

Establecido lo anterior, el proceso consistió en tomar una de las publicaciones seleccionadas, realizar la lectura de la publicación, a medida que el investigador avanza extrae la información en el instrumento de recolección de datos. Al cabo de la lectura, el investigador vuelve a la lista de publicaciones seleccionadas y realiza el mismo proceso hasta acabar con las publicaciones escogidas para revisión.

3.6 Instrumentos para la recolección de información

A continuación, se describen los instrumentos que serán utilizados en el procedimiento de extracción de datos expuesto anteriormente.

Durante la lectura de las publicaciones seleccionadas, el investigador debe registrar sus hallazgos en la tabla de evaluación de calidad y recolección de datos, cuyo formato se observa en la Tabla 9:

Tabla 8: Evaluación de Calidad y Recolección de Datos

ID	Calidad				DATOS						Comentario
	Imparcial		Relevancia		PI1				PI4		
	C1	C2	C3	C4	Herramienta	Procedimiento	Marco de trabajo	Modelo	Otro	Modelo de FS	

Fuente: Elaboración propia

La columna “ID” corresponde al número identificador de la publicación definido previamente en la selección de publicaciones (Tabla 6: Tabla de Registro de publicaciones primarias). Dentro de la sección de “Calidad” se encuentra el código de las preguntas de calidad divididas por su intención. Bajo la sección de “DATOS” se encuentra el código de dos de las cuatro preguntas de investigación, con las que se espera guiar al investigador durante la lectura. Bajo cada pregunta de investigación están las columnas que contendrán los hallazgos, aquí el lector introduce una interpretación del hallazgo en la columna que considere apropiada.

3.7 Proceso e instrumentos de tabulación de los datos

Esta sección contiene la descripción del procedimiento de síntesis de datos con el instrumento de síntesis de datos.

3.7.1 Síntesis de Datos

El proceso de síntesis consistió en la condensación de los datos extraídos. De acuerdo con (Stapić, de-Marcos, Strahonja, García-Cabot, & García López, 2016)

existen dos métodos mutuamente excluyentes para la síntesis de la información recolectada, la Narrativa y la Cuantitativa. Se usó el método cuantitativo, donde se presentará los resultados de la extracción de forma tabular respondiendo puntualmente a las preguntas de investigación con el resultado del conteo en el instrumento tabla de síntesis de datos, definida a continuación en la Tabla 10:

Tabla 9: Tabla de síntesis de datos

PI3	PI2		
Categoría	Tipo Componente	Componente / Elemento	Cantidad de Artículos

Fuente: Elaboración propia

Como se puede observar, la tabla está encabezada por los códigos de las preguntas de investigación que pretendían responder, esto funcionó de guía al investigador durante la síntesis de datos.

La columna “Categoría” puede contener una serie de valores que inicialmente se definen como los componentes genéricos del ciclo de vida del desarrollo de software: Planificación, Requerimientos, Diseño y prototipo, Desarrollo de software, Pruebas, Despliegue, Operación y mantenimiento (Shah, 2016) (Ghezzi, Jazayeri, & Mandrioli, 2003) (Cusumano, 1987b), aunque puede tener otros valores si se encuentra referencia aspectos fuera del ciclo de vida del software.

La columna “Componente / Elemento” almacena el hallazgo, mientras que “Cantidad de Artículos” es la sumatoria de los artículos donde el componente o elemento es mencionado.

En el siguiente capítulo se describen el proceso de ejecución del plan definido en este capítulo. Cada sección definida en previamente es desarrollada y encuentra una sección homologa comprendiendo el procedimiento descrito y los instrumentos elaborados.

4 Capítulo IV: Análisis de resultados

Este capítulo continúa con el formato de revisión de literatura definido anteriormente en la fase ejecución, donde se presenta el resultado de los procesos de selección, extracción y síntesis de la información

4.1 Investigaciones Relevantes

El resultado de las búsquedas aplicando los criterios definidos en Google académico y en el sistema Primo, dio un total de 22.687 publicaciones relacionadas de una u otra forma con fábricas de software. Mediante el proceso de selección definido en la sección 3.3.4, el resultado se redujo a un total de 61 publicaciones.

El detalle de los resultados se encuentra el anexo 8.1

4.2 Publicaciones primarias

Las publicaciones fueron inspeccionadas con el fin de contextualizarlas y aplicar los criterios de inclusión y exclusión. Dos publicaciones no fueron de acceso público o gratuito, mientras que 14 fueron excluidos por los criterios de exclusión durante esta primera revisión o durante la fase de extracción de datos. Un total de 45 publicaciones fueron seleccionadas para la extracción de los datos.

El detalle de este proceso se encuentra en el anexo 8.2

4.3 Datos extraídos y Evaluación de calidad

El proceso de evaluación de la calidad y la extracción de los datos se realizó en paralelo, de manera que por conveniencia se creó la Tabla 8: Evaluación de Calidad y Recolección de Datos . Sin embargo, una vez finalizado el proceso, la tabla fue separada con el propósito de analizar mejor los datos.

Calidad:

La evaluación de calidad de las publicaciones se realizó tomando basado en las cuatro preguntas de calidad y en la rúbrica definida en la sección 3.3.5, donde $S=1$, $P=0.5$ y $N=0$. De esta manera una publicación con calificación de cuatro resulta ser el de mayor calidad. Los resultados sumariados se presentan en la Tabla11:

Tabla 10: Condensado de Calidad

Calificación	Publicaciones
4	6
3,5	18
3	12
2,5	8
2	4
1,5	3
0	8

Fuente: Elaboración propia

Las publicaciones con notas inferiores a 2.5 fueron descartadas a excepción de [7], por presentar una lista considerable de herramientas de software utilizadas en el desarrollo de aplicaciones a través de diversas universidades europeas.

El detalle de las evaluaciones de calidad puede encontrarse en el anexo 8.3

Datos

Con la lectura de las publicaciones se fue extrayendo los elementos que, según el criterio de lector y la definición, formaban parte fundamental de las fábricas de software. Se catalogaron de manera inicial bajo las categorías de Marco de trabajo, Modelo, Procedimiento, Herramienta u Otro. El propósito preliminar fue encontrar lo que explícitamente es utilizado o está diseñado para ser (o está) implementado en una fábrica de software, excluyendo los elementos en las definiciones del concepto. El resultado final fue un total de 64 elementos encontrados. En algunos casos como (Pihoa, Tepandia, Roosta, Parmana, & Puusep, 2011), (Jalil & Bakar, 2017), (Thomas, Ekwoke, & Tangha, 2010), (Ávila, Supino Marcondes, Vieira, & Marques, 2010) y otros, solo se menciona superficialmente algún elemento, pero presentan una instancia de fábrica de software abstraída, por lo que, a pesar de no aportar mucho a los datos de PI1, sirven como referencia para PI4. En estos casos particulares, la columna de comentarios sirvió como elemento de trazabilidad para no perder de vista estas publicaciones.

El detalle de los datos recolectados se encuentra en el anexo 8.4.

4.4 Síntesis de Datos

Con los datos recolectados se realizó un proceso de agrupación y conteo donde se clasificaron de manera que se pudiera distinguir cuales eran los elementos, los tipos de componentes y las categorías más comunes en las experiencias de los autores.

Como resultado se obtuvieron las siguientes categorías y tipos de componentes (tablas 12 y 13), junto a la cantidad de componentes en cada uno de ellos:

Tabla 11: Cantidad de Elementos por Categoría

Categoría	Cantidad de elementos
Diseño	21
Planificación	11
Desarrollo	10
Mejora continua	6
Calidad	4
Productividad	5
Requerimientos	3
Operación y Mantenimiento	2
Seguridad	1
Servicios	1
Total	64

Fuente: Elaboración propia

Tabla 12: Cantidad de Elementos por Tipo de componente

Tipo Componente	Cantidad de Elementos
Herramienta	26
Marco de trabajo	14
Modelo	13
Procedimiento	10
Entorno	1
Total	64

Fuente: Elaboración propia

Finalmente, de los 64 elementos encontrados, estos son los que encabezan la lista con más de una aparición:

Tabla 13: Componentes más comunes en fábricas de software

PI3		PI2	
Categoría	Tipo Componente	Componente / Elemento	Cantidad de Artículos
Desarrollo	Procedimiento	Generación de Código	12
Productividad	Modelo	SPL	10
Diseño	Modelo	DSL Domain Specific Language	7
Diseño	Modelo	MDD	7
Diseño	Marco de trabajo	MDA Model Driven Architecture	6
Planificación	Herramienta	BSC	4
Mejora continua	Modelo	CMMI	4
Diseño	Modelo	KM Gestión Conocimiento	4
Productividad	Procedimiento	PSP/TSP	4
Desarrollo	Procedimiento	Ágil	3
Mejora continua	Modelo	CMMI3	3
Calidad	Marco de trabajo	ISO15504 / SPICE Software Process Improvement Capability Determination	3
Mejora continua	Procedimiento	Mejora continua	3
Desarrollo	Procedimiento	Scrum	3

PI3		PI2	
Categoría	Tipo Componente	Componente / Elemento	Cantidad de Artículos
Planificación	Procedimiento	AHP (Analytic Hierarchy Process)	2
Diseño	Marco de trabajo	CBSE Component-based software engineering	2
Mejora continua	Modelo	CMMI2	2
Operacion y Mantenimiento	Entorno	DevOps	2
Diseño	Herramienta	DSL .Net DSL tools	2
Calidad	Marco de trabajo	ISO9001	2
Desarrollo	Procedimiento	Kanban	2
Productividad	Modelo	Medir productividad	2
Planificación	Herramienta	RBT Resource Based Theory	2
Desarrollo	Herramienta	SCR software component repository	2
Diseño	Modelo	UML	2

Fuente: Elaboración propia

En el anexo 8.5 se puede encontrar la lista con los 64 elementos encontrados

4.5 Modelos de Fabrica de software

Como fue definido anteriormente, algunos elementos importantes de la fábrica de software son la reutilización de activos como diseños, arquitecturas, procedimientos, conocimiento y la automatización de desarrollo. Los resultados listados en la Tabla 14, muestran concordancia con estos elementos que constituyen el concepto de fábrica de software al tener dentro de los elementos más mencionados la generación de código, las líneas de producción de software (SPL), DSL (Lenguajes específicos de dominio), MDD (Desarrollo guiado por modelos), la gestión del conocimiento, controles y mejora en la calidad (ISO y CMMI), entre otros.

Durante la extracción de datos se pudo identificar un número de modelos de fábricas de software, que, aunque similares, varían en propósito. En la Tabla 15 se muestran los modelos y su correspondiente aparición:

Tabla 14: Modelos de Fábricas de Software:

Modelo de Fabrica de Software	Cantidad de FS encontradas
SF ¹ genérica	8
SF académica	6
SF específica	3
SF MDSF: Model Driven Software Factory	3
MDA	2
SF nube	2
SF para juegos	2
CMMI	1
CMMI3	1
ISO9001, ISO20000, ISO270001	1
SF Brasil	1
SF data warehouse	1
SF multimedia	1
SF Ad-hoc :ISO9000 CMM TSP/PSP SPL= SFM-CSO[ISO/CMM]	1

Fuente: Elaboración propia

¹ SF = Software Factory , significa Fabrica de Software en ingles

SF genéricas son los modelos que utilizan el concepto fábricas de software para generar un modelo general o sin un contexto específico. SF académicas son propuestas implementadas en universidades. SF específicas generalmente atienden contexto como en [2], cuyo modelo está inscrito al entorno financiero por lo que le han llamado "*Financial Software Factory*"; o en [20], donde es específico para sistemas ubicuos. Particularmente SF MDFS presenta modelos más abstractos orientados satisfacer más a cabalidad el concepto de fábrica de software principalmente en la reutilización de componentes y el desarrollo dirigido por modelos (MDD). MDA son modelos de fábrica de software enfocadas en el uso del modelo propuesto por OMG (*Object Management Group*). SF nube son dos modelos que pretenden mejorar las mejoras continuas a los sistemas ERP en la nube ofrecidas como SaaS (*Software as a Service*). SF para juegos es en realidad la evolución de un modelo inicial de fábrica para el desarrollo de video juegos.

Los CMMI se basan en niveles de madures donde se estable que en los niveles 2 y 3 los procesos deben ser gestionados y definidos respectivamente, algunos subprocesos evalúan la capacidad de obtener resultados repetibles (Software Engineering Institute, 2010). Al, igual que CMMI, los ISO presentan estándares de producción y calidad, con el propósito de cumplir con ellos, algunas publicaciones se apoyan en el concepto de fábrica de software como herramienta. Brasil cuenta con su propio Modelo de Mejora de Procesos de Desarrollo de Software, basado en otros modelos de madurez y que contiene un apartado para su implementación en fábricas de software (Menolli, Belmonte, Sgarbi, Malucelli, & Reinehr, 2011).

SF data *warehouse* y SF multimedia podría ser fábricas específicas pero su definición contiene elementos que arquitectura que son particulares al entorno, similar la SF nube. Finalmente, el caso más particular, SF Ad-hoc: ISO9000 CMM TSP/PSP SPL= SFM-CSO [ISO/CMM], es un modelo sencillo pero completo de las generalidades de una fábrica de software que satisfaga aspectos de ISO, CMMI y SPL en el ámbito de empresas pequeñas en China, lo cual la vuelve muy específica pues utiliza estudios relacionados la cultura asiática para generar un modelo que permita fortalecer las debilidades de las fábricas de software.

El detalle de la distribución de los modelos en las publicaciones se encuentra en el anexo 8.6

4.6 Modelo de Fabrica de software

Como objetivo de la revisión de literatura se planteó la presentación de un modelo de fábrica de software utilizando los componentes más comunes hallados en las publicaciones. Inicialmente se modelaron solo los elementos más frecuentes de cada categoría, pero a pesar de ser los más mencionados, se consideró que no brindaban un entendimiento completo del entorno de las fábricas de software, por lo que se modelaron todos los elementos encontrados, agrupados por su categoría y tipo, usando la Tabla 22: Elementos de fábrica de software encontrados como fuente de los datos.

El resultado fue un diagrama con las relaciones entre categorías, conteniendo estas a su vez los elementos correspondientes. En las siguientes páginas se describen los elementos del diagrama, seguido del diagrama relacionado.

Elementos de Planificación

El filtrado de los elementos de planificación mostró que los elementos más mencionados fueron el BSC (*Balance Score Card*) con cuatro menciones, RBT (*Resource Based Theory*) con dos menciones, WST (*Work System Theory*) con dos menciones, y el AHP (*Analytical Hierarchy Process*) con dos menciones. Se resaltaron con color para distinguirlos de los menos mencionados.

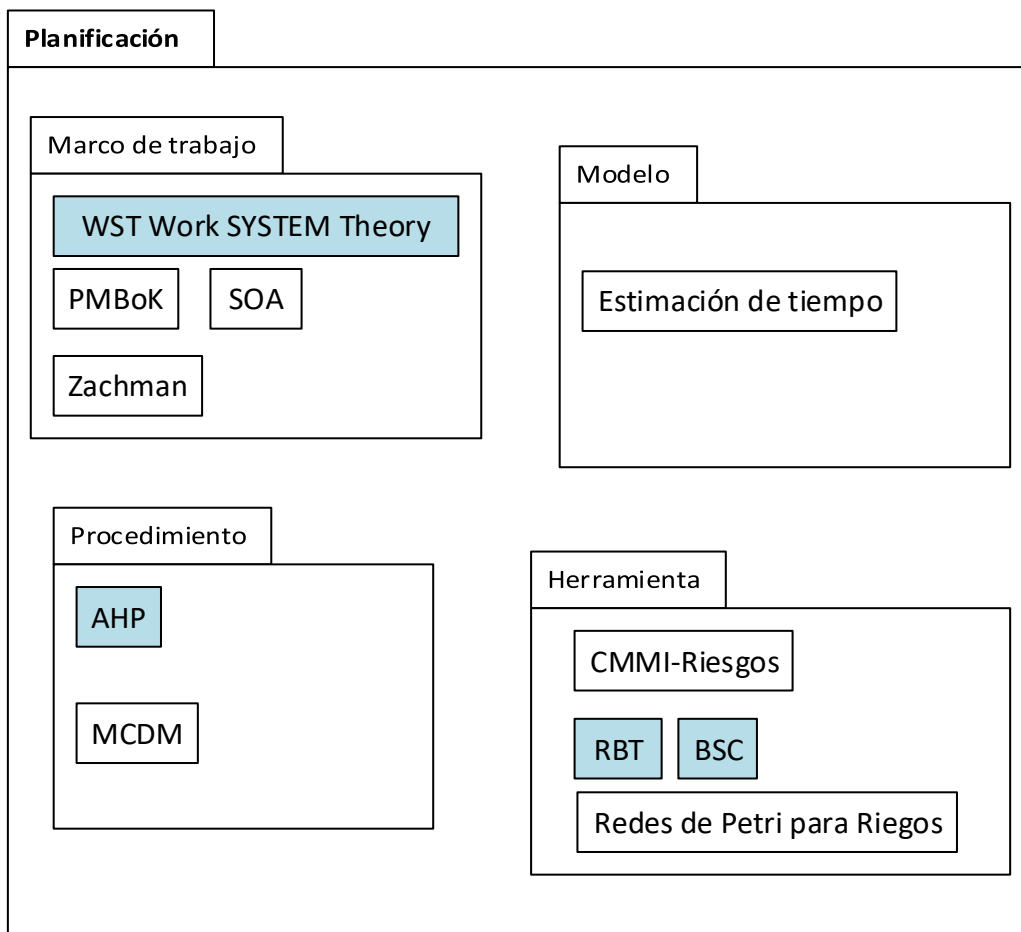


Ilustración 4: Elementos de Planificación

Fuente: Elaboración propia

Elementos de Requerimientos

En el caso de los requerimientos, ninguno de los elementos fue mencionado más de una vez, por lo que no hay elementos resaltados. Esta categoría fue una de las razones por las que se decidió incluir todos los elementos en el modelo, de haberlo incluido, el modelo tendría un espacio entre planificación y diseño, lo cual podría causar confusión al lector.

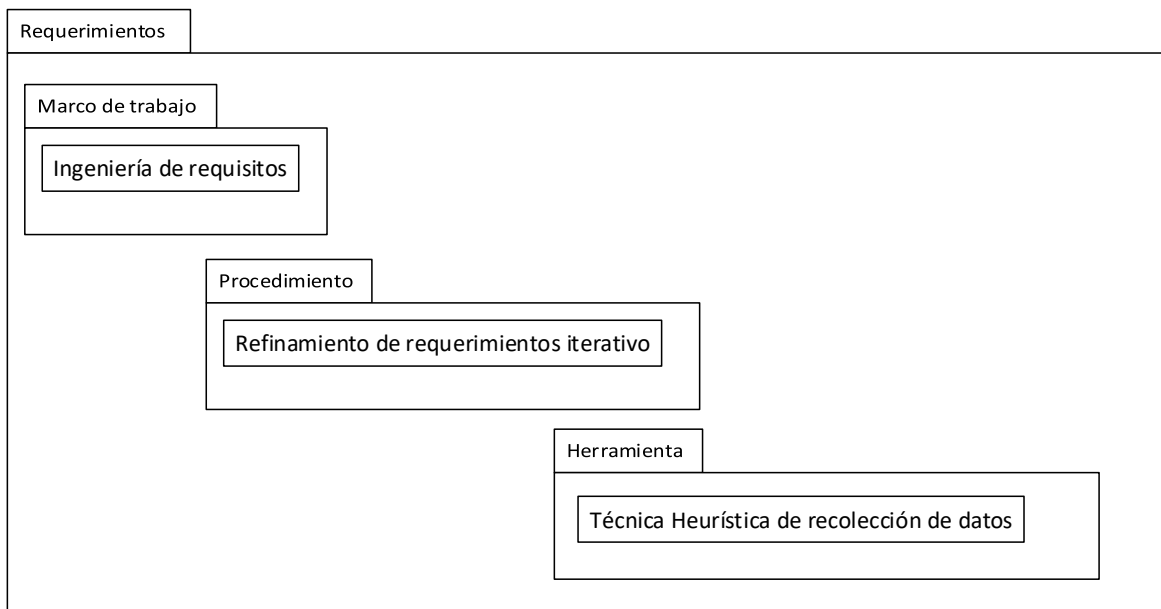


Ilustración 5: Elementos de Requerimientos

Fuente: Elaboración propia

Elementos de Diseño

El filtrado por diseño arrojó los siguientes elementos más comunes: DSL (*Domain Specific Language*) con siete apariciones, MDD (*Model Driven Development*) con siete apariciones, MDA (*Model Driven Architecture*) con seis apariciones, KM (*Knowledge Management*) con 4 apariciones, CBSE (*Component Based Software Engineering*) con 2 apariciones, .NET DSL tools con 2 apariciones y UML (*Unified Modeling Language*) con 2 apariciones.

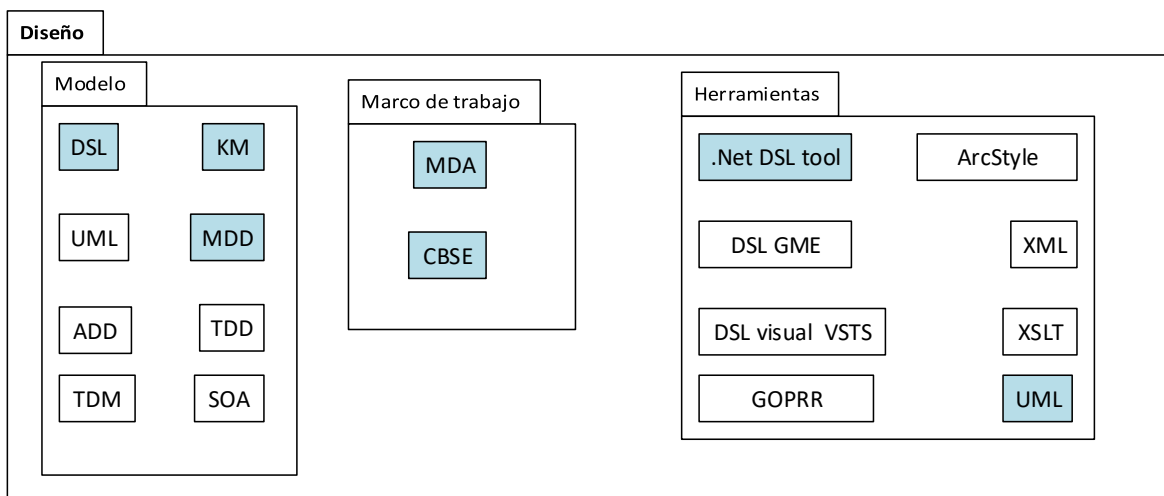


Ilustración 6: Elementos de Diseño

Fuente: Elaboración propia

Elementos de Pruebas

No existía la categoría de pruebas dentro de la tabla de síntesis de datos, pero al terminar el modelado inicial, fue evidente la falta de componentes las tareas de prueba para darle sentido al resto de relaciones entre categorías. Se observó que en la categoría de diseño, se hace mención al modelado dirigido por pruebas (TDM) y el desarrollo dirigido por pruebas (TDD), de los cuales no se encontró descripción detallada en las publicaciones, pero al menos permite visualizar a la existencia de procedimientos que pueden soportar estas labores.

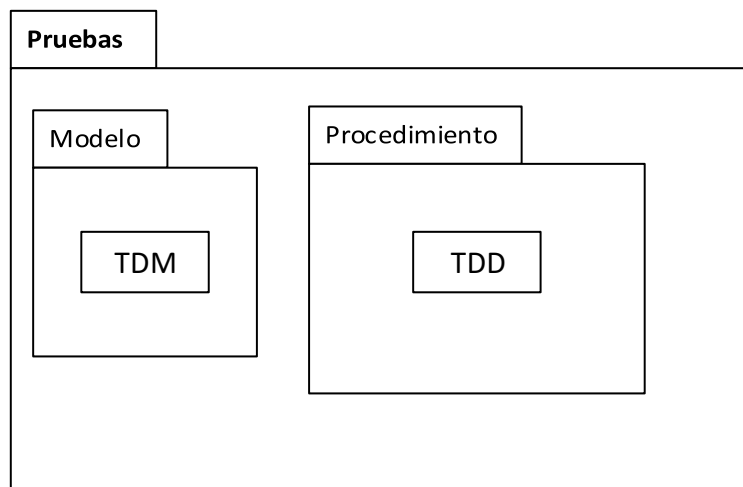


Ilustración 7: Elementos de Pruebas

Fuente: Elaboración Propia

Gestión de Conocimiento

Otro cambio a la presentación de los elementos fue la creación de una agrupación de gestión del conocimiento, cuya importancia está justificada en (Jalil & Bakar, 2017), (Dzulkafli, 2018) y (Maulini, Straccia, & Pollo, 2018), se consideró relevante en el entorno de las fábricas de software. Cabe mencionar que también se percibió la existencia de una relación entre el desarrollo de software basado en componentes (CBSE) y la gestión de conocimiento, pues ambos son elementos de reutilización, siendo el CBSE la parte aplicada del conocimiento.

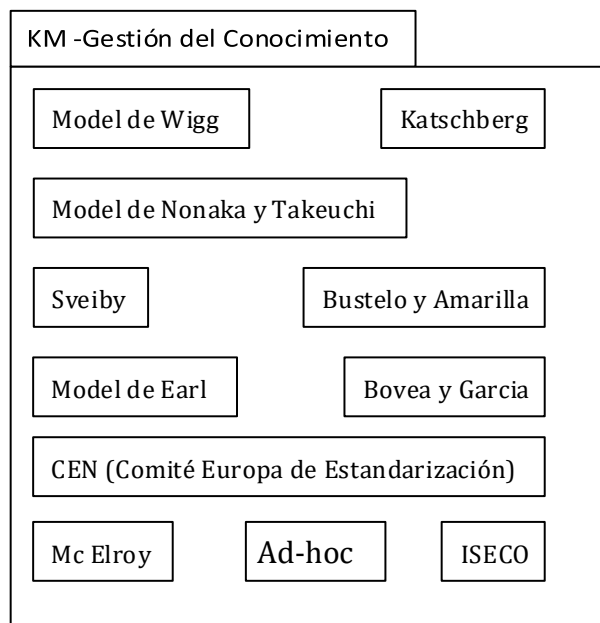


Ilustración 8: Elementos de la Gestión del Conocimiento

Fuente: Elaboración Propia

Elementos de Desarrollo

Los elementos más comunes bajo la categoría de desarrollo son: Generación de código con 12 apariciones, metodologías ágiles con tres, metodología Scrum con tres menciones, método Kanban con dos apariciones y el SCR (*Software Component Repository*) con 2 apariciones.

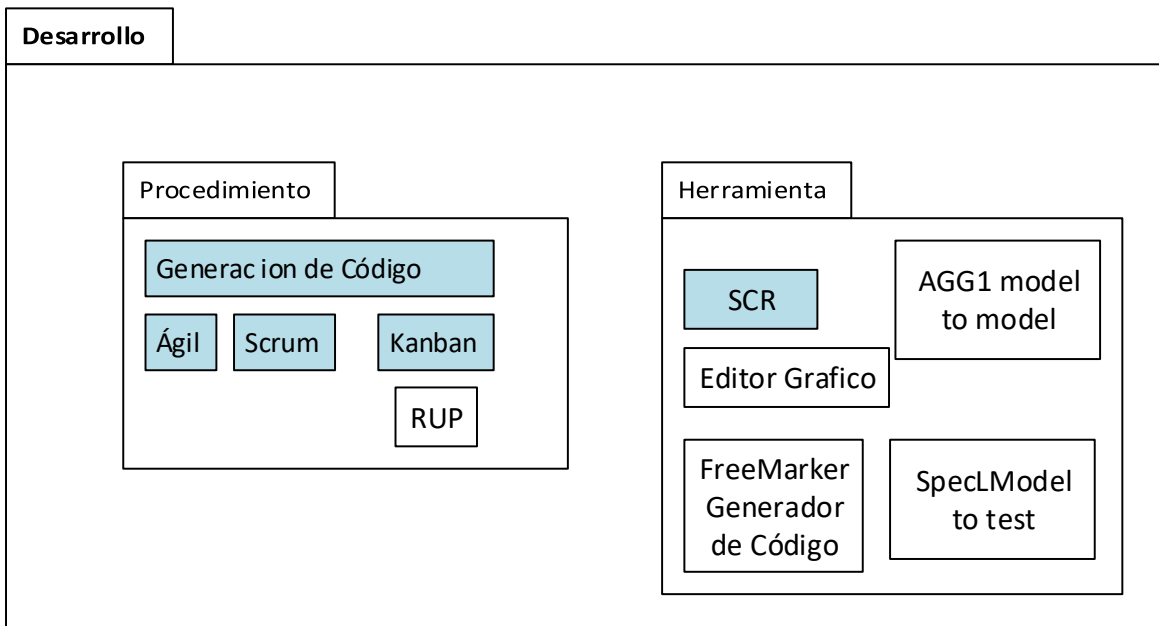


Ilustración 9: Elementos de Desarrollo

Fuente: Elaboración propia

Elementos de Mejora Continua

Mejora continua contiene pocos elementos y similares entre sí: CMM (*Capability Maturity Model*) con cuatro apariciones, CMMI3 (*Capability Maturity Model Integration* nivel 3) con tres apariciones, algún procedimiento de mejora continua no especificado con tres apariciones, CMMI2 (*Capability Maturity Model Integration* nivel 2) dos apariciones. La mejora continua puede ser considerada dentro de la categoría de calidad, pero decidió crear la categoría después de la revisión inicial, al notar que en algunas publicaciones, era un eje central.

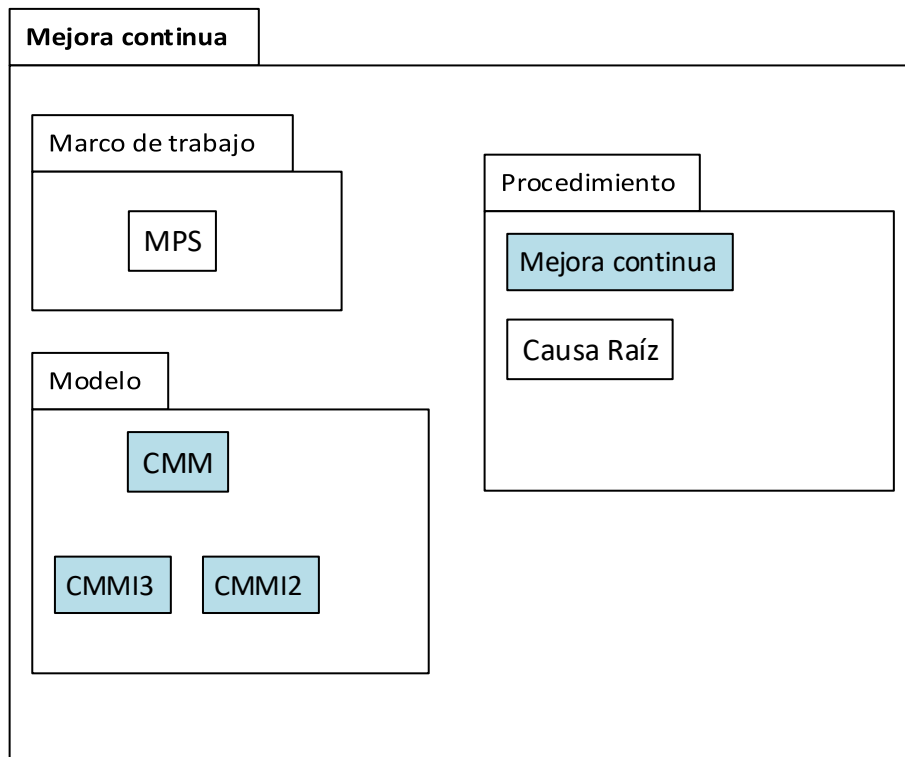


Ilustración 10: Elementos de Mejora continúa

Fuente: Elaboración propia

Elementos de Calidad

Dentro de los elementos de calidad resaltan el ISO/EIC 15504, también conocido como SPICE (*Software Process Improvement Capability Determination*) con 3 menciones e ISO 9001 con 2 menciones.

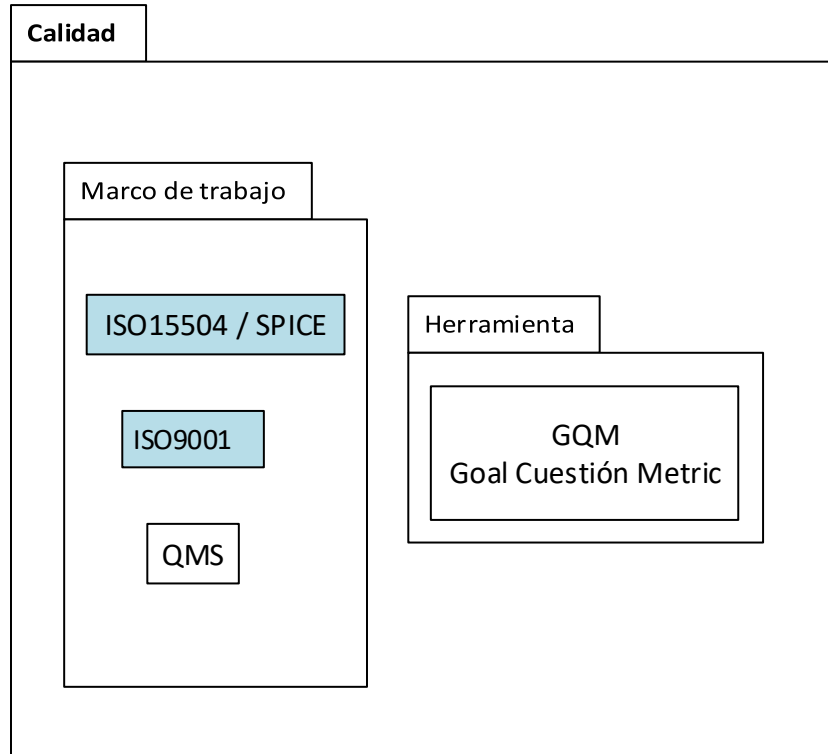


Ilustración 11: Elementos de Calidad

Fuente: Elaboración propia

Elementos de productividad

Uno de los elementos de la fábrica de software más recurrente se encuentra en la productividad, este es el SPL (*Software Product Line*), la línea de producción de software con diez apariciones, seguido del PSP/TSP (*Personal Software Process / Team Software Process*) con cuatro apariciones y un modelo de para medir la productividad no especificado con dos apariciones.

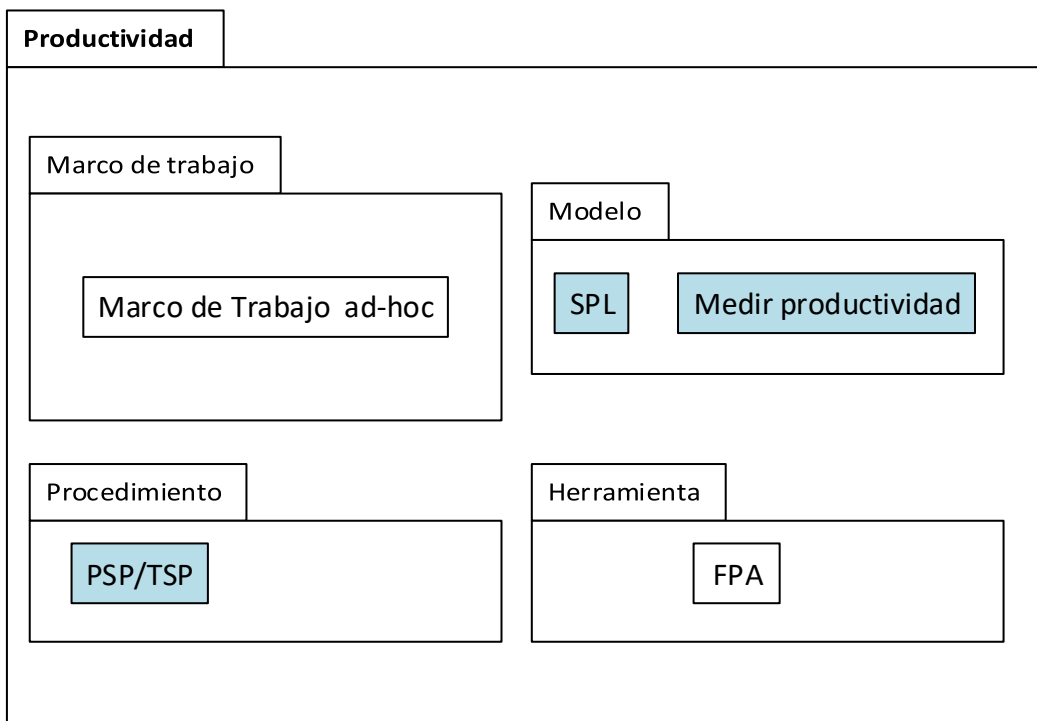


Ilustración 12: Elementos de Productividad

Fuente: Elaboración propia

Seguridad

La categoría de seguridad solo presenta a ISO27001 como marco de trabajo para el manejo de la seguridad de la información.

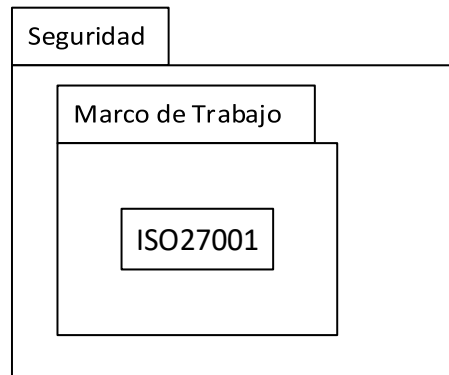


Ilustración 13: Elementos de Seguridad

Fuente: Elaboración propia

Elementos de Operación y Mantenimiento

Las actividades de operación y mantenimiento contienen el concepto de DevOps mencionado dos veces que consiste en integrar la ingeniería de software (Dev) y la gerencia de TI (Ops) en un solo bloque de actividades [52], seguido de SLA (*Service Level Agreement*).

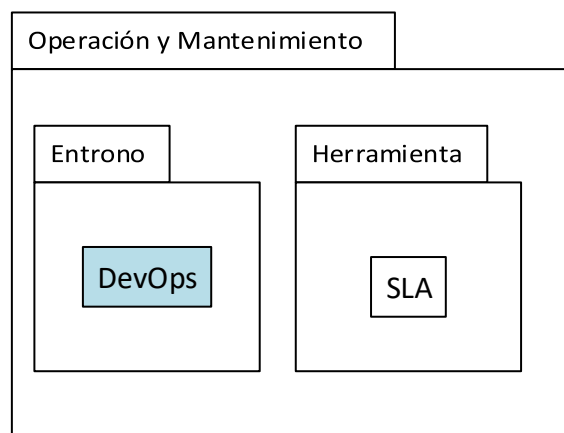


Ilustración 14: Elementos de Operación y Mantenimiento

Fuente: Elaboración propia

Servicios

La categoría de servicios solo menciona la norma ISO20000 para la gestión de servicios de TI.

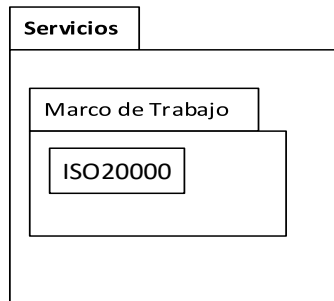


Ilustración 15: Elementos de Servicios

Fuente: Elaboración Propia

El modelo de las relaciones de categorías de elementos de la fábrica de software se encuentra en la Ilustración 16, en la siguiente página:

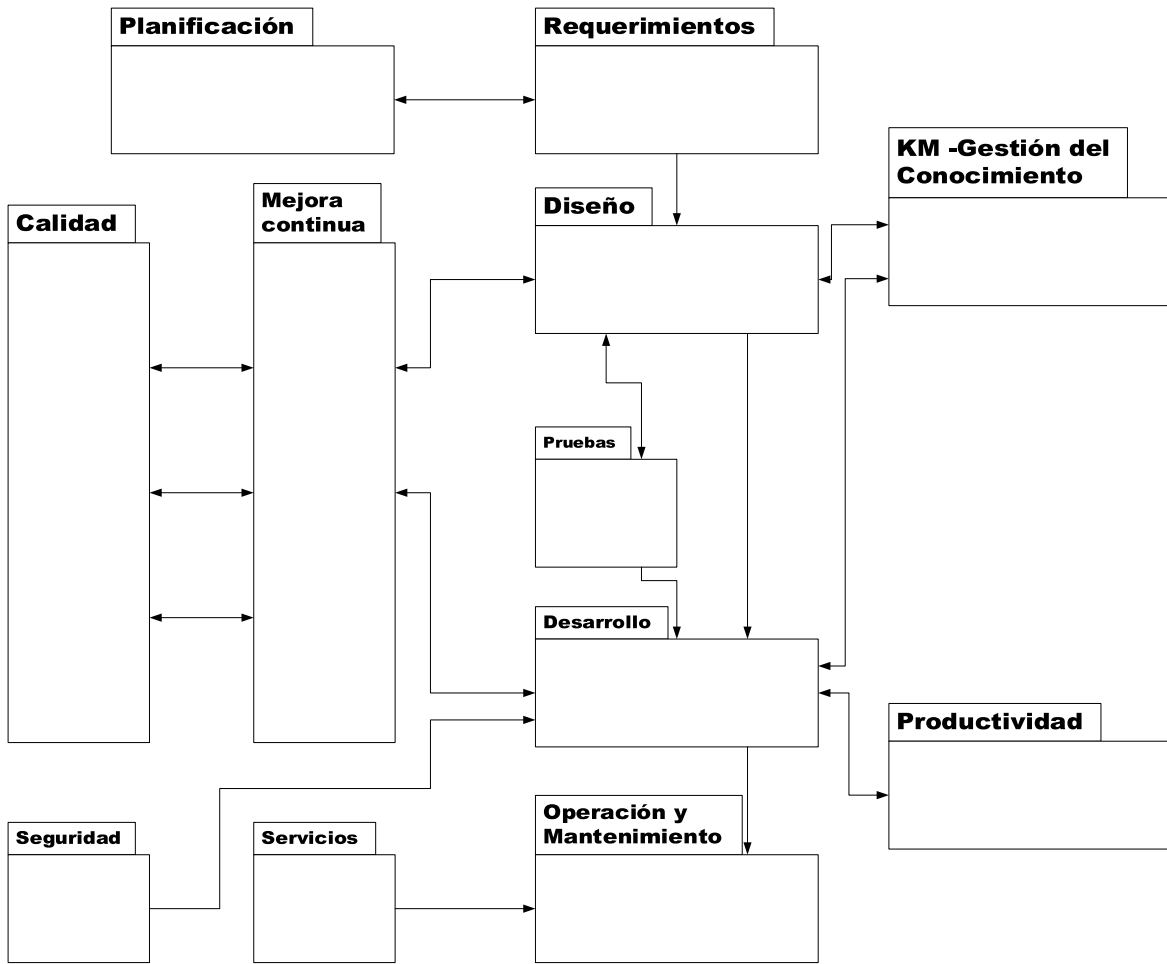


Ilustración 16: Modelo de relaciones entre categorías

Fuente: Elaboración propia

El anexo 8.7 proporciona una visión completa de las categorías con sus elementos y sus relaciones.

Por otro lado, durante la extracción de datos, se identificó un número de publicaciones con modelos de fábrica de software con elementos que satisfacen, de alguna forma, la definición del concepto y que, con el uso de procesos, muestran la forma de trabajar en una fábrica de software. Se marcó un total de 21 publicaciones que contaban con alguna materialización de fábrica de software, pero las siguientes nueve destacaron por su definición puntual en modelos: (Kulkarni & Reddy, 2005), (Pihoa, Tepandia, Roosta, Parmana, & Puusep, 2011), (Jalil & Bakar, 2017), (Brotto Furtado, 2006), (Demir, 2006), (Ávila, Supino Marcondes, Vieira, & Marques, 2010), (Montequín, Pérez, Fernández, & Balsera, 2013), (Stojanovski & Dzekov, 2009), (Domínguez, Pérez, Mendoza, & Grimán, 2005).

Con la ayuda de los modelos presentes en esas publicaciones y del conocimiento adquirido durante la revisión de literatura, se elaboró el modelo de procesos principales de fábrica de software, presentado en la Ilustración 17, en la página siguiente. Los elementos de color rojo representan una guía que permite asociar el modelo de procesos con el modelo de elementos.

La ilustración 17 muestra los elementos ordenados en los procesos generales del ciclo de vida de software (planificación, requerimientos, diseño, desarrollo, pruebas, operación y mantenimiento) agrupados por categoría, de manera que representan focos de conocimiento. Las líneas brindan la noción de la relación entre los focos de conocimiento en el proceso del ciclo de vida del software.

4.7 Revisión de objetivos

Para el desarrollo de la revisión de literatura se estableció un objetivo general y tres objetivos específicos, además de preguntas de investigación y metas, cuya relación se observa en la tabla 15:

Tabla 15: Objetivos, Preguntas y Metas

Objetivo General	Objetivos específicos	Preguntas de investigación	Metas
Modelar los componentes más comunes de fábrica de software para prestación de servicios de TI en la construcción de software que sirva como material de referencia condensada y moderna, a través de una revisión de literatura.	Describir los elementos del esquema de fábrica de software más usados en la industria y sus variantes para enmarcar el concepto de fábrica de software	PI1: ¿Cuáles son los elementos/componentes que definen una fábrica de software?	-Obtener los componentes presentes en modelos de fábrica de software
		PI2: ¿Cuáles son los elementos/componentes más comunes?	
	Generar una clasificación de los elementos del esquema de fábrica de software que permita agruparlos según su relación	PI3: ¿En cuales categorías predominan estos componentes?	-Organizar los componentes a través de una abstracción
	Sintetizar un modelo de prestación de servicios de desarrollo de software basado en fábrica de software.	PI4: ¿Cuáles son los modelos de presentación de fábrica de software?	-Crear un diagrama de fábrica de software -Descripción de los elementos del diagrama

Fuente: Elaboración propia

Adicionalmente se estableció una estrategia para la lograr los objetivos a través del proceso de una serie de actividades e instrumentos que permitieran recolectar y sintetizar la información recolectada. La estrategia puede ser visualizada en la siguiente tabla:

Tabla 16: Objetivos, Actividades, Instrumentos y Metas

Objetivo específico	Actividades	Instrumento	Metas
Describir los elementos del esquema de fábrica de software y sus variantes para enmarcar el concepto de fábrica de software	Identificar y seleccionar las publicaciones.	Primo, Google Académico Criterios de selección de publicaciones	Obtener los componentes presentes en modelos de fábrica de software
	Extraer los componentes presentes en las fábricas de software	Tabla de evaluación de la calidad y extracción de datos	
Generar una clasificación de los elementos del esquema de fábrica de software que permita agruparlos según su relación	Agrupar los componentes	Tabla de Síntesis de Datos	Organizar los componentes a través de una abstracción
Sintetizar un modelo de prestación de servicios de desarrollo de software basado en fábrica de software.	Relacionar los elementos de fábrica de software	Tabla de Síntesis de Datos	Crear un diagrama de fábrica de software
	Justificar los elementos y su relación	Tabla de evaluación de la calidad y extracción de datos Tabla de Síntesis de Datos	Descripción de los elementos del diagrama

Fuente: Elaboración propia

La aplicación de los instrumentos dio como resultado la agrupación de los datos de manera que se pudiera manipular su presentación, con el fin de analizarlos y satisfacer las preguntas de investigación. El resultado final se resume en la tabla 17:

Tabla 17: Validación de Objetivos

Objetivos específicos	Preguntas de investigación	Respuesta
Identificar los elementos del esquema de fábrica de software más usados en la industria y sus variantes para enmarcar el concepto de fábrica de software	<ul style="list-style-type: none"> • PI1: ¿Cuáles son los elementos/componentes que definen una fábrica de software? 	<ul style="list-style-type: none"> – Sección 4.3 – Sección 8.4, Tabla 21
	<ul style="list-style-type: none"> • PI2: ¿Cuáles son los elementos/componentes más comunes? 	<ul style="list-style-type: none"> – Sección 4.4, Tabla 14
Generar una clasificación de los elementos del esquema de fábrica de software que permita agruparlos según su relación	<ul style="list-style-type: none"> • PI3: ¿En cuales categorías predominan estos componentes? 	<ul style="list-style-type: none"> – Sección 4.4, Tabla 12
Sintetizar un modelo de prestación de servicios de desarrollo de software basado en fábrica de software.	<ul style="list-style-type: none"> • PI4: ¿Cuáles son los modelos de presentación de fábrica de software? 	<ul style="list-style-type: none"> – Sección 4.5, Tabla 15 – Sección 4.6, Ilustraciones 16 y 17

Fuente: Elaboración propia

Esto concluye el capítulo IV, más adelante, el capítulo V desarrolla las conclusiones obtenidas por medio de la revisión de literatura y las recomendaciones sobre diversos temas relacionados a las fábricas de software.

5 Capítulo V: Conclusiones y Recomendaciones

5.1 Conclusiones

Desde su aparición pública en los años 60 y 70, el concepto de fábrica de software ha sido un tema recurrente en el desarrollo de software. Sin embargo, su definición en la práctica difiere entre publicaciones, o al menos no tiene un consenso entre los interesados. La definición original dada por R.W. Bemer, y posteriormente por Cusumano, se ha transformado hasta convertirse en la definición más actual de Greenfield, Short y Cook, aunque compleja y abstracta, es la más referenciada dentro de las publicaciones. Lo que sí se puede asegurar es que componentes reusables (diseños, arquitecturas, componentes de software y conocimiento), líneas de producción y ensamblaje, y herramientas de automatización son parte fundamental de lo que pudiera llamarse una fábrica de software.

Los 64 elementos identificados pueden servir como punto de referencia para quienes tengan interés en implementar un modelo de fábrica de software. El lector puede estar en un rango que va desde las organizaciones neófitas en la industria hasta los veteranos en la práctica. Para los primeros brinda un panorama del terreno mientras que los posteriores lo pueden utilizar como lista de chequeo para validar que camino ya tienen recorrido; a ambos les permite discriminar sus fortalezas y debilidades en las áreas de conocimiento y determinar cuál puede ser su punto de partida para una implementación.

Los aspectos económicos son incluidos en el entorno de las fábricas de software cuando se habla de economías de escala y de gama. El software parece no tener problema con la escala pues resulta fácil hacer copias de software, pero crear diferentes productos sin que haya un costo significativo y que, por el contrario, haya sinergia económica, parece no ser tan fácil, posiblemente por la amplitud de áreas de conocimiento evidenciadas en la recolección de datos. Otro aspecto que puede influir es la universalidad del software; se considera como respuesta a problemas de ámbitos muy diversos, desde farmacéuticas hasta la creación de video juegos.

Este “*problema*” de diversidad es atacado por las líneas de producción de software que acotan su uso a familias de software y que es transferido a al concepto fábrica de software. Para la producción de diversos tipos de producto, se requieren múltiples líneas de producción de software.

Una de las suposiciones fue que habría material diverso con el que se pudiera construir claramente un modelo de fábrica de software. Sin embargo la realidad fue que los modelos no fueron detallados lo suficiente, por lo que se tuvo que tomar diferentes partes de los modelos encontrados en múltiples publicaciones para organizar los procesos de fábrica de software de manera que fuera palpable su funcionamiento.

Uno de los elementos más relevantes según la revisión de literatura es el DSL como medio de modelar esas familias de productos contenidos en un mismo dominio. De esta manera, el problema se limita a un ámbito más pequeño de conceptos y procedimientos. Lo cual resulta útil especialmente al momento de desarrollar pues hay menos incertidumbre o las posibilidades de crear una solución son menos, por lo que facilita la elaboración de herramientas de generación de código. Existe una serie de dificultades al utilizar el DSL, como el alto costo de educación en el uso de DSL, alto costo de diseño, implementación y mantenimiento; características similares a los costos de un trabajo de artesano. A este punto, se puede decir que se resuelve la incomodidad de basar el desarrollo de software en la pericia del artesano programador, pero que posiblemente crea otra similar, la dependencia de la pericia del artesano modelador de DSL.

La calidad es un aspecto que se menciona reiteradamente a través de las publicaciones, en particular haciendo referencia a la promesa de entregar productos de software más rápido y de mejor calidad principalmente debido a la reutilización de componentes ya probados. A pesar de eso, la evidencia no deja claro cómo y dónde ejecutar tales actividades de control de calidad y aseguramiento de la calidad, solamente 2 artículos hacen mención superficial de estas actividades.

Las publicaciones sugieren que puede existir un grado de conflicto entre los elementos, en apariencia, el uso de MDA, Fabrica de Software de Microsoft y el

concepto de Fábrica de Software de Greenfield pueden ser mutuamente excluyentes en unos casos y como complementarios en otros.

5.2 Recomendaciones

Los 64 elementos encontrados podrían no representar la totalidad de los elementos a contemplar, un ejemplo son las actividades de calidad, que son vagamente mencionadas. Una revisión más exhaustiva a las publicaciones puede arrojar un mayor número de elementos descritos en el contexto de las investigaciones. Esta revisión más profunda podría revelar relaciones entre los elementos, detalles que permitan reducir el proceso a decisiones dicotómicas entre los elementos que son compatibles y los que no.

Los datos presentados pueden ser utilizados como insumo para un estudio de puntos de referencia ("*Benchmarking*") con el propósito de evaluar las prácticas en el mercado y las que se ejecutan en el entorno propio. De esta manera puede resultar más sencillo dibujar un camino a la implementación de un modelo de fábrica de software, ya sea tomando como eje central el concepto o los elementos ya presentes en la organización.

Sea cual sea el eje central, la evidencia muestra que existe un nivel de complejidad considerable en la implementación de fábricas de software, principalmente en el manejo de las abstracciones necesarias. Usualmente la estrategia para resolver problemas complejos es la división de este en fragmentos menos complejos. Del mismo modo, la implementación de la fábrica debería ser fragmentada, en procesos incrementales, apuntando a resultados modestos pero progresivos. Un plan piloto sencillo puede ser la mejor opción para evaluar el comportamiento de los procesos y de las personas.

Algunos ejemplos de modelos de fábrica de software muestran simplicidad al tratar el concepto de fábrica con ligereza, lo que sugiere que también está el camino de la sencillez si se es menos riguroso. Puede resultar útil determinar algunos

elementos nucleares del practica de fábrica y partir de ahí construyendo capas de procesos.

La compatibilidad entre los elementos de diferentes categorías es un estudio que puede resultar pertinente, especialmente en las áreas de planificación y modelado, para colaborar en mayor medida a facilitar el proceso de implementación.

Una investigación que permita definir donde, cuando y cuales actividades de calidad deben ejecutarse para obtener un producto calidad comprobable podría ser necesaria pues no queda claro de qué manera incorporarlas en los procesos de fábrica de software.

En seguida, el capítulo VI presenta una revisión concienzuda sobre el desarrollo de la revisión de literatura, tomando en cuenta los acontecimientos más relevantes en el trascurso de la planeación ejecución y conclusión. Se hace un análisis ejecutivo de lo que significó la realización de la revisión, el grado de cumplimiento del plan de trabajo inicial, principalmente el cumplimiento de objetivos y las justificaciones de las eventuales desviaciones. Se consignan aspectos que reflejen la vivencia y sus alcances desde la perspectiva propia.

6 Capítulo VI: Análisis retrospectivo

El desarrollo de la revisión planteo retos relacionados a la planificación, se realizaron ajustes a los objetivos de manera que definiera mejor el tema de estudio. El planteamiento inicial se basó en hipótesis e ideas preliminares. Estos cambios definieron el método de recolección de datos al final.

Aunque se pensó en un proceso secuencial de lectura y extracción, tempranamente se identificó la necesidad de realizar revisiones iterativas pues a medida que se avanzaba en conocimiento, más aspectos resaltaban a la vista y ampliaban la perspectiva del tema de fábricas de software. Posiblemente con mayor experiencia en el campo de la investigación, las iteraciones sean menos, pero necesarios.

La lectura al 100% de las publicaciones no fue posible y probablemente nunca lo sea, especialmente en libros y tesis. En estos casos se hizo una lectura selectiva de secciones que en apariencia contuvieran información relevante y a partir de ahí, leer más si era necesario, por ejemplo, la sección donde se describe la metodología fue omitida. Para artículos entre tres y cinco páginas, la lectura completa era viable y se ejecutó de esa manera.

Clasificar los elementos fue una tarea algo difícil debido a que algunos son cuerpos de conocimiento, como DSLs, MDA o SPL. Por lo que se debió realizar pequeñas investigaciones informales para aclarar esos términos y poder clasificarlos apropiadamente.

El desarrollo del modelo de procesos fue una idea que surgió de la evaluación de los objetivos durante el proceso de investigación. Al ver que no había un aterrizaje más concreto de los conceptos, se tomó la decisión de hacerlo para beneficiar a los futuros investigadores a familiarizarse con la implementación.

Como se mencionó en la justificación, se espera que la revisión sirva como material de referencia, de consulta moderna que colabore con la desmitificación del

desarrollo de software, se considera que se lograron estos objetivos al dar una visión de los elementos necesarios para lograrlo y una propuesta de ordenamiento de procesos.

7 Bibliografía

- Alavi, M., Smith, R. H., & Leidner, D. (6 de 1999). Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues.
- Alwazae, M. (2015). Template-driven Documentation of Best Practices. Stockholmo, Suecia.
- Barakat, H. H. (2003). Scale economies, scope economies, and structure conduct performance relationships in Jordanian banking system, 1990–2000. Colorado, EEUU: ProQuest Dissertations Publishing. Recuperado el 21 de 11 de 2018, de <https://search-proquest-com.una.idm.oclc.org/docview/305332218?accountid=37045>
- Bekmamedova, N., Byrne, A., Cecez-Kecmanovic, D., Gilliland, A., Given, L. M., Haddow, G., . . . Williamson, K. (2018). *Research Methods: Information, Systems, and Contexts* (2 ed.). Chandos Publishing.
- Borg, M. (2012). Advancing trace recovery evaluation: Applied information retrieval in a software engineering context. Suecia.
- Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2006). Lessons from applying the systematic literature review process within the software engineering domain. *The Journal of Systems and Software* 80, 571–583.
- Brotto, A. (2006). Sharpludus: Improving Game Development Experience Through Software Factories And Domain-Specific Languages.
- CMMI Institute. (2013). *CMMI® para Servicios, Versión 1.3*.
- Cornford, T., & Shaikh, M. (2013). Introduction to information systems. Londres, Inglaterra: University of London. Recuperado el 21 de 11 de 2018, de http://www.dphu.org/uploads/attachements/books/books_3326_0.pdf
- Creswell, J. W. (2014). *Research design : qualitative, quantitative, and mixed methods approaches* (4 ed.). SAGE Publications.
- Cusumano, M. A. (1987, 09 25). The "Factory" Approach to Large-scale Software Development: Implication's for Strategy, Technology, and Structure. MIT Sloan School of Management. Retrieved 10 21, 2018, from <https://dspace.mit.edu/bitstream/handle/1721.1/47999/factoryapproacht00cusu.pdf?sequence=1>
- Cusumano, M. A. (31 de 5 de 1987b). Hitachi: Pioneering The "Factory" Model For Large-scale Software Development. MIT Sloan School of Management. Recuperado el 21 de 10 de 2018, de <https://dspace.mit.edu/bitstream/handle/1721.1/9563/JP-WP-87-07.pdf?sequence=1>
- Cusumano, M. A. (24 de 10 de 1989). The Factory Approach to Software Development: A Strategic Overview. MIT Sloan School of Management. Recuperado el 24 de 10 de 2018, de

<https://dspace.mit.edu/bitstream/handle/1721.1/2281/SWP-3088-24854433.pdf?sequence=1>

- Cusumano, M. A. (1991, 8 26). *Shifting Economies: From Craft Production Shifting Economies: From Craft Production*. MIT Sloan School of Management. Retrieved 10 21, 2018, from <https://dspace.mit.edu/bitstream/handle/1721.1/2367/SWP-3325-24663534.pdf?sequence=1>
- da Silva, F. Q., Santos, A., Soares, S., França, C., Monteiro, C., & Maciel, F. (2011). Six years of systematic literature reviews in software engineering: An updated tertiary study. Brazil.
- DeLuca, T. A. (2012). Spending Changes And Scale Economies When School Districts Consolidate Services. Michigan State University. Recuperado el 21 de 11 de 2018, de <https://search-proquest-com.una.idm.oclc.org/docview/1012287201?accountid=37045>
- Demir, A. (2006). Comparison of Model-Driven Architecture and Software Factories in the Context of Model-Driven Development. *Fourth Workshop on Model-Based Development of Computer-Based Systems and Third International Workshop on Model-Based Methodologies for Pervasive and Embedded Software (MBD-MOMPES'06)*.
- Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. IEEE Software.
- Ghezzi, C., Jazayeri, M., & Mandrioli, D. (2003). *Fundamentals of software engineering*. Prentice Hall.
- Gill, M. A. (12 de 2005). THE EFFECT OF OUTSOURCING ON FIRM VALUE. Arizona: ARIZONA STATE UNIVERSITY.
- Glass, R. L. (2001, 5). Frequently Forgotten Fundamental Facts about Software Engineering. (R. L. Glass, Ed.) IEEE Software.
- Greenfield, J., & Short, K. (2003). Software factories assembling applications with patterns, models, frameworks and tools.
- Greenfield, J., & Short, K. (10 de 2003). Software Factories: Assembling Applications With Patterns, Frameworks, Models And Tools. Anaheim, California, EEUU.
- Greenfield, J., Short, K., Cook, S., & Kent, S. (2004). *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools* (1st ed.). Wiley.
- Guzman, J. C., Chiozzib, G., Bridgerc, A., & Ibsend, J. (2014). The cost of developing and maintain the monitoring and control software of large ground-based telescopes.
- Humphrey, W. S. (11 de 2000). The Team Software Process (TSP). US Dept of the Air Force.
- Humphrey, W. S., Ferguson, P., Khajenoori, S., Macke, S., & Matvya, A. (1997). Results of Applying the Personal Software Process.
- IEEE Standards Board. (1990). IEEE Standard Glossary of Software Engineering Terminology.
- ISO. (26 de 3 de 2019). *All about ISO*. Obtenido de <https://www.iso.org/about-us.html>

- Khan, S., Niazi, M., & Ahmad, R. (2008). Systematic Literature Review Protocol for a Software Outsourcing Vendors Readiness Model (SOVRM). Pakistan.
- Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and Software Technology* 55, 2049–2075.
- Kitchenham, B., & Charters, S. (9 de 7 de 2007a). Guidelines for performing Systematic Literature Reviews in Software Engineering. UK.
- Kitchenham, B., & Charters, S. (2007b). Guidelines for performing Systematic Literature Reviews in Software Engineering. United Kingdom.
- Kitchenham, B., Turner, M., Niazi, M., Brereton, O., Budgen, D., & Linkman, S. (2010). Systematic literature reviews in software engineering – A tertiary study. *Information and Software Technology*, 52, 792–805.
- Krajnc, A., Andrej, M., Gerlec, Č., Goljat, U., & Polančič, G. (2012). Experimental investigation of the quality and productivity of Software Factories based development. doi:0.2298/CSIS110720007K
- Lajmi, A., Cauvin, S., & Ziane, M. . (2009). A Software Factory for the Generation of CAPE-OPEN compliant Process Modelling Components.
- Laplante, P. A. (2007). *What Every Engineer Should Know about Software Engineering*. CRC Press.
- Li, C., Li, H., & Li, M. (2001). A Software Factory Model Based on ISO9000 and CMM for Chinese Small Organizations.
- Menolli, A., Belmonte, D., Sgarbi, E., Malucelli, M., & Reinehr, S. (2011). Práticas Do Modelo MPS Em Fábricas De Software: Um Estudo Exploratório Sobre As Percepções Dos Gerentes De Projeto. Brazil.
- Montequín, V., Pérez, C., Fernández , F., & Balsera, J. (2013, 01 01). Scorecard and KPIs for monitoring software factories effectiveness in the financial sector. *International Journal of Information Systems and Project Management*, 29-43.
- Muños, J., & Pelechano, V. (2005). Building a Software Factory for Pervasive Systems Development. *Lecture Notes in Computer Science*, 3520. Springer, Berlin, Heidelberg. doi:https://doi.org/10.1007/11431855_2
- Northrop, L., & Clements, P. (12 de 2012). A Framework for Software Product Line Practice , version 5.0.
- Object Management Group. (2019). *MDA FAQ*. Recuperado el 06 de 03 de 2019, de OMG: https://www.omg.org/mda/faq_mda.htm
- Olapurath, R. (1998). Organization of the firm's activities: A pragmatic research paradigm extension of transaction cost theory. Obtenido de <https://search-proquest-com.una.idm.oclc.org/docview/304447786?accountid=37045>

- Owusu-Nyamekye, K. (7 de 2016). *It Outsourcing Strategy: Public Service Management Perspectives*. Columbia, Canada: ProQuest LLC.
- Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). *Systematic Mapping Studies in Software Engineering*. Suecia.
- Pop, A. (June de 2005). *Contributions to Meta-Modeling Tools and Methods*. Obtenido de 0280-7971
- RAE. (s.f.). Recuperado el 6 de 03 de 2019, de <https://dle.rae.es/?id=PTk5Wk1>
- Randolph, J. (6 de 2009). A Guide to Writing the Dissertation Literature Review. *Practical Assessment, Research & Evaluation*, 14(13). Obtenido de <https://pareonline.net/getvn.asp?v=14&n=13>
- Shah, U. S. (2016). An Excursion to Software Development Life Cycle Models: an Old to Ever-growing Models. *ACM SIGSOFT*, 41(1). doi:10.1145/2853073.2853080
- Singh, S., & Zack, M. H. (22 de 3 de 2006). Information Technology Outsourcing: Reducing Costs or knowledge? Recuperado el 22 de 11 de 2018, de <https://pdfs.semanticscholar.org/2316/5002f088e5010ebeba7050d22ac19c30d049.pdf>
- Software Engineering Institute. (2010). *CMMI® para Desarrollo*, Versión 1.3.
- Sommerville, I. (2007). *Software Engineering* (8 ed.). Pearson Education Limited.
- Stapić, Z., de-Marcos, L., Strahonja, V., García-Cabot, A., & García López, E. (2016). Scrutinizing Systematic Literature Review Process in Software Engineering. doi: 10.18421/TEM51-16
- Staron, M. (2003). Customizing UML with Stereotypes.
- Stojanovski, T., & Dzekov, T. (2009). *Rapid Application Development Using Software Factories*. Skopje, Macedonia. Recuperado el 21 de 11 de 2018, de <https://arxiv.org/una.idm.oclc.org/abs/1201.0853>
- Systematic literature reviews in software engineering. (2013). *Information and Software Technology*, 55, 919–920.
- Szyperski, C., Gruntz, D., & Murer, S. (2002). *Component software: beyond object-oriented programming*. Pearson Education.
- The Standish Group International, Inc. (2015). *CHAOS Report 2015*. Recuperado el 21 de 11 de 2018, de https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf
- Vale, T., Crnkovic, I., Santana de Almeida, E., da Mota Silveira Neto, P., Cerqueira Cavalcant, Y., & de Lemos Meira, S. (2016). Twenty-eight years of component-based software engineering. Brasil.
- Williamson, K., & Johanson, G. (2018). *Research Methods Information, Systems, and Contexts*. Chandos Publishing .

Yates Smith, R. R. (28 de 9 de 2005). OUTSOURCING OF INFORMATION TECHNOLOGY: FUTURE ADOPTION PLANS OF U.S. CORPORATIONS. Chicago: Faculty of Argosy University-Chicago.

8 Anexos

En esta sección se encuentra información en bruto o adicional que complementa el proceso de investigación descrito en las secciones anteriores.

8.1 Detalle de resultados de búsqueda:

Resultados detallados de la aplicación de los criterios de búsqueda

Tabla 18: Detalle de resultado de búsqueda

Motor de búsqueda	Cadena de búsqueda	Aplicado	Tipo de publicación	Cantidad de resultados
PRIMO	("software factory" OR "Software factories") AND (model OR implementation OR experience OR components, OR elements OR lessons OR "lessons learned" OR Framework OR "good practices")	Titulo	Actas de Conferencia	1
PRIMO	("software factory" OR "Software factories") AND (model OR implementation OR experience OR components, OR elements OR lessons OR "lessons learned" OR Framework OR "good practices")	Titulo	Articulo	2
PRIMO	("software factory" OR "Software factories") AND (model OR implementation OR experience OR components, OR elements OR lessons OR "lessons learned" OR Framework OR "good practices")	Titulo	Libros	1

Motor de búsqueda	Cadena de búsqueda	Aplicado a	Tipo de publicación	Cantidad de resultados
PRIMO	("fábrica de software" OR "Fabricas de Software") AND (modelo OR implementación OR experiencia OR componentes, OR elementos OR lecciones OR "lecciones aprendidas" OR marco OR "marco de trabajo" OR "buenas practicas")	Titulo	Articulo	1
PRIMO	("fábrica de software" OR "Fabricas de Software") AND (modelo OR implementación OR experiencia OR componentes, OR elementos OR lecciones OR "lecciones aprendidas" OR marco OR "marco de trabajo" OR "buenas practicas")	Titulo	Recursos de texto	2
GOOGLE ACADÉMICO	allintitle: model OR implementation OR experience OR components OR elements OR lessons OR "lessons learned" OR Framework OR "good practices" OR "Software Factory"	Titulo	Articulo	23

Motor de búsqueda	Cadena de búsqueda	Aplicado	Tipo de publicación	Cantidad de resultados
GOOGLE ACADÉMICO	allintitle: model OR implementation OR experience OR components OR elements OR lessons OR "lessons learned" OR Framework OR "good practices" "Software Factories"	Titulo	Articulo	9
GOOGLE ACADÉMICO	allintitle: modelo OR implementación OR experiencia OR componentes OR elementos OR lecciones OR "lecciones aprendidas" OR "marco de trabajo" OR "buenas practicas" "fábrica de software"	Titulo	Articulo	17
GOOGLE ACADÉMICO	allintitle: modelo OR implementación OR experiencia OR componentes OR elementos OR lecciones OR "lecciones aprendidas" OR "marco de trabajo" OR "buenas practicas" "fábricas de software"	Titulo	Articulo	5

Fuente: Elaboración propia

8.2 Detalle de publicaciones primarias

El siguiente cuadro muestra las publicaciones obtenidas

Tabla 19: Detalle de publicaciones primarias

ID	Tipo publicación	Titulo	Autores	Año	Idioma	Cantidad de Paginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Paginas
1	Articulo	Modelo De Relacionamiento Estratégico Entre Factores Organizacionales Y Técnicos En Fábricas De Software Colombianas	Karina J. Cerón y Hugo F. Arboleda	2018	Español	9	EBSCO	25			2,78
2	Articulo	A Scorecard Framework Proposal For Improving Software Factories' Sustainability: A Case Study Of A Spanish Firm In The Financial Sector	César Álvarez , Vicente Rodríguez , Francisco Ortega and Joaquín Villanueva	2015	Ingles	23	MDPI	52			2,26

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
3	Artículo	Making The Software Factory Work: Lessons From A Decade Of Experience	Harvey P. Siy, James D. Herbsleb, Audris Mockus, Mayuram Krishnan , George T. Tucker	2001	Ingles	10	ResearchGate	13			1,30
4	Artículo de conferencia	A Software Factory Model Based On ISO9000 And CMM For Chinese Small Organizations	Li, Chao Li, Huaizhang Li, Mingshu	2001	Ingles	5	IEEEExplore	16		Tomado el 20/02/2019, dejo de ser publico el 21/02/2019	3,20
5	Artículo de conferencia	A Model Driven Software Factory Using Domain Specific Languages	Warmer, Jos	2007	Ingles	9	Google books	5		Parcial. Conclusión y lecciones aprendidas completas	0,56

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
6	Articulo	MDSOFA: A Model-Driven Software Factory	Benoît Langlois Daniel Exertier	2004	Ingles	10	ResearchGate	21		Varias fuentes. Posible conferencia (workshop)	2,10

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
7	Artículo	"Free" Innovation Environments: Lessons Learned From The Software Factory Initiatives	Davide Taibi, Valentina Lenarduzzi, Fabian Fagerholm, Jürgen Münch, Muhammad Ovais Ahmad, Kari Liukkunen, Sami Pietinen, Markku Tukiainen, Sami Pietinen, Markku Tukiainen, Carlos Fernández-Sánchez, Juan Garbajosa, Carlos Fernández-Sánchez, Juan Garbajosa	2015	Inglés	6	REUTL NGEN UNIVER SITY ACADE MIC BIBLIOG RAPHY	14		ejemplos de SF de academia en varios lugares de Europa	2,33

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
8	Artículo de conferencia	A Model For Defining Software Factory Processes	Luzia Nomura, Mauro de Mesquita Spínola, Antonio Carlos Tonini, Oswaldo Keiji Hikage	2007	Ingles	6	Academia.edu	19		Modelo de procesos de SF	3,17
9	Artículo	Integrating Analytic Hierarchy Process (AHP) And Balanced Scorecard (BSC) Framework For Sustainablebusiness In A Software Factory In The Financial Sector	César Álvarez Pérez, Vicente Rodríguez Montequín , Francisco Ortega Fernández and Joaquín Villanueva Balsera	2017	Ingles	16	MDPI	50		KPI para SF	3,13
10	Artículo de conferencia	Enterprise Business Application Product Line As A Model Driven Software Factory	Vinay Kulkarni, Sreedhar Reddy	2005	Ingles	4	ResearchGate	11		SF modelo alto nivel	2,75

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
11	Artículo de conferencia	From Archetypes Based Domain Model Via Requirements To Software: Exemplified By Lims software Factory	Gunnar Pihoa, Jaak Tepandia, Mart Roosta, Marko Parmana and Viljam Puusep	2011	Ingles	6	ResearchGate	30		Zachman DSL	5,00
12	Artículo	Integration Of Balanced Scorecard (BSC), Strategy Map, And Fuzzy Analytic Hierarchy Process (FAHP) For A Sustainability Business Framework: A Case Study Of A Spanish Software Factory In The financial Sector	César Álvarez Pérez, Vicente Rodríguez Montequín, Francisco Ortega Fernández, Joaquín Villanueva Balsera	2017	Ingles	23	MDPI	77		BSC	3,35
13	Artículo	Adapting Software Factory Approach Into Cloud ERP Production Model	Dzulkafli Jalil, Muhamad Shahbani Abu Bakar	2017	Ingles	9	Academia.edu	43		SF para la servicios de la nube	4,78

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
14	Artículo de conferencia	A Practical Example Of A Software Factory: Building A Custom Application For Analysing EU Cyber Physical System (CPS) Projects Using Open Source Software Components	P. J. Palmer, C. E. Siemieniuch, M. A. Sinclair, M. J. De Henshaw	2016	Ingles	15	CORE	16	x	sistema construido con SF	1,07
15	PPT	Experimental Software Engineering: Software Factory Model	Ralph-Johan Back, Ivan Porres and Pekka Abrahamsson	2006	Ingles	40	Åbo Akademi Universit y	1		PPT de Modus operandi de SF	0,03
16	Articulo	An Ontology For Classifying, Searching And Matching Software Components In A Software Factory Framework	Francisco E. Castillo-Barrera and Héctor A. Duran-Limon	2015	Ingles	17	ResearchGate	42		Herramientas en una SF	2,47

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
17	Artículo	A Framework For Semi-Automated Measurement Of A Software Factory Productivity	Andrea Bei, Fabio Rabini, Giovanni Ricciolio	2007	Ingles	19	ResearchGate	15		Método de estimación de tamaño, producto en SF	0,79
18	Artículo de conferencia	A Software Factory For The Generation Of CAPE-OPEN Compliant Process Modelling Components	Amine Lajmi, Sylvie Cauvin, Mika Ziane	2009	Ingles	6	Academia.edu	11		Generación de código con modelos	1,83
19	Tesis	Cloud Enterprise Resource Planning Development Model Based On Software Factory Approach	DZULKAF LI ABD JALIL	2018	Ingles	323	UUM Electronic Theses and Dissertation	201		Infraestructura de SF en la nube	0,62
20	Artículo	Applying Software Factories To Pervasive Systems: A Platform Specific Framework	Javier Munoz, Vicente Pelechano	2006	Ingles	6	ResearchGate	11		solución SF, bueno!	1,83

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
21	Tesis	Sharpludus: Improving Game Development Experience Through Software Factories And Domain-Specific Languages	André Wilson Brotto Furtado	2006	Ingles	137	CiteSeer X	140		Elementos de SF, pág. 46, bueno	1,02
22	Artículo de conferencia	Comparison Of Model-Driven Architecture And Software Factories In The Context Of Model-Driven Development	Ahmet Demir	2006	Ingles	9	ftp://ftp.inf.puc-rio.br/pub/docs/FomularioSolicitud/LeandraMara-06-08-23494.pdf	10		SF vs MDA leer primero	1,11
23	Articulo	MDA (Model-Driven Architecture) As A Software Industrialization Pattern: An Approach For A Pragmatic Software Factories	Thomas Djotio Ndie, Claude Tangha, Fritz Ekwoge Ekwoge	2010	Ingles	11	Academia.edu	23		MDA as SF	2,09

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
24	Artículo de conferencia	Knowledge Management Model Applied To A Complex System: Development Of Software Factories	Víctor Hugo Medina García, Ticsiana Lorena Carrillo, and Sandro Javier Bolaños Castro	2013	Ingles	10	ResearchGate	20	x	Malo, Español: http://laccei.org/LACCEI2011-Medellin/ReferedPapers/CTE234_Carrillo.pdf	2,00
25	Artículo	Application Of Model Psp Manual And Supported By Tool Marries In A Study Of Case Of Brazilian Plant Of Software	Denis Ávila Montini, Mauro de Mesquita Spinola, José Benedito Sacomano, Marcos Nascimento, Danilo Battaglia	2006	Portugués	9	Ecorfan	12		PSP para cumplir con CMMI2 en una FS	1,33

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
26	Articulo	Implementación Del Modelo CMMI Para La Creación De Fábrica De Software De La UTSJR	Fidencio Díaz, Gregorio Rodríguez, Alejandro Valencia, Héctor Saldaña y Miguel González	2013	Español	8	Ecorfan	6		CMMI 2 para definir unas SF	0,75
27	Articulo	Un Marco De Trabajo De Una Fábrica De Software Para El Reuso Del Diseño Arquitectónico Y De Componentes De Software	HÉCTOR A. DURÁN-LIMÓN, MARÍA ELENA MEDA-CAMPAÑA, ALMA LILIA SAPIEN-AGUILAR, LAURA CRISTINA PIÑÓN-HOWLET	2008	Español	16	Tecnocencia chihuahua	25		framework para reusar de diseños y software	1,56

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
28	Artículo de conferencia	Gestión De Requerimientos A Través Del Modelo De Fábrica De Software Aplicando Metodologías Ágiles	Lisset Alexandra Neyra, María Paula Espinoza, María Patricia Samaniego	2017	Español	20	RedClara	10		Forma de lidiar con los Requerimientos	0,50
29	Tesis	Saber Social E Desenvolvimento De Software: Avaliação Crítica Do Modelo Da Fábrica De Software	Leandra Leal	2008	Portugués	116	Universidade Federal de Minas Gerais	46		Crítica a las FS	0,40
30	Artículo de empresa	Modelo De Productividad Basado En Componentes Para La Fábrica De Software	Pedro S. Castañeda Vargas, Luis A. Guevara Sandoval	??	Español	6	Guevaracia.	4		Malo, empresa	0,67

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
31	Trabajo Final	Diseño, Desarrollo E Implementación De Un Sistema Informático Para El Control De Proyectos Y Manejo De Recursos Humanos Del Área De Fabrica De Software Para La Empresa Tata Consultancy Services Ecuador En El Año 2015	CHRISTIAN JAVIER CALDERÓN GÁNDARA	2015	Español	92	Repositorio de la Universidad Internacional SEK Ecuador	16	x	malo, registro de proyecto	0,17
32	Trabajo Final	Elaboración E Implementación De Una Aplicación Web Que Maneje El Tablero De Requerimientos Y Proyectos De La Fábrica De Software Globalhitss Sede Colombia Integrada Al Sistema De Recursos Compartidos En Sharepoint	ESTEBAN NICOLÁS JIMÉNEZ SÁENZ	2015	Español	37	Biblioteca Alberto Quijano Guerrero - Universidad de Nariño	17	x	malo	0,46

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
33	Artículo de conferencia	Estudio De Viabilidad Para Implementación De Una Fábrica De Software En El Municipio De Ixtlahuaca, Edo. De Méx, 2009	Juan Alberto Ruiz Tapia Karina Gonzalez Roldán Silvio Pérez	2009	Español	17	Red Internacional de Investigadores en Competitividad	17	x	Malo	1,00
34	Trabajo Final	Trabajo De Investigación Sobre Aplicación De Métodos De Búsqueda Por Heurística Sobre El Proceso De Administración De Asignaciones De Recursos Humanos A Proyectos De Tecnologías De Información Bajo El Marco De Trabajo De Una Fábrica De Software	Pérez Aquino, Daniel Humberto	2017	Español	32	Repositorio académico UPC	9	x	Malo	0,28

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
35	Trabajo Final	Implementación De Una Fábrica De Software Con CMMI Nivel 2-Cliente AFP	Santa Cruz López	2011	Español	94	Universidad Nacional de Ingeniería - Biblioteca Central	13	x	CMMI 2 para definir unas SF	0,14
36	Articulo	Aplicação Do Modelo Fábrica De Software CMMI Especializado Em Data Warehouse Com Técnicas De Heurísticas Para A Definição De Soluções	Denis Ávila Montini, Francisco Supino Marcondes, Luiz Alberto Vieira Dias, Adilson Marques da Cunha	2014	Portugués	8	ResearchGate	20		CMMI 2 para definir unas SF	2,50
37	Articulo	Fábrica De Software Modelo: Uma Metodologia De Ensino E De Prestação De Serviços	Carlos Roberto Augusto Junior Clávison M. Zapelini	2014	Portugués	10	Portal de Periódicos Unisul	4	x	FS académica para desarrollar un software	0,40

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
38	Articulo	Um Relato De Experiencia Da Implantacao De Um Modelo De Fabrica De Software Escola (Fases)	Edmilson Barbalho Campos Neto, Alba Sandyra Bezerra Lopes, Diego Silveira Costa Nascimento	2017	Portugués	10	ResearchGate	9	x	FS académica para desarrollar un software	0,90
39	Articulo	Práticas Do Modelo MPS Em Fábricas De Software: Um Estudo Exploratório Sobre As Percepções Dos Gerentes De Projeto	Andre Menolli, Danillo Belmonte, Edilson Sgarbi, Andreia Malucelli, Sheila Reinehr	2011	Portugués	15	Programa de Pós-Graduação em Informática Pontifícia Universidade Católica do Paraná	24		Visión de Administrador Proyectos sobre FS	1,60

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
40	Artículo de conferencia	Modelo De Gestión Del Conocimiento Aplicado A Un Sistema Complejo: Desarrollo De Fábricas De Software	Ticsiana Lorena Carrillo Bovea Víctor Hugo Medina García	2011	Español	10	LACCEI	18	x	KM mediante sistemas complejos Malo	1,80
41	Artículo	Una Aproximación A Un Modelo De Gestión De Conocimiento Aplicable A Las Pequeñas Y Medianas Fábricas De Software	Adriana Maulini Luciano Straccia María F. Pollo C.	2018	Español	6	ResearchGate	41		Gestión del conocimiento en un FS	6,83
42	Artículo	Um Modelo De Controle Formal Para O Gerenciamento De Riscos De Processo Em Fábricas De Software	Strauss Cunha Carvalho, Danilo Douradinho Fernandes, Danilo Douradinho Fernandes, Paulo Tasinaffo	2009	Portugués	2	ResearchGate	7		CMMI- Dev Riesgos en FS, MALO	3,50

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
43	Artículo	Experimental Investigation Of The Quality And Productivity Of Software Factories Based Development	Andrej Krajnc, Marjan Heričko, Črt Gerlec, Uroš Goljat, Gregor Polančič	2012	Ingles	24	EBSCO	41		Productividad y calidad de SF vs tradicional	1,71
44	Artículo	Scorecard And KPIs For Monitoring Software Factories Effectiveness In The Financial Sector	Vicente Rodríguez Montequín, Francisco Ortega Fernández, Joaquín Villanueva Balsera, César Álvarez Pérez	2013	Ingles	14	DOAJ	18		Scorecard y KPIs para SF	1,29
45	Artículo	Rapid Application Development Using Software Factories	Toni Stojanovski, Tomislav Dzekov	2012	Ingles	13	Cornel University	13		Experiencia SF vs tradicional	1,00
46	Artículo de empresa	A Foundation For The Pillars Of Software Factories	Tom Fuller	2009	Ingles	14	Microsoft	5		Elementos básicos de FS	0,36

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
47	Articulo	An Approach Toward A Software Factory For The Development Of Educational Materials Under The Paradigm Of WBE	Rubén Peredo Valderrama, Alejandro Canales Cruz, Iván Peredo Valderrama	2011	Ingles	12	LaernTechLib	21		Arquitectura de SF	1,75
48	Articulo de conferencia	A Platform For Teaching Applied Distributed Software Development: The Ongoing Journey Of The Helsinki Software Factory	Fabian Fagerholm, Nilay Oza, Jurgen Munch	2013	Ingles	6	Cornel University	12		SF académica	2,00
49	Articulo de conferencia	Towards A Shared Software Factory: An Industrial Experience	P. D'Ambrosio, R. Francese, A. Marino, I. Passero, G. Tortora	2011	Ingles	6	AMC	21		recursos y PM entorno a SF	3,50
50	Articulo	Domain-Driven Development: The Smarttools Software Factory	Didier Parigot, Carine Courbis y	2005	Ingles	18	HAL				0,00

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
51	Artículo	Computer Games Software Factory And Edutainment Platform For Microsoft .NET	A.W.B. Furtado, A.L.M. Santos and G.L. Ramalho	2007	Ingles	14	IEEEExplore	32		En juegos usando .net. Posible ejemplo de top SF modelo	2,29
52	Libro	Devops For Digital Leaders: Reignite Business With A Modern DevOps-Enabled Software Factory	Aruna Ravichandran, Kieran Taylor, Peter Waterhouse	2016	Ingles	176	AOPEN	69	x	Libro de DevOps. Cap4 puede ser útil	0,39
53	Artículo	Managing Time In A Software Factory: Temporal And Spatial Organization Of IS Development Activities	Joe Nandhakumar	2002	Ingles	12	EBSCO	30		Manejo de espacio en la Fabrica de software	2,50
54	Artículo	Software Factories Assembling Applications With Patterns, Models, Frameworks And Tools	Jack Greenfield Keith Short	2003	Ingles	11	AMC	30		SF	2,73

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
55	Articulo	Quality In Development Process For Software Factories According To ISO15504	Kenyer Domínguez, María Pérez, Luis E. Mendoza, Anna Grimán	2005	Ingles	15	ResearchGate	21		Calidad de SO asociado al ISO15504	1,40
56	Articulo de conferencia	The Software Factory: Combining Undergraduate Computer Science And Software Engineering Education	Tvedt, J.D ; Tesoriero, R ; Gary, K.A	2001	Ingles	8	IEEEExplore	16	x		2,00
57	Articulo	Las Fábricas De Software En España: Organización Y División Del Trabajo. El Trabajo Fluido En La Sociedad De La Información	Juan José Castillo	2009	Español	39	DOAJ	160		tendencias de desarrollo de software	4,10

ID	Tipo publicación	Título	Autores	Año	Idioma	Cantidad de Páginas	Base de Datos	Cantidad de referencias bibliográficas	Excluido	Comentario	Cantidad de Referencias / Cantidad de Páginas
58	Artículo de conferencia	Towards A Domain Specific Application Development Environment For The Elegi Architecture: The Software Factories Approach	Angelo Gaeta, Pierluigi Ritrovato, Matteo Gaeta	2005	Inglés	5	HAL	29		Mapa de SF elementos	5,80
59	Artículo	Diálogos Sobre A Taylorização Da Atividade Dos Desenvolvedores Das Fábricas De Software	Dímitre Sampaio Moita Cássio Adriano Braz de Aquino Verônica Siqueira Araújo	2018	Portugués	13	DOAJ	21		Entrevista a desarrolladores	1,62

Fuente: Elaboración propia

8.3 Detalle de calificaciones de calidad

En la siguiente tabla se presenta el detalle de la evaluación de calidad para cada publicación primaria. Las publicaciones con espacios vacíos bajo las columnas C1-C4 fueron excluidas antes del proceso de revisión de calidad.

Tabla 20: Detalle de evaluación de calidad

ID	Calidad				
	Imparcial		Relevancia		Calificación
	C1	C2	C3	C4	Calidad
1	Y	Y	Y	P	3,5
2	Y	Y	Y	P	3,5
3	Y	Y	Y	P	3,5
4	Y	Y	Y	P	3,5
5	P	P	Y	P	2,5
6	Y	Y	Y	P	3,5
7	Y	Y	N	N	2
8	Y	Y	Y	P	3,5
9	Y	P	Y	P	3
10	Y	P	Y	P	3
11	Y	Y	Y	P	3,5
12	Y	Y	Y	P	3,5
13	Y	Y	Y	P	3,5
14					0
15	Y	Y	P	N	2,5
16	Y	Y	Y	P	3,5
17	Y	Y	Y	P	3,5
18	Y	Y	Y	P	3,5
19	Y	Y	Y	Y	4
20	Y	Y	Y	P	3,5
21	Y	Y	Y	Y	4
22	P	Y	Y	P	3

ID	Calidad				
	Imparcial		Relevancia		Calificación
	C1	C2	C3	C4	Calidad
23	Y	Y	Y	Y	4
24					0
25	Y	Y	Y	P	3,5
26	Y	Y	Y	N	3
27	Y	Y	Y	P	3,5
28	Y	Y	P	P	3
29	P	P	P	P	2
30	Y	P	Y	P	3
31					0
32					0
33					0
34					0
35					0
36	Y	Y	P	P	3
37	Y	Y	N	N	2
38	P	P	P	N	1,5
39	Y	Y	P	N	2,5
40	P	P	N	P	1,5
41	Y	Y	Y	P	3,5
42	Y	P	P	P	2,5
43	Y	Y	Y	P	3,5
44	Y	P	Y	P	3
45	Y	Y	Y	Y	4
46	Y	Y	Y	Y	4
47	Y	Y	Y	P	3,5
48	Y	P	P	P	2,5
49	Y	Y	P	P	3
50	P	P	Y	P	2,5
51	Y	P	Y	P	3
52	Y	P	N	N	1,5

ID	Calidad				
	Imparcial		Relevancia		Calificación
	C1	C2	C3	C4	Calidad
53	Y	Y	N	N	2
54	Y	Y	Y	Y	4
55	P	P	Y	Y	3
56					0
57	Y	Y	N	P	2,5
58	Y	Y	P	P	3
59	Y	Y	N	P	2,5

Fuente: Elaboración Propia

8.4 Detalle de datos recolectados

En seguida se muestra el detalle de los datos recolectados mediante la lectura de las publicaciones, realizada con el propósito de identificar los elementos de fábrica de software. Se puede distinguir cuales elementos pertenecen a cada publicación por medio del color de las filas y los ID.

Tabla 21: Detalle de datos recolectados

ID	DATOS						Comentario
	Marco de trabajo	Modelo	PI1		PI4		
			Procedimiento	Herramienta	Otro	Modelo de FS	
1		Alineación estratégica ad-hoc para SF				ISO9001, ISO20000, ISO270001	4 SF
1		CMMI				CMMI	CMMI3 CMMI4
2				BSC ad-hoc		SF industrial	Sistema de admin. de productividad de SF
2				KPIs para SF		SF como Empresa	
3			Mejora continua				Bueno para lecciones aprendidas
3			Análisis de causa raíz				

DATOS							
PI1				PI4			
ID	Marco de trabajo	Modelo	Procedimiento	Herramienta	Otro	Modelo de FS	Comentario
4	Gestión de Calidad orientado a procesos		CBD Desarrollo basado en componentes (SPL de SEI)			QM y SPL	QM ISO9000 CMM SPICE
4		SPL de SEI				ISO9000 CMM TSP/PSP = SFM-CSO[ISO/CMM]	Muy bueno para elementos de Framework Software Factory ::= <Management Specifications, Product-lines> X <Processes, People, Techniques> {Tool Support } (1)
4							Buen Modelo de SF
5				Microsoft DSL Tools		SF de Microsoft	
5				Generador de código		SF consume Assets	Buen modelo de SF
6		Modelo de Madures ah-hoc				SF MDSF: Model Driven Software Factory	Bueno para lecciones aprendidas

DATOS							
PI1						PI4	
ID	Marco de trabajo	Modelo	Procedimiento	Herramienta	Otro	Modelo de FS	Comentario
7			Kanban, Scrum	Jira, Rise Editor, Myeclipse, Apache Tomcat, PostgreSQL, Dreamweaver, and GitHub		SF académica	Herramientas para SF
8	PMBOK, RUP y otros	Modelo de Definición de procesos en SF				SF genérica	Buen Modelo de SF
9	BSC y AHP			SLA			AHP para poner peso a los indicadores de BSC. Satisfacción del usuario y SLA son los dos indicadores más importantes para una FSF
10			Instantiation, Transformation and Weaving	Generador de código		SF MDSF: Model Driven Software Factory	Buen Modelo de SF
10				SpecL		SF Empresarial	
11	Zachman	TDM Test Driven Modelling					Zachman + IS + TDM = DSL

DATOS							
PI1						PI4	
ID	Marco de trabajo	Modelo	Procedimiento	Herramienta	Otro	Modelo de FS	Comentario
11	XP extreme programming	TDD test driven development					Buen modelo de SF
11		DSL					
12	BSC						Sostenibilidad de SF
12	FAHP Fuzzy Analytic Hierarchy Process						
12	MCDM multicriteria decision-making						
13	WST Work System Theory	KM knowledge management	SPL			SF nube	Modelo de SF
13	RBT Resource base theory	Modelo Plataforma					Elementos SF Tabla 2
13		Workflow Model					
13		Product control model					

DATOS							
		PI1				PI4	
ID	Marco de trabajo	Modelo	Procedimiento	Herramienta	Otro	Modelo de FS	Comentario
14							
15						SF feedback loop	The Software Factory / University Hospital
15						SF académica	
16	QMS quality management system	CBSE			Facets para clasificar elementos de software		
16	TULUM Framework = XELHA + NEON + JADE				SCR software component repository		TULUM Framework = XELHA + NEON + JADE para búsqueda y validación en SCR
17	framework for the semi-automated measurement		FPA Function Point Analysis	www.pentaho.org			Modelo de productividad
17			Regresion lineal				
18			Generación de código	Editor Grafico			Modelo de Generación de código

DATOS							
PI1				PI4			
ID	Marco de trabajo	Modelo	Procedimiento	Herramienta	Otro	Modelo de FS	Comentario
19	QC quality control	KM knowledge management	SPL		DevOps	SF nube	
20	MDA		SPL	Generador de código		SF específica	Tres Formas de abstraer
20	DSL			Modelo-a-Modelo (AGG1 graph grammars engine)			
20				FreeMarker (generador de código basado en plantillas)			
21	DSL			.Net DSL tools		SF para juegos	Buen modelo de SF
21	PL análisis						
21	PL Diseño						
22	MDA	UML	SPL	Generador de código		MDA	MDA vs SF
22	MDD			ArcStyler MDA tool		SF genérica	Buena conclusión de SF
22				GME DSL tool			
23	MDA			Generador de código		MDA	MDA como SF
23							Buen MDA proceso
24							
25			Mejora continua				
25			PSP personal software process				

DATOS							
PI1						PI4	
ID	Marco de trabajo	Modelo	Procedimiento	Herramienta	Otro	Modelo de FS	Comentario
26			Mejora continua				
26			CMMI				CMMI2
27		UML	Generador de código	UML		SF genérica	
28	Ágil		Scrum		DevOps	SF en DTI	Equipo de Fabrica
28			Kanban		Equipo de SF		
29							Critica al fabrica de software
30		Modelo Medir productividad					Productividad en SF
30		Modelo Estimación tiempos					
31							
32							
33							
34							
35							
36	CMMI		Iterativo de refinamiento de requerimientos		Ingeniería de requisitos	SF data warehouse	CMMI3
36	ISO		PSP	Técnicas Heurísticas de recolección de datos para requisitos			Buen modelo General de SF
36	SOA						

DATOS							
PI1			PI4				
ID	Marco de trabajo	Modelo	Procedimiento	Herramienta	Otro	Modelo de FS	Comentario
37						SF académica	
38	Open-UP					SF Académica	
39	CMMI				MPS Melhoria do Processo do Software Brasileiro (MPS)	SF Brasil	Mejora de procesos
39							Conclusión No se aplican los conceptos de SF en Brasil
40		KM gestión de conocimiento					Malo
41		KM gestión de conocimiento					
41		Modelo de Wigg, Modelo de Nonaka Sveiby, Modelo de Earl, Kerschberg, Bustelo y Amarilla, etc.etc					
41		KM ad-hoc					
42	CMMI						Manejo de riesgos

DATOS							
PI1				PI4			
ID	Marco de trabajo	Modelo	Procedimiento	Herramienta	Otro	Modelo de FS	Comentario
42	CMMI-riegos	Manejo de Riego Ad-hoc (CMMI-Redes Petri)		Redes De Petri			
43			SPL				Prueba de que SF mejora la calidad
43		Calidad y Productividad	GQM goal question metric				SF vs Tradicional
44	BSC			KPIs para SF			Buen modelo de FS Básico
45				Generador de código		SF genérica	Buen modelo de FS Básico
45				XML-XSLT			Proceso de Generación de código
46	Marcos de Trabajo de Arquitectura (Factory scheme)	MDD	SPL	Generador de código	ADD architect driven developent	SF genérica	Buen proceso hacia SF
47						SF multimedia	
48			Scrum			SF académica	
49	PM	3 Niveles de recursos			Repositorio de Componentes	SF genérica	
49		KM gestión del conocimiento					

DATOS							
PI1				PI4			
ID	Marco de trabajo	Modelo	Procedimiento	Herramienta	Otro	Modelo de FS	Comentario
50	DSL	MDD		Generador de código			SmartTools, software como SF
50	MDA						
51	DSL visual			VSTS Visual Studio 2005 Team System ()		SF para juegos	
51				Graph Object Property Relationship and Role			
51				Graph Object Property Relationship and Role			
51				Generador de código			
52					DevOps		
53							
54	DSL	MDD	SPL			SF genérica	Muy buen modelo de SF
55		MDD	SPL		ISO15504		Quality in Development Process for Software Factories According to ISO 15504

ID	DATOS						Comentario
	PI1			PI4			
	Marco de trabajo	Modelo	Procedimiento	Herramienta	Otro	Modelo de FS	
55					Componentes de software		SF completa
55							Muy buen modelo SF
56						SF académica	
57	CMMI					CMMI3	Modelo de SF básico
57	SPICE						
58	DSL						Modelo de SF básico
58	MDA						
59	CMMI						Modelo de SF básico
59	ISO						

Fuente: Elaboración propia

8.5 Total de los elementos de fábrica de software encontrados

Este es el total de los elementos encontrados en la revisión de las 47 publicaciones revisadas:

Tabla 22: Elementos de fábrica de software encontrados

PI3	PI2		
Categoría	Tipo Componente	Componente / Elemento	Cantidad de Artículos
Desarrollo	Procedimiento	Generación de Código	12
Productividad	Modelo	SPL	10
Diseño	Modelo	DSL Domain Specific Language	7
Diseño	Modelo	MDD	7
Diseño	Marco de trabajo	MDA Model Driven Architecture	6
Diseño	Modelo	KM Gestión Conocimiento	4
Mejora continua	Modelo	CMMI	4
Planificación	Herramienta	BSC	4
Productividad	Procedimiento	PSP/TSP	4
Calidad	Marco de trabajo	ISO15504 / SPICE Software Process Improvement Capability Determination	3
Desarrollo	Procedimiento	Ágil	3
Desarrollo	Procedimiento	Scrum	3
Mejora continua	Modelo	CMMI3	3
Mejora continua	Procedimiento	Mejora continua	3
Desarrollo	Procedimiento	Kanban	2

PI3	PI2		
Categoría	Tipo Componente	Componente / Elemento	Cantidad de Artículos
Desarrollo	Herramienta	SCR software component repository	2
Diseño	Marco de trabajo	CBSE Component-based software engineering	2
Diseño	Herramienta	DSL .Net DSL tools	2
Diseño	Modelo	UML	2
Mejora continua	Modelo	CMMI2	2
Planificación	Procedimiento	AHP (Analytic Hierarchy Process)	2
Operación y Mantenimiento	Entorno	DevOps	2
Planificación	Herramienta	RBT Resource Based Theory	2
Planificación	Marco de trabajo	WST Work System Theory	2
Productividad	Modelo	Medir productividad	2
Calidad	Marco de trabajo	ISO90001	1
Calidad	Marco de trabajo	ISO9001	1
Calidad	Marco de trabajo	QMS Quality Management System	1
Calidad	Herramienta	GQM Goal Question Metric	1
Desarrollo	Herramienta	Editor Grafico	1
Desarrollo	Herramienta	Generación de Código FreeMarker	1
Desarrollo	Herramienta	Modelo-a-Modelo (AGG1 graph grammars engine)	1

PI3	PI2		
Categoría	Tipo Componente	Componente / Elemento	Cantidad de Artículos
Desarrollo	Procedimiento	RUP	1
Desarrollo	Herramienta	SpecL (model-to-text transformation language)	1
Diseño	Herramienta	DSL GME	1
Diseño	Herramienta	KM Modelo Kerschberg, Bustelo y Amarilla	1
Diseño	Herramienta	KM Modelo de Earl	1
Diseño	Herramienta	KM Modelo Monaka y Takeuchi	1
Diseño	Herramienta	KM Modelo Wigg	1
Diseño	Herramienta	MDA ArcStyle	1
Diseño	Herramienta	Microsoft DSL tools	1
Diseño	Modelo	TDD Test Driven Development	1
Diseño	Modelo	TDM Test Driven modelling	1
Diseño	Herramienta	XML	1
Diseño	Herramienta	XSLT eXtensible Stylesheet Language transformation	1
Diseño	Modelo	ADD architect driven developent	1
Diseño	Herramienta	DSL visual VSTS Visual Studio 2005 Team System	1
Diseño	Herramienta	DSL visual Graph Object Property Relationship and Role	1
Mejora continua	Herramienta	Causa Raíz	1

PI3	PI2		
Categoría	Tipo Componente	Componente / Elemento	Cantidad de Artículos
Mejora continua	Marco de trabajo	MPS Mejora de procesos de software de Brasil	1
Operación y mantenimiento	Herramienta	SLA	1
Planificación	Marco de trabajo	Zachman	1
Planificación	Herramienta	CMMI-Riesgos	1
Planificación	Modelo	Estimación de tiempo	1
Planificación	Procedimiento	MCDM multicriteria decision-making	1
Planificación	Marco de trabajo	PMBok	1
Planificación	Herramienta	Redes de Petri para Riegos	1
Planificación	Marco de trabajo	SOA Service Oriented Architecture	1
Productividad	Herramienta	FPA FuctionPoint Analisis	1
Productividad	Marco de trabajo	Marco de Trabajo ad-hoc	1
Requerimientos	Marco de trabajo	Ingeniería de requisitos	1
Requerimientos	Procedimiento	Refinamiento de requerimientos iterativo	1
Requerimientos	Herramienta	Técnica Heurística de recolección de datos	1
Seguridad	Marco de trabajo	ISO27001	1
Servicios	Marco de trabajo	ISO20000	1

Fuente: Elaboración propia

8.6 Detalle de distribución de modelos de fábrica de software

Tabla 23: Modelos de fábrica de software encontrados

ID	Modelo de FS
1	ISO9001, ISO20000, ISO270001
1	CMMI
2	SF específica
4	SF Ad-hoc :ISO9000 CMM TSP/PSP SPL= SFM-CSO[ISO/CMM]
5	SF MDSF: Model Driven Software Factory
6	SF MDSF: Model Driven Software Factory
7	SF académica
8	SF genérica
10	SF MDSF: Model Driven Software Factory
13	SF nube
15	SF académica
16	SF genérica
19	SF nube
20	SF específica
21	SF para juegos
22	MDA
22	SF genérica
23	MDA
27	SF genérica
28	SF específica
36	SF data warehouse
37	SF académica
38	SF Académica
39	SF Brasil
45	SF genérica
46	SF genérica
47	SF multimedia
48	SF académica
49	SF genérica
51	SF para juegos
54	SF genérica
56	SF académica
57	CMMI3

Fuente: Elaboración propia

8.7 Modelo de Componentes de Fábrica de Software

La siguiente ilustración muestra la relación entre categorías y todos los elementos de fábrica de software encontrados.

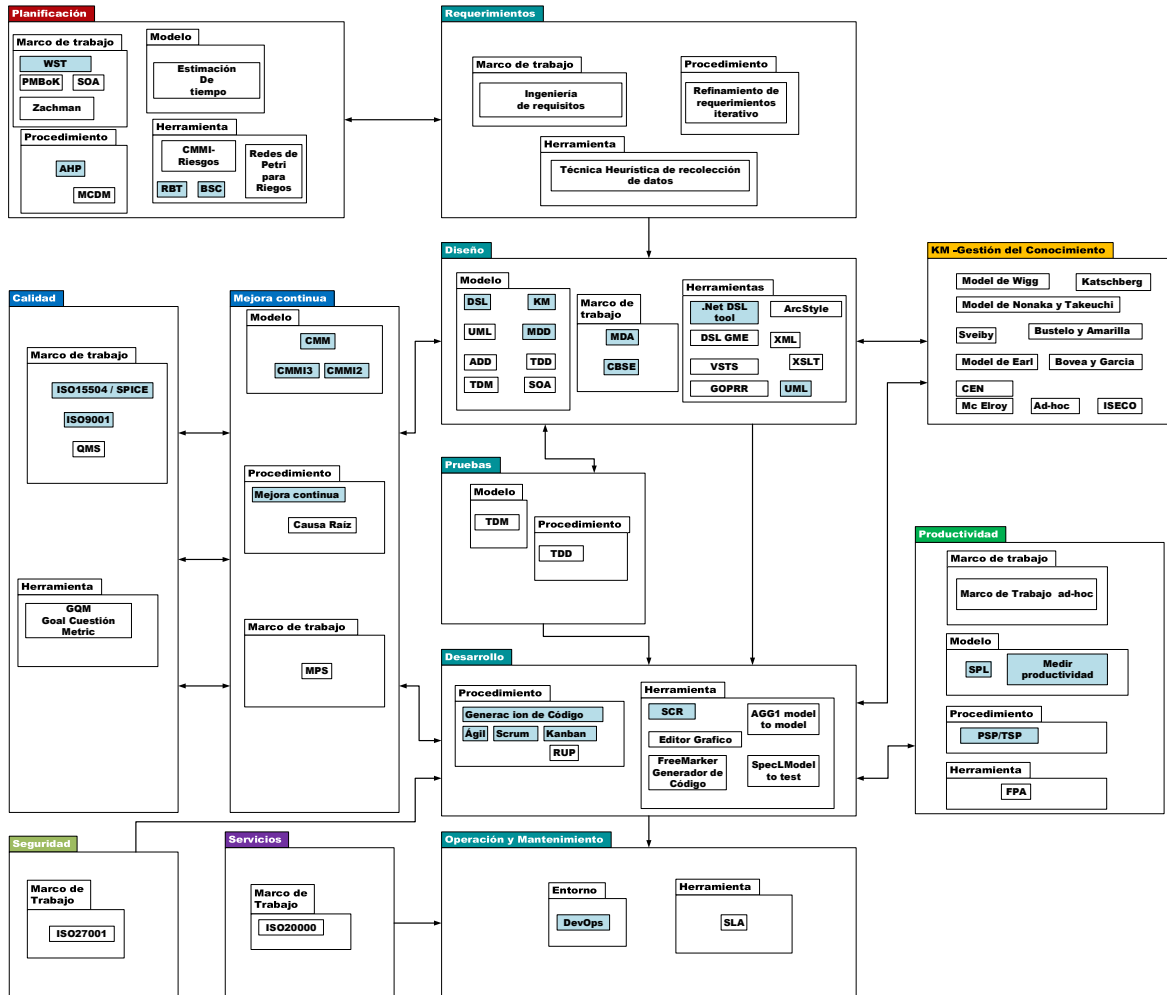


Ilustración 18: Modelo de elementos de Fábrica de Software

Fuente: Elaboración propia

8.8 Bibliometría

Tabla 24: Cantidad de Publicaciones Excluidas

Tipo de Publicación	Cuenta de Publicaciones	Cuenta de Excluidos
PPT	1	
Libro	1	1
Artículo de empresa	2	
Tesis	3	1
Trabajo Final	4	4
Artículo de conferencia	16	5
Artículo	32	3
Total	59	14

Fuente: Elaboración propia

Tabla 25: Cantidad de Publicaciones por Base de Datos

Base de Datos	Cuenta de Publicaciones
LACCEI	1
LaernTechLib	1
Åbo Akademi University	1
Microsoft	1
Biblioteca Alberto Quijano Guerrero -Universidad de Nariño	1
Portal de Periódicos Unisul	1
CORE	1
Programa de Pós-Graduação em Informática Pontifícia Universidade Católica do Paraná	1
Google books	1
Red Internacional de Investigadores en Competitividad UUM Electronic Theses and Dissertation	1
RedClara	1
AOPEN	1
Repositorio académico UPC	1
ftp://ftp.inf.puc-rio.br/pub/docs/FomularioSolicitacoes/LeandraMara-06-08-23494.pdf	1
Repositorio de la Universidad Internacional SEK Ecuador	1
Universidade Federal de Minas Gerais	1
REUTLINGEN UNIVERSITY ACADEMIC BIBLIOGRAPHY	1
Guevara cia.	1
Tecnociencia chihuahua	1
CiteSeerX	1
Universidad Nacional de Ingeniería - Biblioteca Central	1
HAL	2
Ecorfan	2
AMC	2
Cornel University	2
DOAJ	3
MDPI	3
EBSCO	3
IEEEExplore	3
Academia.edu	4
ResearchGate	13
Total	59

Fuente: Elaboración propia

Tabla 26: Cantidad de Publicaciones por Idioma

Idioma	Cuenta de Publicaciones
Español	13
Inglés	38
Portugués	8
Total	59

Fuente: Elaboración propia

Tabla 27: Cantidad de Publicaciones por Año (1998-2018)

Tipo de Publicación	Cuenta de Publicaciones
2001	3
2002	1
2003	1
2004	1
2005	4
2006	5
2007	4
2008	2
2009	5
2010	1
2011	6
2012	2
2013	4
2014	2
2015	5
2016	2
2017	6
2018	4
??	1
Total	59

Fuente: Elaboración propia

Terminus Est