

TESIS DE MAESTRÍA
MESTRÍA EN INGENIERÍA DE SISTEMAS
DE INFORMACIÓN

Título:

“PROPUESTA DE UN MODELO DE GESTIÓN
DEL CONOCIMIENTO PARA FÁBRICAS DE
SOFTWARE”

Autor: Esp. Adriana Maulini Buño

Director de Tesis: Dra. Ma. Florencia Pollo-Cattaneno.
Mg. Luciano Straccia

Buenos Aires - 2021



UTN.BA



**UTN.BA
ESCUELA DE
POSGRADO**

**TESIS de Maestría en
Ingeniería en Sistemas de Información**

**“PROPUESTA DE UN MODELO DE GESTIÓN DEL
CONOCIMIENTO PARA FÁBRICAS DE SOFTWARE”**

Alumno: Esp. Adriana Maulini Buño

Director: Dra. Ma. Florencia Pollo-Cattaneo

Co-Director: Mg. Luciano Straccia

CIUDAD AUTÓNOMA DE BUENOS AIRES, ABRIL 2020

RESUMEN

Desde hace algunos años, el conocimiento se ha convertido en uno de los recursos más importantes de una organización, por lo que se necesita administrarlo de forma adecuada para poder obtener las ventajas competitivas que derivan del mismo y ayudar a alcanzar los objetivos organizacionales.

Para administrar el conocimiento de forma adecuada, la gestión del conocimiento es de vital importancia. Existen rubros en los cuales gestionar el conocimiento implica determinados desafíos en comparación con otros rubros. Uno de estos es la Ing. del Software, en especial, las Fábricas de Software.

El presente trabajo muestra conceptos relacionados con el conocimiento y gestión de este, así como también de fábricas de software. Identifica los principales desafíos que presenta la gestión del conocimiento en el rubro mencionado y analiza los principales modelos de gestión del conocimiento existentes en función de estos desafíos para determinar su aplicabilidad en fábricas de software, posteriormente, propone un modelo de gestión del conocimiento aplicable al tipo de organizaciones mencionadas en función de los desafíos identificados.

ABSTRACT

From years ago, knowledge has become one of the most important resources of an organization. It needs to be properly managed in order to obtain the competitive advantages derived from it and help the organization to successfully achieve its objectives.

To properly manage knowledge, the “knowledge management” process is very important. There are organization types in which managing knowledge involves certain challenges compared to others. One of these types are software engineering organizations, especially software factories.

In this document, definitions and concepts related with knowledge, knowledge management and software factories are presented. The main challenges of managing the knowledge in software factories are identified and most important knowledge frameworks are compared with these challenges to determine if they can be properly applied in a software factory. A knowledge framework is proposed considering the mentioned challenges to be applied in a software factory.

AGRADECIMIENTOS

Ha sido un largo y duro trayecto que no hubiese podido recorrer sin la ayuda de muchas personas que han estado involucradas desde el inicio en este proyecto.

Gracias a familiares y amigos por su apoyo incondicional.

Muchas gracias a mis directores Florencia Pollo y Luciano Straccia por toda su guía e infinita paciencia para llevar todo este esfuerzo a buen puerto.

Gracias a todos los profesores de la especialización y maestría en Ing. de Sistemas de la Universidad Tecnológica Nacional, en especial a Inés Casanovas y Ramon García Martínez (QEPD) por su guía para encontrarle una forma inicial a todo esto.

También estoy muy agradecida con los expertos que participaron en el proceso de validación de esta tesis: Andrei Rukavina, Sebastián Lopreto, Rafael Natera, Ramiro Garbarini, Marcelo Masci, Pablo Cigliuti y Martin Riera, el aporte de su experiencia ha sido muy valioso. Gracias también a los demás expertos que pusieron la mejor onda posible en el proceso de validación.

Gracias a la Universidad Tecnológica Nacional por brindar el ambiente y recursos necesarios para que esto sea posible.

A todos ustedes, ¡Muchas Gracias!

ÍNDICE GENERAL

CONTENIDO

RESUMEN	i
ABSTRACT	i
AGRADECIMIENTOS.....	ii
ÍNDICE GENERAL.....	iii
ÍNDICE DE TABLAS	vi
ÍNDICE DE FIGURAS	viii
1. INTRODUCCIÓN.....	1
1.1. Contexto de la tesis.....	1
1.2. Objetivos del Trabajo de Maestría	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
1.2.3. Alcance	3
1.3. Metodología Empleada	3
1.4. Métodos	4
1.4.2. Juicio de expertos	5
1.4.3. Entrevistas a expertos	6
1.5. Producción Científica Derivada de Resultados Parciales de la Tesis.....	6
1.6. Estructura del Trabajo de Maestría.....	7
2. ESTADO DEL ARTE	8
2.1. Conocimiento	8
2.2. Importancia del conocimiento en las organizaciones.....	10
2.3. Gestión del Conocimiento.....	13
2.4. Gestión del Conocimiento en la Ingeniería del Software.....	21
2.4.1. Ingeniería del Software	21
2.4.2. El conocimiento en la Ingeniería del Software y los desafíos de la GC	21
2.5. Fábricas de Software	24
3. DESCRIPCIÓN DEL PROBLEMA.....	28
3.1. Identificación del problema de investigación	28
3.1.1. Modelos actuales de gestión del conocimiento y su relación con los desafíos de la GC en SF	28

3.1.4. Resultados del análisis	67
4. SOLUCIÓN.....	71
4.1. Modelo propuesto	71
4.1.1. Generalidades	71
4.1.2. Estructura del modelo	72
5. Validación	112
5.1 Definir el objetivo del juicio de expertos.	114
5.2 Selección de los jueces	114
5.2.1. Expertos seleccionados	117
5.3. Explicar las dimensiones e indicadores a medir (definición de los ítems de evaluación) 124	
5.4. Especificar el objetivo de la prueba	126
5.5. Establecer los pesos diferenciales de las dimensiones de la prueba	127
5.6. Diseño de plantilla.....	127
5.7. Conclusiones del juicio de expertos	127
6. conclusiones	139
6.1. Aportaciones de la Tesis.....	139
6.2. Futuras Líneas de Investigación	144
7. Referencias bibliográficas	147
ANEXO A	154
ANEXO B	155
ANEXO C	156
ANEXO D	157
ANEXO E	158
ANEXO F	159
Anexo G	160
Anexo H	161
Anexo I.....	165
C.V. Andrei Rukavina	165
Anexo J	172
C.V. SebastianLopreto	172
Anexo K.....	175

C.V. Ramiro Garbarini.....	175
Anexo L.....	191
C.V. Martín Riera	191
Anexo M	196
C.V. Marcelo Masci.....	196
Anexo N	199
C.V. Pablo Cigliuti	199
Anexo O	207
C.V. Rafael Natera	207

ÍNDICE DE TABLAS

1	Clasificación de las definiciones del conocimiento	9
2	Clasificaciones de las definiciones de GC según Rueda Martínez (2014)	14
3	Clasificación de las definiciones de GC.....	16
4	Unión y clasificación de definiciones de GC bajo los criterios de clasificación de Rueda Martínez y Pérez & Urbáez.....	16
5	Clasificación de definiciones de SF.....	26
6	Etapas, sub-etapas y actividades del modelo de GC de Wiig.....	31
7	Desafíos de la GC en el desarrollo de software en relación con el modelo de GC de Wiig	35
8	Desafíos de la GC en el desarrollo de software con relación al modelo de GC de Nonaka y Takeuchi.....	38
9	Desafíos de la GC en el desarrollo de software en relación con el modelo de GC Sveiby	42
10	Escuelas de GC del modelo de Earl.....	43
11	Desafíos de la GC en el desarrollo de software en relación con el modelo de GC Earl.....	46
12	Desafíos de la GC en el desarrollo de software en relación con el modelo de GC de Kerschberg.....	49
13	Desafíos de la GC en el desarrollo de software en relación con el modelo de GC de Bustelo y Amarilla.....	52
14	. Desafíos de la GC en el desarrollo de software en relación con el modelo de GC de Mc. Elroy.....	55
15	Desafíos de la GC en el desarrollo de software en relación con el modelo CEN.....	58
16	Desafíos de la GC en el desarrollo de software en relación con el modelo de Bovea y García.....	63
17	Desafíos de la GC en el desarrollo de software en relación con el modelo ISECO....	66
18	Matriz de vacancia de los modelos analizados en relación con los desafíos de la GC en la ingeniería de software y las SF.	67

19	Calificación de los modelos analizados en relación con la matriz de vacancia.....	69
20	Análisis del modelo propuesto en relación con los desafíos de GC definidos.....	108
21	Fuentes de argumentación seleccionadas.....	116
22	Cálculo de Ka. AndreiRukavina.....	117
23	Cálculo de Ka. Sebastián Lopreto.....	118
24	Cálculo de Ka. Ramiro Garbarini.....	119
25	Cálculo de Ka. Martín Riera.....	120
26	Cálculo de Ka. Marcelo Masci.	121
27	Cálculo de Ka. Pablo Cigliuti.....	122
28	. Cálculo de Ka. Rafael Natera.....	123
29	Resultados de la selección de expertos, expertos seleccionados.....	123
30	Dimensiones e indicadores por medir.....	124
31	Resultados de las entrevistas con los expertos para cada indicador.....	128
32	Posibilidades de mejoras sugeridas por los expertos.....	131
33	Desafíos de la GC en las Fábricas de Software.....	140

ÍNDICE DE FIGURAS

1	Modelo de Nonaka y Takeuchi	37
2	Modelo de Sveiby.	39
3	Modelo de Kerschberg.....	47
4	Modelo de Bustelo y Amarilla.....	51
5	Modelo de Mc Elroy.....	54
6	Modelo CEN.....	58
7	Modelo de Pons y otros.....	64
8	Fases del modelo propuesto.....	73
9	Actividades de la fase de preparación.....	76
10	Elementos que conforman la base de conocimientos.....	78
11	Actividades de la fase de mantenimiento del modelo.....	82
12	Actividades de la fase 1. Detectar necesidad de conocimiento.....	85
13	Fase 2. Buscar el conocimiento.....	89
14	Actividades de la fase 3. Interiorizar y organizar el conocimiento.....	91
15	Actividades de la fase 4. Utilizar el conocimiento.....	93
16	Actividades de la fase 5. Compartir / Validar el conocimiento.....	97
17	Actividades de la fase 6 – Formalizar.....	100

1. INTRODUCCIÓN

En el presente Capítulo se describe el contexto de esta Tesis de Maestría (sección 1.1), los objetivos generales y específicos establecidos, y su cómo su alcance (sección 1.2). Se describe también la metodología de desarrollo, materiales y métodos empleados (sección 1.3) y la producción científica derivada de resultados parciales de la Tesis (sección 1.4) El capítulo finaliza con la descripción de la estructura general del presente trabajo de investigación (sección 1.5).

1.1. Contexto de la tesis

La creación efectiva, uso y difusión del conocimiento es la clave del éxito de las organizaciones y del desarrollo económico y social sustentable. El conocimiento es altamente valioso, único para la organización, difícil de copiar y difícil de sustituir, ya que la misma información puede generar conocimiento diferente en personas y organizaciones diferentes (Pedraja-Rejas, 2006).

Sin embargo, debe ser administrado de forma adecuada para poder obtener las ventajas competitivas que derivan del mismo, y es precisamente por esto que la gestión del conocimiento se ha vuelto tan importante. El objetivo principal de la gestión del conocimiento es proporcionar herramientas a las organizaciones para identificar, crear, mantener y medir el conocimiento que ellas generan, maximizando los beneficios individuales y globales (Perez y Urbáez, 2016).

Un rubro en donde la gestión del conocimiento presenta desafíos importantes es el área de la Ingeniería de Software. En esta disciplina de la ingeniería, el conocimiento es dinámico y evoluciona constantemente con las tecnologías, la rotación del personal es elevada, los profesionales son altamente especializados y hay altas posibilidades de cancelación de proyectos (Aurum et al, 2013). Estos y otros factores, como la posible localización horaria y geográfica diferente de personas de un mismo proyecto (Dingsoyr y Smite, 2014) y la

existencia de diferentes roles especializados, pueden derivar en dificultades específicas del rubro para generar, compartir y almacenar conocimiento.

Uno de los mejores ejemplos de aplicación de procesos de ingeniería de software, y que ha tenido mucho auge últimamente, debido a la necesidad cada vez más fuerte de producir software en cortos períodos de tiempo, son las llamadas fábricas de software (FS). Dado que las fábricas de software aplican procesos de ingeniería de software para desarrollar sus productos, no están exentas de los desafíos de gestión del conocimiento que se mencionaron anteriormente e incluso pueden sumarse problemas relacionados con los tiempos reducidos y la reutilización de código.

Es por tanto que en este trabajo de investigación se enumeran los principales desafíos que presenta la gestión del conocimiento en el rubro específico de las fábricas de software, se analizaron los principales modelos de gestión del conocimiento existentes para determinar si dichos desafíos son o no tomados en cuenta y en base a esto se propuso un modelo de gestión del conocimiento adaptado al mencionado rubro.

1.2. Objetivos del Trabajo de Maestría

Se dividen los objetivos de esta tesis en un objetivo general a alcanzar (sección 1.3.1), un conjunto de objetivos específicos que definen los pasos a seguir para alcanzar el objetivo general (sección 1.3.2), y el alcance previsto de dichos objetivos (1.3.3).

1.2.1. Objetivo General

El presente trabajo de investigación propone la creación de un modelo de gestión de conocimiento para ser aplicado en el ramo de las fábricas de software. Se procura de esta manera aportar a las fábricas de software una forma de gestionar su conocimiento de manera adecuada y teniendo en cuenta los desafíos que implica la gestión del conocimiento en este rubro.

1.2.2. Objetivos Específicos

A partir del objetivo general planteado, se derivan los siguientes objetivos específicos.

- Identificar los principales desafíos que presenta la gestión de conocimiento en las fábricas de software.
- Identificar, describir y comparar los principales modelos existentes de gestión del conocimiento en relación con los desafíos que presenta dicha disciplina en las fábricas de software.
- Desarrollar un nuevo modelo de gestión del conocimiento que pueda ser aplicado en una fábrica de software.
- Validar el modelo propuesto mediante el uso de la técnica de juicio de expertos.
- Exponer las conclusiones sobre el tema y posibles aportes futuros.

1.2.3. Alcance

En el marco del presente trabajo, se define a continuación el alcance del mismo.

- La propuesta debe estar conformada por fases y actividades que permitan solucionar los desafíos de la gestión del conocimiento en las fábricas de software.
- La propuesta debe ser flexible para que cada fábrica de software pueda aplicarla de la forma más conveniente y adaptarla a la organización.
- La propuesta debe ser compatible con las metodologías de desarrollo de software, tanto tradicionales como ágiles.
- La propuesta debe ser validada utilizando la técnica de juicio de expertos, no entra dentro de este alcance, la validación por estudio de casos principalmente por limitantes de tiempo.

1.3. Metodología Empleada

Para construir el conocimiento de la presente investigación, se sigue una filosofía de investigación constructivista. Para Gergen (2007) este paradigma permite abordar acontecimientos de alta complejidad en donde el saber no se considera como absoluto y acumulado, ya que en lo social los fenómenos evolucionan constantemente. Gergen et al. (2007) indica que mediante el constructivismo social se intenta delinear un cuerpo de trabajo en el que los procesos cognitivos y el entorno social son importantes. Para Creswell (2013), el

constructivismo en vez de comenzar con una teoría (como lo hace el positivismo) genera una teoría o patrón de conocimiento durante la investigación. Durante la realización del presente trabajo, no solo se estudia un tema de índole social como es la gestión del conocimiento en fábricas de software, sino que, además, los desafíos que presenta la GC del conocimiento en el área, así como su posible solución serán desarrollados durante el mismo, no siendo un resultado exacto o limitante.

El abordaje de la investigación es de tipo cualitativo, ya que se busca examinar la naturaleza de los fenómenos en su ambiente usual (Cuenya y Ruetti, citados en Ramos, 2015).

Para alcanzar los objetivos previstos se siguen los siguientes pasos: (i) realizar una investigación documental exploratoria sobre conocimiento, gestión de conocimiento, fábricas de software y desafíos de la GC en FS (ii) identificar los desafíos de la GC en FS mediante investigación documental, (iii) analizar los principales modelos de GC existentes comparándolos con los desafíos de GC para FS, determinando de esta manera qué modelos podrían ser aplicables en una FS, (iv) validar los desafíos determinados con entrevistas a expertos en el área, (v) elaborar una propuesta de modelo de GC que pueda ser aplicable en una FS, (vi) validar la propuesta utilizando la técnica de Juicio de Expertos.

El enfoque de esta investigación es de tipo cualitativo. Según Cuenya y Ruetti citados en Ramos (2015), este enfoque busca la comprensión de los fenómenos en su ambiente usual, desarrollando la información basada en la descripción de situaciones, lugares, periódicos, textos, e individuos. Este enfoque, suele ser utilizado para el descubrimiento y relevamiento de preguntas de investigación. En esta investigación se buscan describir los desafíos de la GC en las FS, utilizando técnicas cualitativas como juicio de expertos, entrevistas y revisión documental.

El alcance de la investigación es de tipo descriptivo (Hernández et al, 2010), ya que se describen los desafíos de la GC en las FS y su posible solución.

1.4. Métodos

A continuación, se describen los métodos de investigación que se utilizan a lo largo de este proyecto.

1.4.1. Revisiones sistemáticas

Las revisiones sistemáticas (Argimón, 2004) de artículos científicos siguen un método explícito para resumir la información sobre determinado tema o problema. Durante esta investigación se utilizará esta técnica para llevar a cabo la revisión documental.

Para la revisión de documentos y libros se utilizaron repositorios conocidos tales como el Sistema de Bibliotecas y de Información de la Universidad de Buenos Aires (<http://www.sisbi.uba.ar>), el Sistema Nacional de Repositorios Digitales (SNRD) del Ministerio de Ciencia, Tecnología e Innovación Productiva (<http://repositorios.mincyt.gob.ar>), la biblioteca de la Universidad Tecnológica Nacional, entre otros.

1.4.2. Juicio de expertos

Para la validación del modelo se utiliza una técnica denominada juicio de expertos que “consiste (...) en solicitar a una serie de personas la demanda de un juicio hacia un objeto, un instrumento, un material de enseñanza, o su opinión respecto a un aspecto concreto.” (Cabero & Llorente, 2013). El juicio de expertos se define también como una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos cualificados en éste, y que pueden dar información, evidencia, juicios y valoraciones (Escobar y Cuervo, 2008).

En el caso de esta investigación, debido a limitaciones de tiempo y de instrumentos exactos para medir el éxito de un modelo de gestión de conocimiento en fábricas de software (las métricas comunes de un proyecto de software pueden depender de muchos factores, por lo que utilizar éstas no es acertado para medir el modelo en particular) se utiliza el juicio de expertos como instrumento principal de validación.

Los jueces son seleccionados mediante el coeficiente de juicio de expertos (Cabero y Llorente, 2013) y se siguen los siguientes pasos para el proceso de validación (Escobar y Cuervo, 2008):

1. Definir el objetivo del juicio de expertos.
2. Seleccionar los jueces.

3. Explicitar tanto las dimensiones como los indicadores que está midiendo cada uno de los ítems de la prueba.
4. Especificar el objetivo de la prueba.
5. Establecer los pesos diferenciales de las dimensiones de la prueba.
6. Diseñar las planillas.
7. Calcular la concordancia entre jueces
8. Elaborar las conclusiones del juicio

1.4.3. Entrevistas a expertos

Se realizan entrevistas estructuradas a los expertos seleccionados para recolectar su opinión sobre el modelo propuesto y los desafíos de la GC en FS determinados previamente mediante revisión documental.

1.5. Producción Científica Derivada de Resultados Parciales de la Tesis

Durante el desarrollo de esta investigación se han comunicado los resultados parciales de la misma a un congreso nacional y un congreso internacional (en dos ocasiones).

Straccia, L; Maulini, A; Pytel, P; Masci, M; Vegega, C; Pollo-Cattaneo, Ma. F (2017). La Gestión del Conocimiento en Pequeñas y Medianas Fábricas de Software en el Área Metropolitana de Buenos Aires. Proceedings XIX Workshop de Investigadores de Ciencias de la Computación, WICC 2017. Instituto Tecnológico de Buenos Aires ITBA. Pág.575-579. ISBN: 978-987-42-5143-5.

Maulini, A.; Straccia, L; Pollo-Cattaneo, M.F. (2018) Una aproximación a un modelo de gestión de conocimiento aplicable a las pequeñas y medianas fábricas de software. InNGENIO. ISBN 978-958- 59127-9-3. Pág 53- 63.

Maulini, A.; Straccia, L; Pollo-Cattaneo, M.F. (2019) Un modelo de gestión de conocimiento aplicable a las pequeñas y medianas fábricas de software. En “Desarrollo e Innovación en Ingeniería - Cuarta Edición” (Ed. Prof. Edgar Serna M). Capítulo 1, Pág. 5-16. Editorial Instituto Antioqueño de Investigación DOI: <http://doi.org/10.5281/zenodo.3387679>.

1.6. Estructura del Trabajo de Maestría

En el capítulo introductorio de este proyecto de investigación se presenta el contexto de la tesis (1.1) seguido de los objetivos de esta (1.2), la metodología empleada (1.3), los métodos empleados (1.4) y la producción científica derivada de la misma.

En el segundo capítulo (Estado del Arte) se definen y explican conceptos relevantes para la investigación como lo son: conocimiento (2.1), importancia del conocimiento en las organizaciones (2.2), gestión del conocimiento (2.3), gestión del conocimiento en la ingeniería de software (2.4) y fábricas de software (2.5).

En el tercer capítulo, se describe el problema a abordar junto con una descripción y análisis de los principales modelos actuales de gestión del conocimiento en relación con los desafíos que presenta esta área en las fábricas de software (3.1.1) y se presentan los resultados del análisis (3.1.2).

En el cuarto capítulo se presenta la solución al problema, se propone un modelo de gestión de conocimiento adecuado para fábricas de software, detallando sus generalidades (4.1.1) y su estructura (4.1.2).

En el quinto capítulo se detalla la validación del modelo mediante el método llamado Juicio de Expertos, se detalla el objetivo de la validación (5.1), el proceso de selección de los expertos (5.2), Se detallan las dimensiones e indicadores a medir (5.3), se especifica el objetivo de la prueba (5.4), se establecen los pesos diferenciales para cada dimensión (5.5), se diseña la plantilla para las entrevistas (5.6) y se detallan las conclusiones de la validación (5.7).

En el sexto capítulo se detallan las conclusiones de la investigación, las aportaciones de la tesis (6.1) y las futuras líneas de investigación (6.2).

2. ESTADO DEL ARTE

En este capítulo se define el término “conocimiento” desde las perspectivas existentes, se clasifica, y se explica la importancia de este para una organización. Además, se describe el término “Gestión del Conocimiento”, su importancia a nivel organizacional y los distintos enfoques que existen para el mismo.

Posteriormente, se describe la GC desde el punto de vista de la ingeniería de software y los desafíos que se presentan para gestionar el conocimiento en este rubro. Así mismo, se describe el significado y diversas definiciones del término fábrica de software o *software factory* (SF) y los desafíos de GC que se presentan en este tipo de organización.

2.1. Conocimiento

El término “conocimiento” es utilizado con frecuencia, y en muchas ocasiones no se conoce su significado preciso, ni mucho menos el hecho que existen diferentes perspectivas para definirlo.

Para Díaz y Millán (2013) el conocimiento se define como: “la mezcla de creencias cognitivas y contextualizadas, perspectivas, juicios, metodologías, información, experiencias y expectativas hechas sobre un objeto, que se adaptan y potencializan por la mente de un individuo (conocedor)” (p. 88).

Pérez y Urbáez (2016), analizan diversas definiciones existentes para el término conocimiento y concluyen que existen dos campos epistemológicos sobre la naturaleza de este: “una perspectiva objetivista, donde se parte que el conocimiento puede ser un recurso físico tangible de la organización y otra perspectiva basada en que el conocimiento es esencialmente personal e inmerso en las prácticas individuales y organizacionales” (p. 206). Clasifican a su vez diversas definiciones del término señalado según puede observarse en la tabla 1.

Autor	Año	Definición	Clasificación
Nonaka y Takeuchi	1995	Creencia en una verdad justificada.	Personal
Davenport	1998	Fluida mezcla estructurada de experiencias, valores, información contextualizada y ojo clínico muy experto que proporciona un marco de trabajo excelente para evaluar e incorporar nuevas experiencias e información.	Personal - Objetivista
Davenport y Prusak	1998	Mezcla de experiencias estructuradas, valores e información no contextual que proporciona un marco para evaluar nuevas experiencias e información.	Objetivista
Wiig	1993	Yuxtaposición, integración y relación de la información aislada para desarrollar nuevos significados	Objetivista

Tabla 1: Clasificación de las definiciones del conocimiento. Fuente: Pérez y Urbáez (2016)

En relación con la definición de conocimiento de Nonaka y Takeuchi, Reyes (2005) la explica y da su opinión de la siguiente manera: “esto quiere decir que si se cree en una propuesta de conocimiento ésta solo puede ser justificada por hechos, siendo esta definición a mi juicio una de las de mayor peso” (p. 3). En relación con esto Pérez (2004) indica que: “ese elemento que le falta a la creencia para convertirse en conocimiento puede ser la verdad de lo que se cree.” (p. 19). Esto refuerza la definición de Nonaka y Takeuchi, indicando que, si se cree algo y resulta ser verdad, entonces es conocimiento.

Sobre la definición de Davenport, Ropain y otros (2014) indican que las capacidades intelectuales de los seres humanos, los valores culturales, las habilidades, la experiencia, inclusive los modelos mentales, pueden convertirse en una potente herramienta que permita crear valor a la empresa. Por lo cual reafirma la definición de Davenport (1998), ya que las experiencias y los valores de los individuos se incluyen en la misma.

Briceño y Bernal (2010), aseguran que Wiig identifica el conocimiento como pragmático, es decir; útil y enfocado en la resolución de problemas (guía de la conducta humana). Esto sigue justificando la clasificación de la definición de Wiig como objetivista según Pérez y Urbáez (2016) que se puede observar en la tabla anterior.

Siguiendo en la línea de definiciones del término conocimiento, Uribe y otros (2008) indican que el conocimiento no solo es resultado de actividades científicas, sino que deriva de todas las elaboraciones humanas relacionadas con el desarrollo de experiencias vivenciales que conllevan al descubrimiento del mundo para bien de la humanidad. Esta definición podría ser clasificada dentro de la perspectiva personalista de Pérez y Urbáez (2016). Sin embargo, el

resultado de estas experiencias vivenciales se puede tornar de gran valor para una organización y volverse un activo tangible. El mismo autor indica al respecto que las organizaciones no pueden generar conocimiento por sí solas, sino que necesitan de las personas para generarlo.

El conocimiento puede ser clasificado según diversos criterios. La más conocida y utilizada es la de Nonaka y Takeuchi (1995). Estos autores dividen el conocimiento en “tácito” – aquel que es difícil de expresar, formalizar y compartir, muy personal y subjetivo, derivado de la experiencia - y conocimiento “explícito” – aquel que se puede expresar y formalizar fácilmente, por lo cual, se adquiere mediante métodos formales de estudio -. Para estos autores, ambos tipos están presentes en una organización, y tienen igual de importancia para la misma.

Pérez y Urbáez (2016) también clasifican el conocimiento según su origen en: “perceptivo” (resultado de las experiencias de cada persona), “abstracto” (compuesto de reglas de comportamiento de un problema y una solución, comprende en gran parte el resultado de investigaciones científicas anteriores y se puede aprender de libros) y “experimental” (el resultado de la inducción sobre el conocimiento perceptivo).

Otra clasificación del conocimiento es la que realiza Wiig (1993). Para este autor, el conocimiento se divide en: “público” (se encuentra disponible de forma fácil para cualquier persona), “compartido” (conocimiento comunicado a través del lenguaje y representaciones) y “personal” (es el conocimiento tácito que está en una persona).

2.2. Importancia del conocimiento en las organizaciones

Para Druker (1994) el significado del conocimiento cambió radicalmente en Europa alrededor del año 1700. Este autor indica que, para Sócrates, la función del conocimiento era netamente personal: intelectual, moral y espiritual, mientras que, para Pitágoras, el conocimiento era aquello que indicaba a una persona qué decir y cómo decirlo. Ambos puntos de vista del significado del conocimiento coinciden en que no se consideraba como la habilidad de hacer algo.

Sin embargo, Druker (1994) indica que actualmente la importancia del conocimiento es otra: lo que se puede hacer con éste, puesto que aplicar el conocimiento ha desarrollado economías y generado explosiones de productividad a lo largo de los últimos 100 años.

Según Rueda Martínez (2014) el conocimiento provee las bases que permiten generar una visión crítica del entorno. Es decir, gracias al conocimiento se puede entender e interpretar el entorno.

Con base en ambos autores se puede concluir que la importancia del conocimiento radica en sentar las bases para poder interpretar nuestro entorno de forma crítica y actuar sobre el mismo.

Sin embargo, han sido muchos los intentos de entender y definir el conocimiento, de hecho, ha sido en numerosas ocasiones un tema filosófico, desde Platón hasta Averroes, desde Kant hasta Nietzsche (Ruda Martínez, 2014), por lo tanto, la importancia del conocimiento en este trabajo se centrará en el punto de vista económico y organizacional.

Son varios los autores que señalan la importancia del conocimiento a nivel económico, no sólo en las organizaciones sino en la sociedad en general. Gibbs y otros (2013) definen el conocimiento como “cualquier sentencia, procedimiento u objeto que puede ser propiedad (patente, publicación) y convertirse en un recurso económico, o una mercancía en el mercado”. Bajo esta definición puede afirmarse entonces que el conocimiento que se encuentre dentro de la organización puede traducirse en un valor económico.

A nivel organizacional, Levison y otros (2013) afirman que:

Hoy día, como una forma estratégica de obtener mejores resultados, la sociedad y las organizaciones de trabajo, se ven forzados a enfilar sus acciones hacia un nuevo orden de cosas, en el que el conocimiento juega un valor significativo como garantía de éxito organizacional y, el talento humano, es considerado como el capital más importante para la consecución de sus metas. (p.15)

Drucker (1994), indica que la base de la economía actual y de la producción ya no son los recursos naturales o la mano de obra, sino el conocimiento. Afirma además que el valor ahora se crea mediante la productividad y la innovación, ambas cosas dependientes del conocimiento. Resalta el conocimiento como el recurso más importante de una compañía y la necesidad de saber adquirirlo y administrarlo.

Drucker fue uno de los primeros autores en sacar a la luz el término “sociedad del conocimiento”, haciendo énfasis en lo explicado recientemente (la economía actual se basa en el conocimiento). Sobre las declaraciones de Drucker; Fernández y Borjas (2008), indican que la

llamada “sociedad del conocimiento” se integra por un conjunto de información, inteligencia y experiencia práctica, que, si se gestionan adecuadamente, pueden constituir la base de las cualidades diferenciadoras de las organizaciones de hoy. Indican también que la capacidad para convertir esa información en conocimiento, para compartirlo o involucrarlo en las prácticas diarias de la organización constituye la capacidad organizativa más importante para enfrentar el entorno de hoy en día.

Para Ezponda (2008), Drucker estaba pensando únicamente en “empresas del conocimiento”, puesto que el conocimiento del que habla es un medio para ganar posiciones en un mercado previamente existente en el que compiten diversas organizaciones ofertando bienes y servicios, pero, indica que estas definiciones podrían adaptarse también al conocimiento científico, donde la clave de la innovación sería la gestión de este. Para Ezponda, las definiciones de Drucker son criticables, pues solo tienen en cuenta el mercado económico, y se olvida de las “comunidades del conocimiento” (por ejemplo: artistas, científicos), en donde el conocimiento aportado a la sociedad es previo a la generación de algún valor en el mercado. Sin embargo, este trabajo busca destacar la importancia del conocimiento y su gestión a nivel organizacional, por lo cual no serán tomados en cuenta factores no relacionados directamente con las organizaciones.

En la misma línea, siguiendo con la “sociedad del conocimiento”, Pedraja-Rejas (2006) indica que, en la actualidad, las tecnologías de la información y comunicación facilitan una rápida globalización de la actividad económica, por lo tanto, la creación efectiva, uso y difusión del conocimiento es la clave del éxito de las organizaciones y del desarrollo económico y social sustentable. El mencionado autor indica también que, en efecto, el conocimiento es altamente valioso, único para la organización, difícil de copiar y difícil de sustituir, ya que la misma información puede generar conocimiento diferente en personas y organizaciones diferentes. Plantea también que:

El crear, compartir y aplicar el conocimiento, no sólo para la generación de productos sino para el diseño de la estrategia, es el proceso que da sustentabilidad a la firma. Al compartir conocimiento de manera participativa, además, se logra compromiso, confianza y cooperación al momento de la implementación de la estrategia. (p. 570).

Con esta afirmación ya no solo se habla de ventajas competitivas para la organización desde el punto de vista económico, se va más allá, se está afirmando que el compartir conocimiento incluso puede mejorar el clima laboral de una organización en pro de esta.

Con base en las definiciones anteriores, se puede entonces decir que el conocimiento es de vital importancia en una organización, pues se ha vuelto un recurso importante, capaz de generar activos y ventajas competitivas si se recolecta, retiene y administra de manera correcta. Es importante mencionar además que el conocimiento que está dentro de una organización se encuentra principalmente en la cabeza de sus miembros, de ahí la importancia y dificultad de gestionarlo adecuadamente. En los siguientes puntos se explica con más detalle qué es la gestión del conocimiento y cuál es su importancia dentro de una organización.

2.3. Gestión del Conocimiento

Según todo lo descrito anteriormente en relación con el conocimiento y su importancia, surge la necesidad de administrar el mismo dentro de las organizaciones para obtener provecho de los beneficios y ventajas competitivas que pueden generarse a partir del mismo. Es aquí donde nace el término “Gestión del Conocimiento” (GC de ahora en adelante).

Perez y Urbáez (2016) definen la GC como:

Un enfoque gerencial o disciplina emergente que busca de manera estructurada y sistemática aprovechar el conocimiento generado para alcanzar los objetivos de la organización y optimizar el proceso de toma de decisiones. La idea central de la GC es proporcionar herramientas a las organizaciones para identificar, crear, mantener y medir el conocimiento que ellas generan, maximizando los beneficios individuales y globales. No obstante, lo anterior, la GC es un concepto en construcción y con muchas interpretaciones que sigue alimentándose de los trabajos de investigación que se han hecho y que se harán en el futuro. (p. 210)

Para Cruz (2018)

La Gestión del Conocimiento tiene lugar en todas las organizaciones y está relacionada con el adecuado uso del conocimiento organizacional para elevar y fortalecer aspectos como la productividad, innovación, incremento de las competencias de los trabajadores y un mejor aprovechamiento del conocimiento existente y su aplicación en todos los procesos organizacionales. (p. 157).

Rueda Martínez (2014), indica que no existe un consenso definitivo en lo que al término GC concierne, y que la mayoría de las definiciones académicas se pueden clasificar en tres grupos:

- 1- La GC entendida como la **explotación** de un recurso a disposición de la organización y que, si bien, teóricamente, se asume como un recurso intangible, realmente se maneja como un recurso material sin conexión con su creador, creador que es eliminado del

proceso. El conocimiento existe y está a disposición de la organización, que lo puede mover, almacenar y distribuir sin que exista ningún conflicto o cortapisa.

- 2- La GC definida desde una **perspectiva humana productiva**. Humano es el originador del conocimiento y el que lo utiliza en beneficio de la organización durante su permanencia en la misma. La organización puede convertir este conocimiento en un activo accesible para el resto de sus miembros, y generar ventajas competitivas a partir del mismo.
- 3- La GC conceptualizada en base a una descripción del **proceso de producción y aplicación** del conocimiento.

En la siguiente tabla (ver tabla 2) se pueden observar algunas definiciones y su clasificación dentro de estos tres grupos, según Rueda Martínez (2014).

Autor	Año	Definición	Clasificación
O'Dell y Grayson	1998	GC es el proceso de identificación, captura y aprovechamiento del conocimiento para ayudar a la compañía a competir. El intercambio y la transferencia son evidencias tangibles de una organización que aprende.	Explotación
Owen	1999	La GC es más que la aplicación de TIC's para gestionar aplicaciones intensivas en conocimiento. La GC es una nueva manera de pensar sobre las modernas organizaciones. La GC ayuda a los directivos a poner en relación todos los aspectos de la organización con los temas relacionados con el conocimiento, facilitándoles responder cuestiones tales como cómo apoyar a los trabajadores del conocimiento o cómo transformar el conocimiento en productos y servicios.	Explotación
Greiner, Böhmman y Krcmar	2007	La GC incluye todas las actividades que utilizan conocimiento para alcanzar los objetivos de la organización con el fin de afrontar los desafíos del medio y mantener una posición competitiva en el mercado.	Explotación
González-García y Parés-Ferrer	2012	"La gestión del conocimiento identifica y explota, en el trabajo cotidiano, el conocimiento creado en la organización y el adquirido del exterior, generaliza las mejores prácticas, propicia el incremento del capital intelectual de la organización y su valor de mercado, a la vez que facilita la generación de nuevos conocimientos y su materialización en productos y servicios".	Explotación
Marshall, Prusak y Shpilberg	1996	La GC supone la tarea de reconocer un activo personal (encerrado en la mente de un ser humano) y convertirlo en un activo empresarial al que se pueda acceder y que pueda ser utilizado por un amplio número de individuos para la toma de decisiones en la organización.	Humano-Productiva
Todd	2001	La GC concierne el lado humano de la información, el saber humano que se encuentra en las mentes de las personas, el aspecto documental de la información, lo tangible, los artefactos informativos que han registrado las ideas de generaciones previas y los sistemas que permiten generar una estructura que facilite y potencie su usabilidad.	Humano-Productiva

Pascual y Sundardas	2003	La GC sería un método para cubrir una necesidad histórica de las empresas: "retener el conocimiento generado por su capital humano durante el periodo de permanencia de este en la empresa.	Humano-Productiva
Gao, Meng y Clarke	2008	La GC en las empresas supone la gestión de las actividades de los trabajadores del conocimiento., gestión que se alcanza facilitando, motivando, destacando y respaldando a estos trabajadores del conocimiento y proporcionando o alimentando un entorno de trabajo adecuado.	Humano-Productiva
Jones y Leonard	2009	Es el proceso de adquirir conocimiento de la organización o de otra fuente y convertirlo en información explícita que los empleados pueden utilizar para transformarla en conocimiento propio que les permite crear y aumentar el conocimiento de la organización.	Humano-Productiva
O'Dell y Grayson	1998	La GC es el proceso de identificar, capturar y potenciar el conocimiento para contribuir a la competencia de la empresa.	Procesos
Swan, Newell y Robertson	2000	Aquella actividad que comprende cualquier proceso y práctica relativa a la creación, adquisición, captura, intercambio y uso del conocimiento, habilidades y pericia.	Procesos
Duffy	2000	El término GC se usa comúnmente para describir el proceso de localizar, organizar, transferir y usar información.	Procesos
Mowery	2010	Gestión de los procesos que afectan a la creación, comunicación, distribución y explotación del conocimiento relativo a la estrategia corporativa, mercados, tecnologías de producto y procesos y prácticas de gestión en una empresa.	Procesos
Choo y Alvarenga Neto	2010	La GC organizacional trata, realmente, de la gestión del contexto y condiciones en las que el conocimiento puede ser creado, compartido y puesto en uso con el fin de alcanzar las metas de la organización.	Procesos

Tabla 2. Clasificaciones de las definiciones de GC según Rueda Martínez (2014). Fuente: Elaboración propia con base a la clasificación de Rueda Martínez (2014).

Siguiendo con la diversidad de definiciones existentes para la GC, Pérez y Urbáez (2016), indican que también existen dos enfoques o clasificaciones principales para las distintas definiciones, un enfoque organizacional y un enfoque económico. El enfoque organizacional establece que el único recurso competitivo en la organización es el conocimiento y considera primordial que la empresa sistematice los procesos mediante los cuales sus empleados adquieren, generan y comparten los conocimientos, estos conocimientos son necesarios para anticiparse a los retos presentes y futuros y generar una organización adaptativa. El enfoque económico define a la GC como un proceso mediante el cual las organizaciones generan riqueza a partir de sus activos intelectuales, los cuales permiten generar ventajas competitivas.

En la siguiente tabla (ver tabla 3) se presentan la clasificación en los grupos mencionados de distintas definiciones de GC según Pérez y Urbáez (2016).

Autor	Año	Definición	Clasificación
Nonaka y Takeuchi	1995	Capacidad de la empresa para crear conocimiento nuevo, diseminarlo en la organización e incorporarlo en productos, servicios y sistemas.	Organizacional-Económico
Sveiby	2001	Arte de crear valor con los activos intangibles de una organización.	Económico
Bueno	2000	Función que planifica, coordina y controla los flujos de conocimiento que se producen en la empresa en relación con sus actividades y con su entorno, con el fin de crear unas competencias esenciales.	Organizacional
Rodríguez	2006	Conjunto de procesos sistemáticos (identificación y captación del capital intelectual; tratamiento, desarrollo y compartimiento del conocimiento; y su utilización) orientados al desarrollo organizacional o personal y, consecuentemente, a la generación de una ventaja competitiva para la organización o el individuo.	Organizacional-Económico
Flores Urbáez y Peña-Cedillo	2008	Proceso organizacional dirigido a crear una cultura del compartir, generar, orientar, aplicar y evaluar el conocimiento con la finalidad de ser aplicado por los miembros de la organización para hacerla más productiva y competitiva a través de procesos, productos y servicios innovadores, permitiendo tomar decisiones exitosas en entornos dinámicos.	Organizacional-Económico
Daft	2010	Esfuerzo sistemático en encontrar, organizar y dar acceso al capital intelectual de la organización e introducir una cultura de aprendizaje continuo y de compartir conocimiento, de tal forma que las actividades de la organización puedan basarse en el conocimiento existente.	Organizacional

Tabla 3. Clasificación de las definiciones de GC. Fuente: Pérez y Urbáez (2016).

Como se puede observar, durante este apartado se han mostrado clasificaciones diferentes de la definición del término “gestión del conocimiento” de diversos autores según dos puntos de vista. Cada autor de los mencionados realiza una clasificación diferente basado en las definiciones tenidas en cuenta para su investigación, sin embargo, en la tabla 4 se agrupan todas las definiciones dadas, y se clasifica cada una de ellas en ambos puntos de vista, según el criterio que se tiene en este trabajo.

Autor	Año	Definición	Clasificación Rueda-Martínez	Clasificación Pérez y Urbáez
O'Dell y Grayson	1998	GC es el proceso de identificación, captura y aprovechamiento del conocimiento para ayudar a la compañía a competir. El intercambio y la transferencia son evidencias tangibles de una organización que aprende.	Explotación	Organizacional-Económico
Owen	1999	La GC es más que la aplicación de TIC's para gestionar aplicaciones intensivas en conocimiento. La GC es una nueva manera de pensar sobre las modernas organizaciones. La GC ayuda a los directivos a poner en relación todos los aspectos de la organización con los temas relacionados con el conocimiento,	Explotación	Organizacional

		facilitándoles responder cuestiones tales como cómo apoyar a los trabajadores del conocimiento o cómo transformar el conocimiento en productos y servicios.		
Greiner, Böhmman y Krcmar	2007	La GC incluye todas las actividades que utilizan conocimiento para alcanzar los objetivos de la organización con el fin de afrontar los desafíos del medio y mantener una posición competitiva en el mercado.	Explotación	Organizacional-Económico
González-García y Parés-Ferrer	2012	“La gestión del conocimiento identifica y explota, en el trabajo cotidiano, el conocimiento creado en la organización y el adquirido del exterior, generaliza las mejores prácticas, propicia el incremento del capital intelectual de la organización y su valor de mercado, a la vez que facilita la generación de nuevos conocimientos y su materialización en productos y servicios”.	Explotación	Económico
Marshall, Prusak y Shpilberg	1996	La GC supone la tarea de reconocer un activo personal (encerrado en la mente de un ser humano) y convertirlo en un activo empresarial al que se pueda acceder y que pueda ser utilizado por un amplio número de individuos para la toma de decisiones en la organización.	Humano-Productiva	Económico
Todd	2001	La GC concierne el lado humano de la información, el saber humano que se encuentra en las mentes de las personas, el aspecto documental de la información, lo tangible, los artefactos informativos que han registrado las ideas de generaciones previas y los sistemas que permiten generar una estructura que facilite y potencie su usabilidad.	Humano-Productiva	Organizacional
Pascual y Sundardas	2003	La GC sería un método para cubrir una necesidad histórica de las empresas: “retener el conocimiento generado por su capital humano durante el periodo de permanencia de este en la empresa.	Humano-Productiva	Organizacional
Gao, Meng y Clarke	2008	La GC en las empresas supone la gestión de las actividades de los trabajadores del conocimiento., gestión que se alcanza facilitando, motivando, destacando y respaldando a estos trabajadores del conocimiento y proporcionando o alimentando un entorno de trabajo adecuado.	Humano-Productiva	Organizacional
Jones y Leonard	2009	Es el proceso de adquirir conocimiento de la organización o de otra fuente y convertirlo en información explícita que los empleados pueden utilizar para transformarla en conocimiento propio que les permite crear y aumentar el conocimiento de la organización.	Humano-Productiva	Organizacional
O'Dell y Grayson	1998	La GC es el proceso de identificar, capturar y potenciar el conocimiento para contribuir a la competencia de la empresa.	Procesos	Organizacional-Económico
Swan, Newell y Robertson	2000	Aquella actividad que comprende cualquier proceso y práctica relativa a la creación, adquisición, captura, intercambio y uso del conocimiento, habilidades y pericia.	Procesos	Organizacional

Duffy	2000	El término GC se usa comúnmente para describir el proceso de localizar, organizar, transferir y usar información.	Procesos	Organizacional
Mowery	2010	Gestión de los procesos que afectan a la creación, comunicación, distribución y explotación del conocimiento relativo a la estrategia corporativa, mercados, tecnologías de producto y procesos y prácticas de gestión en una empresa.	Procesos	Organizacional-Económico
Choo y Alvarenga Neto	2010	La GC organizacional trata, realmente, de la gestión del contexto y condiciones en las que el conocimiento puede ser creado, compartido y puesto en uso con el fin de alcanzar las metas de la organización.	Procesos	Organizacional
Nonaka y Takeuchi	1995	Capacidad de la empresa para crear conocimiento nuevo, diseminarlo en la organización e incorporarlo en productos, servicios y sistemas.	Explotación	Organizacional-Económico
Sveiby	2000	Arte de crear valor con los activos intangibles de una organización.	Explotación	Económico
Bueno	2000	Función que planifica, coordina y controla los flujos de conocimiento que se producen en la empresa en relación con sus actividades y con su entorno, con el fin de crear unas competencias esenciales.	Procesos	Organizacional
Rodríguez	2006	Conjunto de procesos sistemáticos (identificación y captación del capital intelectual; tratamiento, desarrollo y compartimiento del conocimiento; y su utilización) orientados al desarrollo organizacional o personal y, consecuentemente, a la generación de una ventaja competitiva para la organización o el individuo.	Procesos	Organizacional-Económico
Flores Urbáez y Peña-Cedillo	2008	Proceso organizacional dirigido a crear una cultura del compartir, generar, orientar, aplicar y evaluar el conocimiento con la finalidad de ser aplicado por los miembros de la organización para hacerla más productiva y competitiva a través de procesos, productos y servicios innovadores, permitiendo tomar decisiones exitosas en entornos dinámicos.	Humano productiva - Procesos	Organizacional-Económico
Daft	2010	Esfuerzo sistemático en encontrar, organizar y dar acceso al capital intelectual de la organización e introducir una cultura de aprendizaje continuo y de compartir conocimiento, de tal forma que las actividades de la organización puedan basarse en el conocimiento existente.	Explotación	Organizacional

Tabla 4. Unión y clasificación de definiciones de GC bajo los criterios de clasificación de Rueda Martínez y Pérez & Urbáez. Fuente: Elaboración propia tomando como base las tablas 2 y 3

El presente trabajo se enfoca en base las perspectivas humano-productiva y de procesos (según los criterios de clasificación de Rueda Martínez). Humano-productiva porque el conocimiento proviene de los humanos y son ellos quienes lo utilizan durante su

permanencia en la organización, para desempeñar de forma adecuada su trabajo y de esta manera generar ventajas competitivas a la misma. La organización a su vez puede retener y convertir este conocimiento en un activo, almacenarlo y distribuirlo entre sus demás miembros. Además, el conocimiento presente en la industria del software es altamente especializado y vinculado a las personas que lo utilizan.

Otra definición de GC que puede ser vista bajo el enfoque humano-productiva es la de Hernández (2018).

La G.C. es la función, consciente y planificada, con métodos profesionales, que integra la gestión de datos e información, del conocimiento y de los aspectos cognoscitivos y emocionales de la inteligencia, a través de la comunicación, que involucra a la organización como un todo, así como a sus grupos e individuos en compleja interacción, y que tiene como resultado la formación compartida de conocimientos, habilidades, capacidades (inteligencia en su sentido integral cognoscitivo y afectivo-volitivo), motivación, intereses, ideales (visión compartida), sentimientos y su integración en la personalidad (sentido personal, grupal, cultura). (P 96).

Por otra parte, la GC puede ser vista como un proceso que ayuda además a mejorar los procesos de la organización. En la industria del software, el desarrollo y calidad de los productos depende de los procesos y madurez de estos, por lo tanto, también coincide este trabajo con la perspectiva de procesos.

En cuanto a los criterios de clasificación de Pérez y Urbáez, tanto el criterio económico como el criterio organizacional son tenidos en cuenta, ya que se considera que ambos criterios se complementan, pues el conocimiento es un factor competitivo en las organizaciones y los procesos relacionados con la gestión de este deben ser sistematizados para poder, de esta manera, generar beneficios económicos a la misma.

Como se puede observar, tanto las clasificaciones de Martínez, como las de Pérez y Urbáez, a pesar de ser perspectivas diferentes, indican la importancia de la GC del conocimiento desde diversos puntos de vista.

Otro factor que remarca la importancia de la GC en una organización, relacionado también con las definiciones dadas anteriormente, es que el manejar de forma adecuada el conocimiento dentro de la misma, sumado a la cultura organizacional, aumenta considerablemente su valor de venta factible. Un ejemplo se observa en el caso de la compra

de la empresa Lotus, por IBM en el año 1995. Sobre este tema, Manucci (2008) indica lo siguiente:

Cuando el desarrollo de la industria de software estaba en sus inicios y las Empresas punto com no existían, una de las primeras adquisiciones marcó para siempre el valor del conocimiento como principal capital operativo. En 1995 IBM compra Lotus pagando 3.5 billones de dólares, 14 veces más del valor del software que era 250 millones de dólares. Claramente IBM no pagó todo este dinero por la empresa Notes y otros productos Lotus ni por las capacidades de producción y de venta. Los 3,5 billones extra que IBM pagó representan el valor del conocimiento y de la cultura de la empresa que inventó esos programas. Las mentes que inventaron Notes, son mucho más valiosas que el software en sí mismo. (p. 48).

Sin embargo, la GC es un área emergente que aún requiere de muchas investigaciones e innovación para adaptarse a los distintos rubros y explotar realmente los beneficios del conocimiento en la sociedad actual, de ahí se deriva en parte la importancia de este trabajo. En este contexto Canals y otros (2003) indican que:

[...] con la bien denominada o mal denominada nueva economía, economía del conocimiento o economía de la información, el conocimiento y la información son progresivamente más importantes, como recurso y también como producto. Ello hace que las empresas estén cada vez más preocupadas por cómo utilizan estos recursos, y también por lo que sucede cuando hay gente que abandona la organización, por ejemplo, como resultado de programas de reingeniería o de reducción de personal. Es entonces cuando, de repente, las empresas se dan cuenta de que personas que ellos creían que eran prescindibles en realidad tienen un conocimiento que es vital para la organización. (p.2)

Canals y otros (2003), destacan que otra de las razones por las cuales surge la GC es porque en la era de la globalización, las grandes empresas han visto la necesidad de que el conocimiento que poseen y generan en una parte del mundo, sea fácilmente aplicado en otras partes del mundo donde también tienen intereses, así como también la aparición de nuevas tecnologías que han aportado herramientas y metodologías que permiten hacer muchas cosas sobre el conocimiento que antes no se podían.

Existen distintos modelos que involucran guías y actividades diversas para gestionar en conocimiento en una organización (ver sección 3), sin embargo, Canals y otros (2003) indican que la GC se divide en dos partes fundamentales: creación y transmisión del conocimiento. Crear el conocimiento es la parte más complicada, pues depende de la persona, de la fuente y del tipo de conocimiento, mientras que la transmisión puede hacerse por diversos medios y puede perdurar en el tiempo. Estos dos procesos van a tener nombres diferentes y subdivisiones en cada uno de los modelos de GC que se explicarán en la sección 3.

2.4. Gestión del Conocimiento en la Ingeniería del Software

2.4.1. Ingeniería del Software

Antes de explicar las diferencias que existen en cuanto al conocimiento y el manejo de este en la ingeniería de software, es necesario definir qué es la Ingeniería del software.

Según Braude y Bernstein (2016) la ingeniería de software es una disciplina de la ingeniería que involucra todos los aspectos de desarrollo y mantenimiento de un producto de software. El Instituto de Ingeniería de Software (2000) en su manual *CMMISM for Systems Engineering/Software Engineering* define este término de la siguiente manera: “La ingeniería del software cubre el desarrollo de sistemas de software. [...] Se centra en aplicar enfoques sistemáticos, disciplinados, y cuantificables para el desarrollo, uso, y mantenimiento del software”.

2.4.2. El conocimiento en la Ingeniería del Software y los desafíos de la GC

Para Rus y Lindvall (2002), cuando se trata de procesos ingenieriles, existen dos nuevos tipos de conocimiento:

1. Conocimiento embebido en los productos (artefactos).
2. Metaconocimiento.

Para Rus y Lindvall (2002) el conocimiento está embebido en el producto porque es el resultado de actividades de alto intelecto y razonamiento.

Nicholson & Sahay (2004) indican que el conocimiento en el desarrollo de software puede estar embebido en varios productos (como procesos y herramientas de trabajo que se utilizan), procesos (procesos de desarrollo y metodologías de gerencia utilizadas) y prácticas (normas y comunicación). Sin embargo, ven el conocimiento embebido desde otro punto de vista. Para ellos el conocimiento embebido reside en relaciones entre los individuos de un

equipo y en las particulares normas, actitudes, flujo de información u forma de tomar las decisiones de este. Es decir, el conocimiento está embebido en principios, rutinas y procesos estandarizados que utiliza un equipo de desarrollo o una organización. Indica que este tipo de conocimiento no es tenido en cuenta en rutinas de GC.

En cuanto al metaconocimiento, Son y Terrance (2009) lo definen como “el conocimiento que tenemos acerca de nuestro conocimiento, a menudo expresado como juicios confiados acerca de lo que sabemos.” Para estos autores, el metaconocimiento es la forma más compleja de conocimiento que se haya estudiado.

Cox (2005) marca una relación estrecha entre el metaconocimiento, la inteligencia y el razonamiento de alto nivel. El monitoreo del conocimiento le permite al humano no solamente buscar soluciones a un problema, sino también determinar una estrategia efectiva para resolverlo. En relación con esto, Rus y Lindvall (2002) indican que el mayor problema en la industria del software es que, a diferencia de otros rubros, solo una pequeña fracción del metaconocimiento puede capturarse y hacerse explícita.

En la siguiente lista, se detallan los problemas o desafíos que presenta la GC en la industria del software para Rus y Lindvall (2002):

- La tecnología cambia constantemente y todos los días se produce conocimiento nuevo, además es necesario buscarlo fuera de los límites de la organización.
- Cada proyecto de software es diferente, tiene un contexto diferente y los procesos pueden variar de un proyecto a otro, las personas involucradas en el desarrollo de software a menudo están expuestas a esta diversidad.
- El desarrollo de software no sólo necesita conocimiento sobre su propio dominio, sino sobre el dominio para el cual se realizan los productos, por lo cual adquirir la experiencia y el conocimiento necesario para cada proyecto toma un tiempo largo.
- Los desarrolladores de software necesitan tener conocimiento muy específico acerca de las convenciones locales de desarrollo, este conocimiento se maneja como un “folklore” de manera informal.
- Dentro de una organización de desarrollo de software, es necesario saber “quién sabe qué cosa”, esto reduce los tiempos necesarios para buscar expertos que ayuden en alguna tarea. Además, ayuda a los empleados a conocer la importancia de sus conocimientos.

- El desarrollo de software de gran tamaño son normalmente una actividad grupal, la división del trabajo en fases significa a menudo que varios grupos se involucran en el proceso de desarrollo, y con la globalización es posible que estos grupos diversos estén en locaciones geográficas y zonas horarias diferentes.
- El conocimiento en la industria del software es mayormente implícito, y no se vuelve explícito en primera instancia por la falta de tiempo.
- El conocimiento es altamente especializado, por lo cual es difícil de formalizar.

Aurum y otros (2013) indican también problemas con los que se encuentra la GC en la industria del software. Explican que la alta rotación de personal en el rubro ocasiona que el conocimiento sea más volátil. Además, las posibilidades de que los proyectos fracasen o se cancelen son elevadas, por lo cual se debe registrar también qué cosas no se deben hacer para que el proyecto tenga éxito.

Siguiendo en la línea de los desafíos de la GC en la ingeniería de software, Dingsoyr y Smite (2014), indican que cuando existen proyectos globales, con grupos divididos en locaciones geográficas diferentes, “las distancias temporales y geográficas afectan en primera instancia la habilidad para compartir y acceder al conocimiento” (p. 3).

Otro factor que se debe tener en cuenta y que pueden afectar la forma en que el conocimiento se genera, comparte y mantiene en relación a otros rubros, es que en un mismo proyecto de desarrollo de software existen diversos roles, con conocimientos y experticias diferentes y el conocimiento debe ser accesible y entendible por todos los miembros del proyecto, además pueden dentro de un mismo rol, existir diferentes niveles de experticia (*seniority*) o incluso experiencia en diferentes tecnologías (por ejemplo pueden haber desarrolladores especializados en diferentes tecnologías involucrados en el mismo proyecto).

Las características de la industria del software que pueden resultar un desafío para la gestión del conocimiento y deben considerarse se pueden encontrar resumidas en los siguientes ítems:

- cambios constantes en las tecnologías utilizadas,
- necesidad de buscar conocimiento fuera de la organización,
- contextos y procesos difieren en cada proyecto,

- se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto,
- se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal),
- es necesario saber “Quién sabe qué cosa” para reducir tiempos,
- el conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo,
- conocimiento altamente especializado, difícil de formalizar,
- alta rotación del personal vuelve al conocimiento volátil,
- altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo,
- en equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento,
- en un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles,
- pueden existir dentro de un mismo rol diferentes niveles de experticia (*seniority*),
- pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes).

2.5. Fábricas de Software

Uno de los mejores ejemplos de aplicación de procesos de ingeniería de software, que ha tenido mucho auge, debido a la necesidad cada vez más fuerte de producir software en cortos períodos de tiempo, son las llamadas Fábricas de software o *Software Factories* (SF de ahora en adelante). Para Bovea y García (2011) en la próxima década la demanda de software aumentará de manera significativa, promovida por los cambios económicos mundiales, las nuevas aplicaciones tecnológicas, al auge de los dispositivos móviles, el internet, la inteligencia artificial, por lo cual la forma de suplir esta demanda no será buscando más desarrolladores, sino cambiando radicalmente los procesos de producción de software. Es este cambio en la demanda de software lo que ha impulsado el crecimiento de las SF.

El término SF emerge por primera vez entre los años 60 y 70, en Estados Unidos y Japón, pero no es sino hasta los años 90, en donde se comienza a dar forma a este concepto.

Durante el desarrollo del presente trabajo, fueron encontrados 3 enfoques fundamentales en los que se pueden clasificar las definiciones de SF. Estos enfoques son:

- **enfoque de procesos y profesionales especializados:** esta clasificación engloba a las definiciones que hacen énfasis en la especificación de procesos delimitados y definidos para la producción de software en masa;
- **enfoque de plantilla:** existen autores que definen a la SF como una plantilla de trabajo, que se debe seguir dentro de una organización para lograr la producción de software en masa;
- **enfoque económico:** dentro de esta clasificación están las definiciones que se centran más en los beneficios económicos de la producción de software en masa que en su caracterización como tal.

En la tabla 5, se puede encontrar una clasificación de diversas definiciones de SF, de distintos autores, según el criterio anterior.

Autor	Año	Definición	Enfoque
Fernstromy Narfelt	1992	"El concepto de SF simboliza un cambio de paradigma deseado entre la producción de software mediante mano de obra intensiva, a un estilo basado en capital, en el que se pueden realizar inversiones importantes a un nivel de riesgo aceptable. La SF representa un paso evolutivo en la escala de soporte de ingeniería de software, un refinamiento de ambientes de desarrollo de software y entornos de soporte de proyectos integrados."	Económico
Crnkovic	2000	Al encaminarse hacia el proceso de desarrollo del producto aparecen las fábricas de software con el fin de mejorar la productividad, éstas a través de un banco de conocimientos, componentes, herramientas y procesos utilizan la reutilización para otorgar al cliente una mayor calidad, flexibilidad, menores precios y tiempos de entrega	Económico
Greenfield y Short	2003	"Una plantilla de software puede ser cargada en una herramienta, tal como un Ambiente de Desarrollo Interactivo (IDE por sus siglas en inglés), para producir un tipo específico de producto. De la misma manera una plantilla de documento puede ser cargada en un editor de documentos, como Word o Excel, para producir un tipo de documento específico. Una IDE configurada con una plantilla de software para una familia de productos se convierte en una fábrica para producir miembros de dicha familia. Esto es a lo que llamamos SF ... Podemos ahora definir una SF como un modelo guía para producir una línea de productos"	Plantilla
Short	2004	"Es una línea de producción de software que configura herramientas extensibles, procesos y contenido mediante una plantilla de fábrica de software basada en un esquema de fábrica de software para automatizar el desarrollo y mantenimiento de las variantes de un producto arquetípico mediante la adaptación, montaje y configuración de marco basado en componentes"	Plantilla
Nomura y Otros	2007	Para que el proceso total de creación de productos de software sea efectivo en una SF, es necesario contar con modelos de desarrollo de software basados en procesos claros y bien definidos, profesionales especializados y una reutilización productiva de activos de software.	Procesos
Santos y Soares	2013	"El concepto adoptado de SF está relacionado con unidades de desarrollo estructuradas, con roles y responsabilidades claras, soportada por herramientas y procesos bien definidos."	Procesos

Tabla 5. Clasificación de definiciones de SF. Fuente: Elaboración propia.

De lo anteriormente descrito, se puede concluir que no existe una definición unificada de lo que es, o lo que representa, una SF, sin embargo, con base en las definiciones anteriores se puede inferir que una SF es un modelo de referencia, plantilla o forma de trabajo, que se

puede aplicar en empresas que desarrollan productos de software, basado en equipos o células de trabajo multidisciplinarios, conformados por profesionales especializados, con roles y responsabilidades claras, que siguen procesos bien definidos, para producir una familia de productos de software.

Los problemas descritos en el punto anterior relacionados con la GC en la ingeniería de software, también están presentes en las SF. Además se introducen nuevos desafíos: por un lado cómo se busca aumentar la productividad y reducir los tiempos de entrega de productos, esto hace que en las SF haya menos tiempo para dedicarse a la GC, por otro lado, como se puede observar en algunas de las definiciones planteadas anteriormente, las SF reutilizan elementos para aumentar la productividad, por lo cual el conocimiento también debería ser reutilizado cuándo sea necesario, introduciendo así otro desafío para la GC en este tipo de organización.

3. DESCRIPCIÓN DEL PROBLEMA

3.1. Identificación del problema de investigación

En la sección anterior se habló de los desafíos que presenta la GC en la industria del software, específicamente en el rubro de las fábricas de software, siendo el resultado 15 desafíos en total (Ver punto 2.4.2).

Es importante mencionar que dichos desafíos resultantes de la investigación documental han sido triangulados mediante entrevistas informales con los expertos que participaron del proceso de validación de esta investigación (ver capítulo 5), estando todos de acuerdo con los desafíos mencionados.

Con el objetivo de lograr utilizar el conocimiento de forma tal que se puedan obtener ventajas competitivas derivadas de este, existen diversos modelos de gestión del conocimiento creados por investigadores u organizaciones para guiar y facilitar el proceso; estos modelos han sido creados a medida que las organizaciones y la sociedad han ido notando la importancia del conocimiento y su adecuado manejo. Si no existe un modelo de GC capaz de manejar los desafíos planteados en el punto anterior, es probable que este tipo de organización (FS) esté perdiendo la importancia del conocimiento y su gestión; es decir, está perdiendo las ventajas competitivas que implica gestionar el conocimiento

Durante este trabajo de investigación, mediante investigación documental, se han analizado los principales modelos de gestión del conocimiento y aquellos que están un poco más relacionados con la ingeniería del software o FS.

3.1.1. Modelos actuales de gestión del conocimiento y su relación con los desafíos de la GC en SF

A medida que las organizaciones y la sociedad han ido notando importancia de la gestión del conocimiento, diversos investigadores han propuesto una serie de pasos o guías para identificar cómo debe ser llevado a cabo el proceso de GC para que sea exitoso y genere los beneficios esperados para la organización, estas propuestas las han hecho mediante modelos de GC.

Según la Real Academia Española (2014), un modelo es un “arquetipo o punto de referencia para imitarlo o reproducirlo”.

En relación con esto, Pérez y Urbáez (2016) indican que los investigadores en las últimas décadas, a través de sus propuestas teóricas y de sus investigaciones de experiencias en empresas, han intentado aproximarse a explicar cómo debe organizarse, gestionarse y fluir el conocimiento en la organización.

Gómez (2006) indica que “La multidisciplinariedad inherente al estudio de la gestión del conocimiento supone la existencia de diferentes perspectivas para el desarrollo y el estudio de los sistemas y modelos de gestión del conocimiento”. (p. 29).

Con base en estos autores se puede concluir que un modelo de GC es un conjunto de aproximaciones a explicar cómo debe ser gestionado el conocimiento, basados en investigaciones y experiencias y enfocados en una disciplina en particular, que sirven como referencia o guía a las organizaciones o entidades en los que se podrían aplicar.

Además, Gómez (2006) sugiere que los modelos de GC existentes pueden ser clasificados en tres categorías, correspondiendo cada una de ellas al núcleo o centro en el que se basa dicho modelo. Estas categorías son:

- Almacenamiento, acceso y transferencia de conocimiento: son aquellos modelos que buscan almacenar el conocimiento de la organización en repositorios de fácil acceso, y lo conciben como una entidad independiente de quién lo produce o lo utiliza.
- Sociocultural: modelos que buscan generar una cultura organizacional y promueven cambios de actitud en los miembros de la organización, fomentando la confianza y creatividad, y concientizando sobre la importancia del conocimiento y su gestión.
- Tecnológicos: modelos que promueven el uso de sistemas y herramientas tecnológicas para acompañar el proceso de GC.

Son muchos los modelos de GC que existen en la actualidad. Heisig (2009) analiza 160 modelos generales, según varios descriptores y clasificaciones, sin embargo, son modelos cuya fecha no supera al año 2001 y la mayoría de ellos no han sido referenciados en investigaciones posteriores, lo cual lleva a pensar que probablemente estén en desuso.

En este contexto, en el presente apartado analiza modelos generales (aplicados a organizaciones sin distinguir rubro). Se realiza una revisión sistemática de los modelos existentes y se seleccionan aquellos que no derivan de otros modelos o que incluyen cambios significativos con respecto a los demás. Como estos modelos no cubren la mayoría de los desafíos específicos del rubro de las SF, ya que son modelos muy generales, se analizan también aquellos modelos orientados al rubro de la ingeniería de software que tienen en cuenta más desafíos de la GC en el rubro.

3.1.2. Modelos más conocidos

- Modelo de Wigg

Wigg (1993) propuso uno de los primeros modelos o guías para la GC en las organizaciones, haciendo énfasis en la falta de este tipo de guías en su época, y en la importancia de gestionar el conocimiento para obtener ventajas competitivas para la organización.

El modelo de GC propuesto por Wigg consta de cuatro etapas principales relacionadas al conocimiento: construir, retener, agrupar y utilizar.

En la tabla 6 se muestran cada una de estas etapas, sus divisiones y las principales actividades que la integran.

Etapa	Sub-Etapas	Actividades
Construir conocimiento	Obtener conocimiento	Crear nuevo conocimiento
		Traer conocimiento de fuentes existentes
		Observar el mundo real
	Analizar el conocimiento	Extraer lo que parece ser conocimiento de la fuente utilizada.
		Abstraer el conocimiento
		Identificar patrones
		Explicar las relaciones entre los diferentes elementos
		Verificar que el material extraído y procesado se corresponde con la fuente.
	Reconstruir (resumir) el conocimiento	Generalizar el material analizado
		Generar hipótesis
		Establecer concordancia entre el conocimiento nuevo y el ya existente
		Actualizar el repositorio de conocimiento con el nuevo conocimiento.
	Codificar y modelar el conocimiento	Representar el conocimiento en nuestras mentes
		Modelar el conocimiento
Documentar el conocimiento en libros y manuales		
Guardar el conocimiento en bases de conocimiento		
Organizar el conocimiento	Organizar el conocimiento por usos específicos.	
	Organizar el conocimiento según el tipo de negocio.	
Retener el conocimiento	Recordar el conocimiento	Retener el conocimiento como individuo.
	Acumular el conocimiento en repositorios	Crear una base de datos de conocimiento
	Incluir el conocimiento en procesos	Crear procesos que incluyan el conocimiento adquirido
	Archivar el conocimiento	Crear librería científica
		Eliminar el conocimiento desactualizado de los repositorios actuales.
Agrupar el conocimiento	Coordinar el conocimiento	Desde equipos colaborativos para trabajar en una situación en particular
		Crear "Redes de acceso a los expertos", identificar quién sabe qué cosa.
	Armar el conocimiento	Reunir fuentes de conocimiento
	Acceder y recuperar el conocimiento	Consultar con gente que tenga conocimiento sobre cómo resolver un problema en particular
		Obtener el conocimiento directamente de un repositorio

Aplicar (usar) el conocimiento	Utilizar el conocimiento para desarrollar tareas rutinarias	Hacer productos estandarizados Proveer servicio sestandarizados Usar la "red de expertos"
	Utilizar conocimiento para examinar las situaciones excepcionales	Inspeccionar la situación para determinar cuál es el problema.
	Utilizar el conocimiento para describir una situación y determinar el alcance.	Identificar y describir el problema para determinar cómo debe ser manejado en general.
	Seleccionar conocimiento importante especial para manejar la situación	Identificar las fuentes de experticia para manejar la situación especial
	Observar y caracterizar la situación con conocimiento especial	Examinar la situación y organizar las observaciones para resolverla con conocimiento
	Analizar la situación con conocimiento	Determinar los patrones y comparar con situaciones conocidas
	Pensar soluciones alternativas con conocimiento	Identificar las opciones y crear nuevas soluciones.
	Evaluar alternativas potenciales utilizando el conocimiento	Identificar ventajas y desventajas de todas las alternativas.
	Utilizar el conocimiento para decidir qué hacer	Seleccionar la alternativa más efectiva y chequear que cumple con los criterios requeridos.
	Implementar la alternativa seleccionada	Ejecutar las tareas asociadas con la implementación de la alternativa seleccionada

Tabla 6. Etapas, sub-etapas y actividades del modelo de GC de Wiig. Fuente: Elaboración propia en base al modelo de Wiig (1993)

Según Garvin (1993) una organización que aprende es una organización capacitada para la creación, adquisición y transferencia de conocimiento, así como para la modificación de su comportamiento para reflejar el nuevo conocimiento adquirido.

Además de las etapas y actividades, Wiig (1993) en su modelo, explica las características que debe tener una “organización que aprende”.

- La organización debe determinar los diferentes flujos de información dentro de la misma para poder convertirla en conocimiento y transferirlo.
- La organización aprende si los individuos aprenden, por lo tanto, la organización debe contratar personas dispuestas a aprender.
- Se considera el conocimiento en tres niveles: personal, grupal y organizacional. Estos tres niveles deben existir para que la organización aprenda, por lo tanto, el trabajo grupal debe existir dentro de la organización.
- La cultura organizacional debe influenciar a los empleados a admitir que no saben algo y buscar ayuda en otras personas o departamentos que conozcan sobre el problema.
- No hay necesidad de “reaprender” si alguien en la organización ya ha asumido el mismo conocimiento, es por esto que es importante saber “quién sabe qué cosa”.
- La organización debe permitir e influenciar la innovación y generación de conocimiento en cualquier nivel. Si los superiores no están dispuestos a que personas con menor cargo sean innovadores, entonces será más difícil gestionar el conocimiento.
- Los expertos en la organización deben seguir aprendiendo y renovar sus conocimientos periódicamente.
- Wiig propone el uso de tecnologías, pero solamente para almacenar el conocimiento.

Se puede observar que el modelo de Wiig es muy general, describe etapas y actividades generalizadas, pero no especifica pasos detallados o procesos para gestionar el conocimiento en la organización, queda en la misma definir la forma en la que se adaptaría este modelo.

Según los criterios de Gómez (2006) explicados anteriormente, se puede categorizar este modelo bajo el apartado: almacenamiento, acceso y transferencia de conocimiento.

En cuanto a los desafíos que representa la GC en la ingeniería de software y en las SF, mencionados anteriormente, se puede observar en la siguiente tabla (ver tabla 7) cuáles aspectos son considerados en el modelo de Wiig.

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	PARCIAL	Habla de eliminar el conocimiento desactualizado, sin embargo, no destaca la frecuencia.
Necesidad de buscar conocimiento fuera de la organización	SI	Habla de observar al mundo real, es decir, traer conocimiento externo
Contextos y procesos difieren en cada proyecto	NO	
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	NO	
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	NO	
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	SI	Indica que no es necesario que otra persona genere lo que alguien ya sabe en la organización, para eso hay que saber quién lo sabe
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	NO	
Conocimiento altamente especializado, difícil de formalizar.	NO	
Alta rotación del personal vuelve al conocimiento volátil.	NO	
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	NO	
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	NO	
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	NO	
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	NO	
Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	NO	
Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	NO	

Tabla 7. Desafíos de la GC en el desarrollo de software en relación con el modelo de GC de Wiig. Fuente: Elaboración propia.

- Modelo de Nonaka y Takeuchi

Nonaka y Takeuchi (1995), propusieron un modelo de GC que validaron con casos de estudio en organizaciones de Japón. Centrarón su modelo en la importancia de convertir el conocimiento implícito en conocimiento explícito.

Para convertir el conocimiento de implícito a explícito, se siguen 4 fases principales, en un modelo permanente en espiral. Este modelo se centra en lo personal, grupal, organizacional e interorganizacional.

Para Nonaka y Takeuchi, el conocimiento es creado por los trabajadores de primera línea mediante procesos de investigación y desarrollo, se asume entonces que el trabajador ha adquirido de esta manera conocimiento implícito y es necesario convertirlo en explícito. El conocimiento en este modelo debe ser un compromiso significativo para la organización y debe ser justificado.

Las etapas para convertir el conocimiento implícito en explícito según este modelo son las siguientes:

- **Socialización:** el conocimiento es compartido con otras personas mediante diálogo, conversación, imitación o guiando a la otra persona. Puede incluir también socializar con los usuarios o clientes para crear conocimiento desde distintas perspectivas.
- **Externalización:** implica convertir el conocimiento compartido en la fase anterior en conocimiento explícito, en fuentes accesibles para otras personas: bases de datos, manuales, guías. Este conocimiento puede ser refinado posteriormente.
- **Combinación:** consiste en combinar conocimiento desde distintas fuentes de conocimiento explícito. Para los autores, los sistemas informáticos pueden jugar un papel muy importante en esta fase.
- **Internalización:** es la recepción del conocimiento por una persona, desde una fuente de conocimiento explícito o desde otra persona.

En la siguiente imagen (ver figura 1), se muestra el proceso espiral de las fases explicadas.

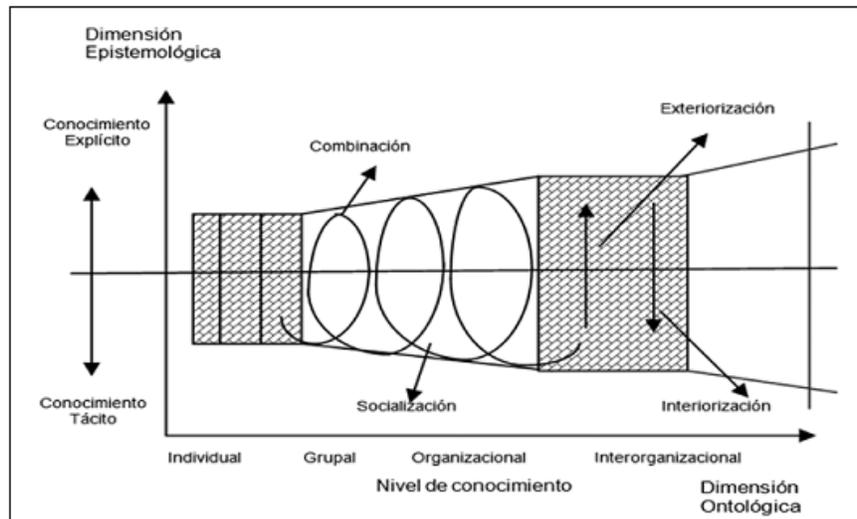


Figura 1: Modelo de Nonaka y Takeuchi. Fuente: Nonaka y Takeuchi (1995). P. 73.

Traducción de la imagen: Díaz, M. T. R., & Millán P.96 (2014).

Por otra parte, en este modelo la organización debe apoyar el proceso de conversión del conocimiento de la siguiente manera:

- Intención y compromiso de la organización.
- Autonomía en todos los niveles.
- Flexibilidad y caos creativo (romper estándares en pro de la GC).
- Redundancia (repetir el conocimiento en toda la organización).
- Requerimiento de variedad (diversidad de personal y de conocimiento).

El presente modelo también toma en cuenta la GC en organizaciones globales con culturas diferentes, indicando que la socialización entre extranjeros es la mejor forma de superar estas diferencias.

También se puede destacar que este modelo, al igual que el anterior, es general, sin estar aplicado a ningún rubro y sin mucho detalle de cómo debe ser aplicada cada una de las fases que propone.

Según los criterios de Gómez (2006) explicados anteriormente, se puede categorizar este modelo bajo el apartado: almacenamiento, acceso y transferencia de conocimiento, así mismo, se puede clasificar también como sociocultural, pues busca establecer una cultura organizacional que apoye el proceso de GC.

En la siguiente tabla (ver tabla 8) se relacionan los desafíos de la GC en la ingeniería de software y en las SF con el modelo de GC de Nonaka y Takeuchi.

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	NO	
Necesidad de buscar conocimiento fuera de la organización	PARCIAL	Si, propone hablar con clientes y proveedores para tener otra visión, sin embargo, no habla de otro tipo de fuentes externas
Contextos y procesos difieren en cada proyecto	NO	
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	NO	
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	PARCIAL	No habla directamente de esto, pero el proceso de socialización podría resolverlo.
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	NO	
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	NO	
Conocimiento altamente especializado, difícil de formalizar.	PARCIAL	Propone la socialización como medio para compartir conocimiento implícito, pero no hace referencia al conocimiento altamente especializado
Alta rotación del personal vuelve al conocimiento volátil.	NO	
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	NO	
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	PARCIAL	Propone la socialización para solucionar diferencias culturales, pero no indica nada de los distintos horarios.
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	NO	
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	NO	
Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	NO	

Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	NO	
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----	--

Tabla 8. Desafíos de la GC en el desarrollo de software con relación al modelo de GC de Nonaka y Takeuchi. Fuente: Elaboración propia.

- **Modelo de Sveiby**

Sveiby (2001) creó un modelo de GC orientado a guiar la generación de una estrategia corporativa, afirmando que el conocimiento es un activo intangible, y al contrario de los activos tangibles, que se devalúan a medida que se usan, el conocimiento adquiere valor cuando se usa y se devalúa como activo cuando entra en desuso. En la siguiente imagen se puede observar en detalle el modelo propuesto por este autor.

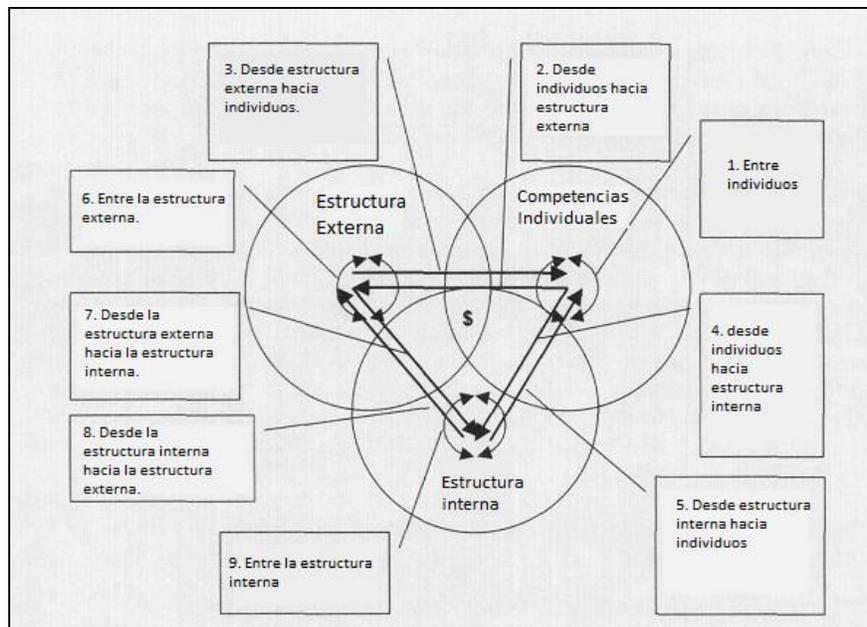


Figura 2: Modelo de Sveiby. Fuente: Sveiby (2001). Traducción propia.

En este modelo, los integrantes de la organización son tomados como principales actores. Se basa en la transferencia de conocimiento entre 9 aspectos de la organización (se pueden observar en la figura 2), y para cada aspecto existe una pregunta estratégica que debe ser respondida.

1. **Entre individuos:** consiste en mejorar la transferencia de conocimiento entre individuos de la organización, para ello sugiere responder a la pregunta: ¿Cómo podemos mejorar la transferencia de competencias entre miembros de la organización? Para ello aconseja enfocarse en actividades que aumenten la confianza entre empleados, actividades grupales, rotación de responsabilidades o esquemas maestro-aprendiz.
2. **Desde individuos hacia la estructura externa:** consiste en definir cómo los miembros de la organización van a transferir sus conocimientos al mundo exterior. La pregunta estratégica en este caso es: ¿Cómo van a ayudar los empleados a mejorar las competencias de clientes, proveedores y otros interesados? Para responderla propone actividades enfocadas en que los empleados ayuden a los clientes a aprender de los productos como: rotación de puestos de trabajo con los clientes, seminarios de productos, y educación del cliente.
3. **Desde la estructura externa hacia individuos:** se basa en la adquisición de conocimiento por parte de los empleados de fuentes externas como clientes, proveedores o la comunidad en general. La pregunta para responder en este caso es: ¿Cómo pueden las fuentes externas mejorar las competencias de los empleados? Para responderla se tiene que buscar una estrategia que mantenga buenas relaciones entre los empleados y el mundo exterior.
4. **Desde los individuos hacia la estructura interna:** la idea en este caso es convertir el conocimiento de los empleados en repositorios de datos accesibles a toda la organización. La pregunta para responder es: ¿Cómo podemos convertir los conocimientos individuales en sistemas, herramientas y plantillas? En esta parte, el autor indica que muchos gerentes tienden a creer que ésta es la única actividad que se debe llevar a cabo, sin embargo, aplicarla sola no garantiza el éxito en la GC.
5. **Desde la estructura interna hacia los individuos:** este punto es la contraparte del anterior. Una vez que la información ha sido capturada en sistemas, es necesario que llegue a los demás miembros de la organización para poder mejorar sus competencias individuales. La pregunta en este punto es: ¿Cómo podemos mejorar las competencias individuales utilizando sistemas, herramientas y plantillas? Para responderla, el autor sugiere las mejoras en la interacción hombre-máquina con los sistemas, simulaciones y ambientes de *e-Learning*.

6. **Entre la estructura externa:** este punto incluye la comunicación entre los clientes relacionada a los servicios de la organización. La pregunta estratégica es ¿Cómo podemos mejorar la comunicación entre los clientes? Para responderla propone mejorar la calidad de los productos, mejorar la imagen de la empresa, hacer seminarios de productos y generar alianzas corporativas.
7. **Desde la estructura externa hacia la estructura interna:** concierne a qué conocimiento puede adquirir la organización desde el mundo exterior y cómo lo aprendido puede convertirse en acción. La pregunta en este caso es: ¿Cómo pueden las competencias de los clientes, proveedores y otros interesados mejorar las herramientas, procesos y productos de la organización? Para responderla sugiere mejorar los centros de llamados para atender las quejas de los clientes, crear alianzas para generar nuevas ideas, alianzas de investigación y desarrollo, entre otras.
8. **Desde la estructura interna hacia la estructura externa:** esta es la contraparte del punto anterior y la pregunta estratégica es la siguiente: ¿Cómo pueden las herramientas, procesos y productos de la organización mejorar las competencias de los clientes, proveedores y otros interesados? Para ello propone mejorar los procesos y herramientas de la empresa en función de brindar servicio al cliente, como: extranets, rastreo de productos, e-business, entre otros.
9. **Entre la estructura interna:** para el autor, la estructura interna es la columna vertebral de la organización y la pregunta estratégica en este caso es: ¿Cómo pueden ser integrados eficientemente los sistemas, herramientas, procesos y productos de la organización? Para responderla propone sistemas informáticos integrados, bases de datos compartidas, mejorar el diseño de la oficina, entre otras.

Según los criterios de Gómez (2006) explicados anteriormente, se puede categorizar este modelo bajo el grupo sociocultural pues involucra cambios de comportamiento tanto en la organización como en los individuos, también puede ser categorizado como tecnológico, pues promueve el uso de herramientas de tecnología informática.

Al igual que las propuestas anteriores, se puede notar que este modelo es muy general, no está orientado a un rubro específico y no detalla la forma en la que debe ser implementado (esto puede ser debido a la flexibilidad y generalidad de estos modelos).

En cuanto a los desafíos que presenta la GC en la ingeniería del software y en las SF, se puede observar en la tabla 9, si son, o no, tomados en cuenta por este modelo.

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	NO	
Necesidad de buscar conocimiento fuera de la organización	SI	En su ciclo destaca incluir conocimiento desde fuentes externas.
Contextos y procesos difieren en cada proyecto	NO	
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	NO	
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	PARCIAL	Habla de compartir conocimiento entre individuos mediante socialización, sin embargo, esto es conocimiento muy específico.
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	NO	
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	NO	
Conocimiento altamente especializado, difícil de formalizar.	PARCIAL	Habla de compartir conocimiento implícito por socialización, pero no habla de conocimiento altamente especializado.
Alta rotación del personal vuelve al conocimiento volátil.	NO	
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	NO	
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	NO	
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	PARCIAL	Propone la rotación de responsabilidades, sin embargo, en el desarrollo de software los roles no se pueden rotar porque son especializados.
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	NO	
Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	NO	

Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	NO	
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----	--

Tabla 9. Desafíos de la GC en el desarrollo de software en relación con el modelo de GC Sveiby. Fuente: Elaboración propia.

- **Modelo de Earl**

Earl (2001), propone un modelo de GC dividido en lo que él llama “escuelas de gestión del conocimiento”.

Dingsoyr y Smite (2014) hacen referencia a este modelo, indicando que es muy utilizado en ingeniería de software, y lo aplican para resolver problemas de GC en equipos globalmente distribuidos.

Las llamadas escuelas en este modelo, son ramas o variantes de este, que se pueden combinar a conveniencia de la organización, según las prioridades y objetivos de esta. Cada escuela del modelo de Earl tiene una filosofía, ésta indica hacia donde se orienta la escuela, es decir, qué es lo que se busca al aplicarla. Los elementos de cada una de las escuelas pueden observarse en la tabla 10.

Escuela	Foco	Objetivo	Factores de éxito	Aporte	Filosofía
Sistemas	Tecnología	Bases de conocimiento	Validación de contenido, incentivos para generar contenido	Sistemas basados en conocimiento	Codificación
Cartográfica	Mapas	Directorios de conocimiento	Cultura/ incentivos para producir conocimiento, redes que conecten a las personas	Perfiles y directorios	Conectividad
Ingeniería	Procesos	Flujos de conocimiento	Distribución de la información y aprendizaje	Bases de datos compartida	Capacidad
Comercial	Ingresos	Evaluaciones de conocimiento	Equipos especializados/ procesos institucionalizados	Sistemas de registro y procesamiento, Evaluación intelectual	Comercialización
Organizacional	Redes	Agrupación de conocimiento	Intermediarios del conocimiento/ cultura social	Agrupaciones e intranets	Colaboración
Espacial	Espacio	Intercambio de conocimiento	Diseño para un propósito	Herramientas de acceso y representación	Conectividad (contacto)
Estratégica	Mentalidad	Capacidades de conocimiento	Artefactos retóricos	Eclécticos	Conciencia

Tabla 10. Escuelas de GC del modelo de Earl. Fuente: Elaboración propia en base al modelo de Earl (2001).

- **Escuela de sistemas:** su objetivo es generar base de conocimientos utilizando la tecnología disponible. La idea es recolectar conocimiento especializado y hacerlo accesible para otros especialistas y facilitar la toma de decisiones. Deben ser de un dominio específico y la organización debe soportar tareas de extracción y manejo de este tipo de conocimiento. El conocimiento no debe ser solo proveniente de fuentes de datos típicas sino también de la experiencia ganada por los especialistas al realizar sus tareas. Se sugieren recompensas por generar y almacenar conocimiento útil. La filosofía de esta escuela es codificar información específica relacionada al dominio de la organización.
- **Escuela Cartográfica:** la idea de esta escuela es relacionar el conocimiento de la organización con las personas que lo tienen. Es saber quién conoce cada cosa. Para eso se propone la construcción de directorios de conocimiento. Como el conocimiento puede ser difícil de convertir de implícito a explícito, la idea no es crear repositorios de conocimiento, sino proveer a los miembros de la organización la conexión necesaria con otros miembros que posean ese conocimiento para luego compartirlo. Se debe incentivar el compartir el conocimiento, más que el almacenarlo.
- **Escuela de ingeniería:** el objetivo de esta escuela es proveer a las personas que realizan una actividad el conocimiento necesario para realizarla. Implica la reutilización del conocimiento para mejorarlo con cada uso, lo cual conlleva a la mejora de los procesos de la organización.
- **Escuela comercial:** el principal objetivo de esta escuela es utilizar el conocimiento para producir valor económico a la organización. Es la más relacionada con explotación del conocimiento que con exploración de este. Tiene en mente proteger el conocimiento de la organización e incluso generar patentes.
- **Escuela organizacional:** implica el uso de las estructuras de la organización para compartir y agrupar el conocimiento. De aquí nacen las “comunidades del conocimiento” que son grupos de personas con actividades en común, o que pertenecen a la misma partición de la estructura de la organización que comparten conocimiento. Esta escuela funciona en organizaciones que tienen una cultura de comunicación y socialización arraigadas.

- **Escuela espacial:** en esta escuela la distribución espacial o física es muy importante y está diseñada para compartir conocimiento. La idea es fomentar la socialización entre los miembros de la organización para compartir conocimiento implícito, basado en la idea de que el ser humano es un ser social y prefiere conversar antes que sacar conocimiento de fuentes escritas o formales.
- **Escuela estratégica:** esta escuela ve la GC del conocimiento como parte de la estrategia competitiva de la organización. La idea es construir, nutrir y explotar el conocimiento a través de sistemas, procesos y personas y convertirlo en valores, cómo productos o servicios basados en el conocimiento.

Se puede observar que este modelo, aunque es flexible, no especifica un proceso determinado para implementar cada una de las escuelas que lo componen.

Según el criterio de Gómez (2006) este modelo se podría clasificar bajo cualquiera de las tres categorías: almacenamiento, acceso y transferencia de conocimiento, sociocultural o tecnológico, ya que cada escuela está orientada a casos diferentes. Por ejemplo: la escuela de sistemas puede ser clasificada como tecnológica porque promueve el uso de tecnologías, la escuela de espacio puede ser clasificada como sociocultural porque promueve la socialización entre miembros de la organización y la escuela de ingeniería puede ser clasificada como almacenamiento-acceso-transferencia ya que promueve estas actividades para mejorar los procesos de la organización.

Unificando las 7 escuelas, se puede observar en la tabla 11, qué desafíos de GC en ingeniería de software y SF son tomados en cuenta en este modelo.

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	NO	
Necesidad de buscar conocimiento fuera de la organización	NO	
Contextos y procesos difieren en cada proyecto	NO	
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	SI	
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	Parcial	Habla del conocimiento implícito e informal, sin embargo, este es conocimiento muy específico.
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	SI	Esto es tratado en la escuela cartográfica.
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	Parcial	No habla de tiempo, pero si de conocimiento implícito
Conocimiento altamente especializado, difícil de formalizar.	SI	En la escuela de sistemas, menciona que el conocimiento debe provenir de la experiencia de los especialistas y se debe almacenar conocimiento específico.
Alta rotación del personal vuelve al conocimiento volátil.	NO	
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	SI	Habla de reutilización de conocimiento
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	NO	
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	SI	Las comunidades de conocimiento pueden tener efecto en este desafío.
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	SI	Las comunidades de conocimiento pueden tener efecto en este desafío.
Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	SI	Las comunidades de conocimiento pueden tener efecto en este desafío.
Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	SI	Habla de reutilización de conocimiento.

Tabla 11. Desafíos de la GC en el desarrollo de software en relación con el modelo de GC Earl. Fuente: Elaboración propia.

- Modelo de Kerschberg

Kerschberg (2001) propuso un modelo para gestionar el conocimiento, teniendo en cuenta que el mismo puede provenir de fuentes heterogéneas. Para el autor, las organizaciones utilizan en gran medida bases de datos de tipo “*data warehouse*”. Kerschberg (2001) indica que éstos son grandes repositorios de datos principalmente operacionales de los cuales se puede extraer conocimiento. Para Portal y Jaramillo (2017) un “*data warehouse*” es un repositorio de datos, el cual tiene como finalidad almacenar los datos provenientes de una o más bases de datos y fuentes operativas.

Este modelo, posibilita incluir en dichos repositorios otro tipo de información, proveniente de la creación de conocimiento por parte de los miembros de la organización (trabajadores del conocimiento).

La figura 3, muestra de manera gráfica el modelo propuesto por Kerschberg.

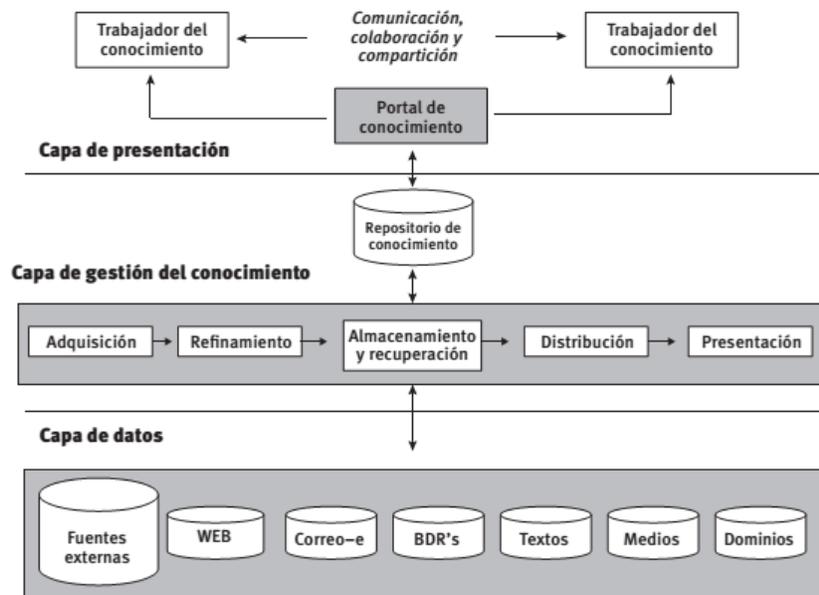


Figura 3: Modelo de Kerschberg. Fuente: Kerschberg (2001). P. 7.

Traducción de la imagen: Pérez y Urbáez (2016) P.213.

En la capa de presentación, los trabajadores del conocimiento se comunican y comparten en conocimiento. Éstos reciben el mismo a través del portal de conocimiento.

El portal del conocimiento puede contener sistemas que permitan al trabajador adquirir conocimiento a través de búsquedas personalizadas, así como también contribuir con la creación de conocimiento nuevo y convertir el conocimiento implícito en explícito.

La capa de GC, como su nombre lo indica, es en donde se siguen una serie de actividades para adquirir, refinar, almacenar, distribuir y presentar el conocimiento; esta capa provee información a la capa de datos y a su vez se alimenta de ésta. Puede incluir sistemas intermedios para crear conocimiento como por ejemplo los sistemas de minería de datos.

A continuación, se describen las actividades que conforman esta capa:

- **Adquisición:** durante esta etapa se captura el conocimiento mediante entrevistas y otras técnicas. Esto puede ser utilizado como reglas para sistemas expertos.
- **Refinamiento:** el conocimiento puede venir de diversas fuentes, durante esta etapa se organiza esta información, se clasifica y se establecen relaciones. El dominio y contexto debe estar claramente identificado. Pueden ser utilizadas técnicas de minería de datos para encontrar patrones.
- **Almacenamiento y recuperación:** en esta etapa el conocimiento es almacenado y se permite su fácil acceso por diversos criterios de búsqueda: Tema, autor, área, entre otros.
- **Distribución:** el conocimiento puede ser distribuido de muchas formas, entre ellas, un portal de la organización y correo electrónico se consideran buenas alternativas.
- **Presentación:** la presentación debe ser configurable de acuerdo con las necesidades de cada usuario, y debe permitir la colaboración entre los mismos como manera de convertir el conocimiento implícito en explícito.

Según los criterios de Gómez (2006) explicados anteriormente, se puede categorizar este modelo bajo el apartado: almacenamiento, acceso y transferencia de conocimiento, ya que se centra en este tipo de actividades para que el conocimiento sea fácil de acceder y compartir. También se puede

clasificar dentro del criterio tecnológico, pues promueve el uso de la tecnología como soporte a las actividades del modelo y como medios de almacenamiento de conocimiento.

En relación con los desafíos de la GC en la ingeniería de software y en las SF, se puede observar en la tabla 12, cuáles de estos son tomados en cuenta en el presente modelo.

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	NO	A pesar de que en la capa de GC se habla del uso de distintas tecnologías, no toma en cuenta que las herramientas de trabajo cambian constantemente.
Necesidad de buscar conocimiento fuera de la organización	PARCIAL	Aunque no nombra directamente el mundo exterior, indica que el conocimiento puede venir de diversas fuentes.
Contextos y procesos difieren en cada proyecto	SI	Habla de la identificación clara de los contextos y dominios.
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	PARCIAL	No habla directamente del tema, pero al indicar que se deben especificar los dominios, da a pensar que el dominio propio de la ingeniería puede ser tomado en cuenta si se identifica claramente.
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	NO	
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	NO	
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	NO	
Conocimiento altamente especializado, difícil de formalizar.	PARCIAL	Habla de convertir el conocimiento de implícito a explícito mediante la colaboración, pero no menciona técnicas aplicables a este caso.
Alta rotación del personal vuelve al conocimiento volátil.	NO	
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	NO	
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	NO	
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	NO	
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	NO	

Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	NO	
Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	NO	

Tabla 12. Desafíos de la GC en el desarrollo de software en relación con el modelo de GC de Kerschberg. Fuente: Elaboración propia.

- Modelo de Bustela y Amarilla

Bustelo y Amarilla (2001) proponen un modelo de GC enfocado principalmente en la gestión de la información, haciendo énfasis en el hecho de que sin una adecuada gestión de la información no puede existir la GC.

Bustelo y Amarilla (2001) definen a la gestión de la información como: “el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades” (p. 226). Para ellos, el centro de la gestión de la información es a su vez la gestión de la documentación (información que queda plasmada en documentos). Indican que la documentación puede ser de 3 tipos:

- Interna: documentos que surgen de la actividad de la organización.
- Externa: proviene de fuentes de información externa que los miembros de la organización necesitan consultar a menudo (libros, revistas, internet...).
- Pública: información que la organización produce de cara al público.

Suman a este tipo de información documental, la información proveniente de fuentes electrónicas como bases de datos.

En su modelo, Bustelo y Amarilla (2001) plantean los siguientes elementos que debe tener en cuenta una organización para pasar de gestión de la información a GC.

- 1- A diferencia de la gestión de la información, la gestión del conocimiento se aplica a toda la organización.

- 2- La gestión del conocimiento no solo involucra gestión de la información, sino que también involucra personas y procesos. De nada sirve tener un buen modelo de gestión de la información si no se utiliza. Por lo tanto, la GC y la gestión de recursos humanos deben estar vinculados, para ofrecer un ambiente laboral propicio que incentive la GC en la organización.

En la figura 4 se puede observar la visión de la GC que tiene los mencionados autores en su modelo.

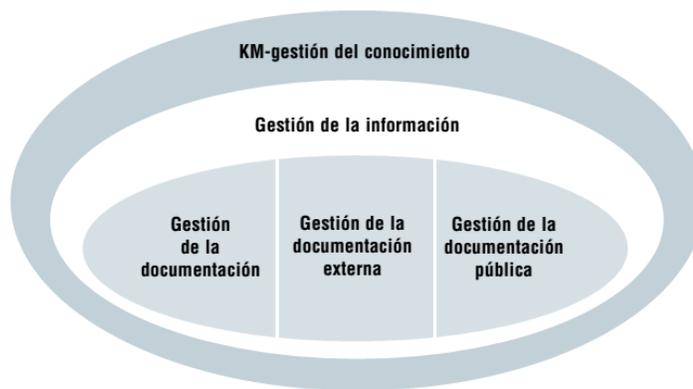


Figura 4: Modelo de Bustelo y Amarilla. Fuente: Bustelo y Amarilla (2001), P. 229.

Es importante destacar que en este modelo no se definen pasos ni procesos detallados que la organización debe seguir, sino simplemente una estructura global de lo que el modelo considera como GC.

Según los criterios de Gómez (2006) explicados anteriormente, se puede categorizar este modelo bajo el apartado sociocultural, ya que define actitudes que debe tener en cuenta la organización para pasar de gestión de la información a gestión del conocimiento. Además, propone que se involucre en el proceso a la gestión de recursos humanos para promover el mismo.

En cuanto a los desafíos que la GC presenta en las SF, en la tabla 13 se analiza si son, o no, tenidos en cuenta por este modelo y de qué manera.

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	NO	
Necesidad de buscar conocimiento fuera de la organización	SI	Toma la información externa como un tipo de documentación que debe ser gestionada.
Contextos y procesos difieren en cada proyecto	NO	
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	NO	
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	NO	
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	NO	
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	NO	
Conocimiento altamente especializado, difícil de formalizar.	NO	
Alta rotación del personal vuelve al conocimiento volátil.	NO	
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	NO	
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	NO	
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	NO	
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	NO	
Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	NO	
Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	NO	

Tabla 13. Desafíos de la GC en el desarrollo de software en relación con el modelo de GC de Bustelo y Amarilla. Fuente: Elaboración propia.

- **Modelo de Mc. Elroy**

Este modelo fue creado por Mc. Elroy (2003), y se basa en buscar espacios vacíos de conocimiento para crear, a partir de este hallazgo, conocimiento sólido. En la figura 5 se puede observar un diagrama correspondiente a las fases que lo componen.

El modelo incluye dos actividades principales: producción del conocimiento e integración del conocimiento. La primera implica todas aquellas actividades necesarias para crear nuevo conocimiento y la segunda, aquellas actividades necesarias para integrar el conocimiento nuevo a las bases de conocimiento existentes en la organización. Se toman en cuenta los siguientes aspectos:

- La gente tiende a comprometerse con el aprendizaje como resultado de haberlo necesitado en su trabajo diario. Detectar estas necesidades conlleva a tomar acciones para generar conocimiento, y se conoce como “solicitud de conocimiento”. Estas solicitudes incluyen las posibles acciones necesarias para generar conocimiento.
- Mientras los individuos se involucran en la creación de solicitudes, a menudo atraen a otros, formando grupos en donde se comparte el conocimiento de manera informal, y se validan las solicitudes. Se requiere validación a nivel de la organización.
- No todas las solicitudes individuales o grupales son exitosas a nivel de la organización. Las exitosas son llamadas “solicitudes sobrevivientes” y las no exitosas pueden ser “solicitudes no definidas” o “solicitudes falsificadas”.
- Como las solicitudes son validadas en diferentes niveles de la escala organizacional, los gerentes y otros miembros de la organización, intentan compartir este conocimiento generado, esto es llamado “integración”.
- La combinación de conocimiento subjetivo con conocimiento objetivo, forman la base de conocimiento distribuida en la organización.
- El ciclo de vida del conocimiento comienza con la detección de problemas, y termina con conocimiento validado en la base de conocimiento.

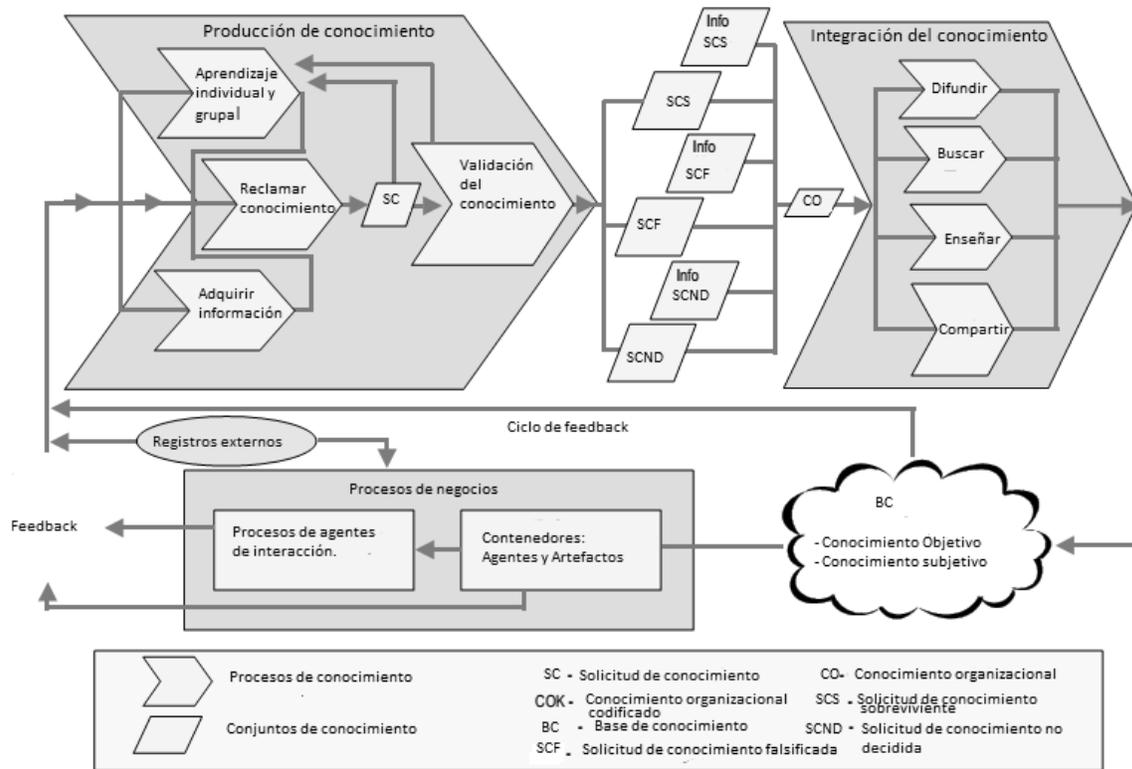


Figura 5: Modelo de Mc Elroy. Fuente: Mc Elroy (2003). P. 3. Traducción propia.

En la siguiente tabla (ver tabla 14), se muestran los desafíos de la GC en SF e ingeniería de software en relación con el modelo de Mc. Elroy.

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	SI	Si el miembro de la organización detecta un cambio en una tecnología puede hacer una solicitud para generar el nuevo conocimiento.
Necesidad de buscar conocimiento fuera de la organización	PARCIAL	Una solicitud de conocimiento podría incluir obtener conocimiento externo a la organización, sin embargo, no está especificado.
Contextos y procesos difieren en cada proyecto	NO	
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	NO	
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	PARCIAL	Podrían manejarse mediante solicitudes grupales que manejan conocimiento informal, sin embargo, los tiempos acortados en las SF hacen que un proceso de validación a nivel organizacional para formalizar sea complicado.
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	NO	
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	NO	
Conocimiento altamente especializado, difícil de formalizar.	NO	
Alta rotación del personal vuelve al conocimiento volátil.	NO	
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	NO	
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	NO	
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	NO	
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	NO	
Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	NO	
Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	NO	

Tabla 14. Desafíos de la GC en el desarrollo de software en relación con el modelo de GC de Mc. Elroy. Fuente: Elaboración propia.

En cuanto a los criterios de clasificación de modelos de GC de Gómez (2006) explicados anteriormente, se puede categorizar este modelo bajo el apartado almacenamiento, acceso y transferencia de conocimiento.

- **Modelo CEN**

En el año 2004 el CEN (Comité Europeo de Estandarización) publicó un modelo de GC llamado “Guía de buenas prácticas en gestión del conocimiento” (CEN 2004). A pesar de ser llamado guía de buenas prácticas, por su estructura se puede considerar como un modelo. Según Pawlowski y Bick (2015) este es uno de los modelos más utilizados actualmente. Se puede apreciar el esquema general de este modelo en la figura 6. Consta de tres capas principales:

- **Foco de negocios:** debe estar en el centro de cualquier iniciativa de GC y representa los procesos de la organización, los cuales son el contexto organizacional en donde el conocimiento crítico es creado y aplicado.
- **Actividades centrales:** se identifican 5 actividades centrales relacionadas con el conocimiento: identificar, crear, almacenar, compartir y utilizar. Estas actividades conforman un proceso integrado en este modelo y se realizan para apoyar los procesos de la organización.

A continuación, se detallan un poco más estas actividades en relación con el presente modelo:

1. **Identificar:** la organización debe estar consciente de cuál es su meta y qué conocimiento necesita para alcanzarla. Este proceso incluye determinar qué conocimiento ya existe y qué conocimiento necesita ser creado. Es necesario realizarlo siempre para garantizar la reutilización del conocimiento existente y reducir la redundancia.
2. **Crear:** a nivel individual y grupal propone actividades de interacción social como, por ejemplo: entrenamientos, aprender haciendo, tormentas de ideas. A nivel organizacional propone procesos de investigación y desarrollo a través de la creación de grupos de

- expertos como comunidades de práctica, la incorporación a la organización de expertos o la compra de otras organizaciones.
3. Almacenar: el modelo propone almacenar el conocimiento implícito, en costumbres de la organización y cultura organizacional, de esta forma, mientras estas personas estén disponibles, el conocimiento estará “memorizado” en la organización y disponible para su reutilización. Para el conocimiento explícito propone actividades como: seleccionar, categorizar y actualizar el conocimiento.
 4. Compartir: la meta de esta actividad es transferir el conocimiento a la persona adecuada, en el momento adecuado y con la calidad adecuada. Indican que el conocimiento puede ser transferido a través de bases de datos y documentos, o bien de persona a persona. Los métodos que permiten compartir el conocimiento en este modelo son: intranets, bases de datos, colaboración, entrenamientos, rotación de roles y seminarios.
 5. Usar: el conocimiento solo puede aportar valor si es usado, por lo cual esta actividad se centra en garantizar que todo el esfuerzo de las actividades anteriores va a tener frutos. En esta etapa se puede validar el conocimiento y determinar nuevas necesidades.
- **Facilitadores:** se dividen en dos categorías principales que son: capacidades personales de conocimiento y capacidades organizacionales de conocimiento. La primera incluye capacidades o aptitudes como: ambición, habilidades, comportamiento, experiencia, herramientas y distribución del tiempo. Estas deben ser desarrolladas a nivel personal y grupal. La segunda incluye aquellas capacidades que tienen que ser establecidas por los líderes de la organización para facilitar una efectiva gestión del conocimiento tanto interna (gerentes y empleados), como externa (clientes y proveedores).



Figura 6: Modelo CEN. Fuente: Transcripción y traducción propia a partir de CEN (2004)

Según los criterios de clasificación de modelos de GC de Gómez (2006) se puede categorizar este modelo bajo el apartado almacenamiento, acceso y transferencia de conocimiento por las actividades que propone; bajo el apartado sociocultural, pues promueve una cultura organizacional en pro de la GC y bajo el apartado tecnológico, pues promueve el uso de las tecnologías de información para soportar los procesos de GC.

En la siguiente tabla se muestran los desafíos de la GC en SF e ingeniería de software en relación con el modelo CEN.

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	SI	Habla de que el conocimiento debe ser actualizado constantemente.
Necesidad de buscar conocimiento fuera de la organización	PARCIAL	Habla de obtener conocimiento proveniente de proveedores y clientes, pero no del mundo exterior en general.
Contextos y procesos difieren en cada proyecto	SI	Habla de que el conocimiento debe estar siempre contextualizado y se debe garantizar que se aplique en el contexto correcto.

Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	NO	
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	PARCIAL	Habla de compartir conocimiento implícito, sin embargo, no habla de una forma de convertir ese conocimiento en explícito, o de una forma de garantizar qué esté disponible para los demás miembros de la organización.
Es necesario saber “Quién sabe qué cosa” para reducir tiempos	NO	
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	PARCIAL	Habla de compartir conocimiento implícito sin necesidad de convertirlo en explícito, sin embargo, no habla de tiempos reducidos.
Conocimiento altamente especializado, difícil de formalizar.	NO	
Alta rotación del personal vuelve al conocimiento volátil.	NO	En este punto en particular falla este modelo, pues confía en que el conocimiento implícito quede memorizado en los miembros y costumbres de la organización, de la organización sin contar con que puede haber rotación alta de personal.
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	PARCIAL	Hace énfasis en la reutilización del conocimiento, sin embargo, no dice nada del conocimiento negativo (que cosas no se deben hacer)
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	NO	
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	SI	Propone rotación de roles como método para compartir conocimiento.
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	PARCIAL	Propone comunidades de práctica, estas comunidades pueden ayudar a aprender más rápido a las personas con menos conocimientos, sin embargo, lo toma desde el punto de vista del conocimiento organizacional y no del conocimiento individual o grupal.
Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	PARCIAL	La rotación de roles o comunidades de práctica puede ayudar a que el conocimiento sea más uniforme.
Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	SI	El modelo propone garantizar la reutilización del conocimiento existente.

Tabla 15. Desafíos de la GC en el desarrollo de software en relación con el modelo CEN. Fuente: Elaboración propia.

- Otros modelos generales

Además de los modelos ya analizados, existen modelos generales más actuales que están basados en los ya vistos y que no van a ser mencionados en profundidad ya que las diferencias con los anteriores no son significativas, y no toman tampoco en cuenta la mayoría de los desafíos de GC en SF descritos anteriormente.

Entre estos modelos se pueden destacar:

- **Modelo KMC (Knowledge Management Cycle):** fue creado por Evans y Otros (2013), se basa en los hallazgos de Heisig (2009), quien analizó 160 modelos. Los autores toman las principales actividades encontradas en la mencionada investigación y generan un híbrido con las fases: identificar, almacenar, compartir, usar, aprender y crear.
- **Modelo K-TSACA:** creado por Shongwe (2016), está basado en las investigaciones de Heisig (2009) y busca unificar modelos existentes, entre ellos: Wiig (1999), Mc Elroy (2003) y Evans y otros (2013). Las fases resultantes de su investigación fueron: Transferencia o compartir, almacenamiento, aplicación o uso, creación, adquisición. No es específico tampoco en cómo aplicar el modelo resultante en una organización y se puede observar que las actividades propuestas se encuentran en otros modelos.
- **Modelo de Capas:** creado por Guevara y otros (2017), a partir de modelos anteriores, entre ellos: Wiig (1999), Nonaka y Takeuchi (1995), las investigaciones de Davenport y Prusak (1998), y las investigaciones de Heisig (2009). Propone un modelo por capas, en donde las principales actividades son: adquirir, organizar, difundir, transferir, aplicar y mantener. Son actividades que se han visto en modelos anteriores, esta vez más específicas, pues por cada una propone tareas (al igual que el modelo de Wigg), sin embargo, no resuelve la mayoría de los desafíos de GC en SF mencionados anteriormente.

3.1.3. Modelos orientados a desarrollo de software

Además de los modelos analizados anteriormente, existen algunos modelos de GC que están directamente orientados al rubro de desarrollo de software. Los modelos de este tipo que han sido encontrados durante esta investigación se analizarán a continuación.

- ***Modelo de Bovea y García***

Bovea y García (2011) diseñaron un modelo de GC que toma como base la teoría de la complejidad, y lo aplica a lo que ellos consideran como un modelo complejo: el desarrollo de fábricas de software.

En referencia a la teoría de la complejidad, Cárdenas y Rivera (2004), indican lo siguiente:

En la actualidad la teoría de la complejidad busca dar razón del universo como un todo, más allá de la simple suma de sus partes, y de cómo sus componentes se unen para producir nuevas formas. [...] En las ciencias, la complejidad es el término utilizado para connotar una nueva forma de pensar sobre el comportamiento colectivo de muchas unidades básicas que interactúan entre sí. [...] (p. 133)

Este modelo propone etapas para el desarrollo de software y trata de explicar cómo debería fluir el conocimiento en cada una de estas etapas. No parece ser directamente un modelo de GC sino un modelo de funcionamiento de una fábrica de software en donde se toman en cuenta los flujos de conocimiento.

Las etapas que plantea el modelo son iterativas, es decir, se pueden repetir para adecuarse a la reutilización de paquetes de software y por ende del conocimiento, se presentan a continuación:

1. **Organización:** propone que para el desarrollo de cada paquete de software se organicen equipos de trabajo de acuerdo con las habilidades de cada persona, horas de trabajo y características de cada producto. Según los autores, esto genera que el conocimiento fluya en varias direcciones.

2. Interacción: propone la interacción no solo entre miembros del equipo, sino también con clientes y el mundo exterior para aumentar las habilidades de cada individuo a la hora de resolver problemas.
3. FeedBack: las interacciones generan una retroalimentación que apoya los procesos que se llevan a cabo.
4. Adaptación: gracias a la fase anterior, se genera un acoplamiento que se traduce en mejoras de los procesos y productos. Los autores indican que en esta etapa los conocimientos se adaptan de igual forma, permitiendo que se entienda el porqué del cambio y se puedan consultar bitácoras para mejorar la toma de decisiones.
5. Evolución: las etapas anteriores forjan con el tiempo la evolución de los productos y de los procesos de gestión del conocimiento, logrando así un mejoramiento continuo.

Estas etapas se repiten, por lo cual en cada iteración la evolución es mayor.

Como los paquetes que se desarrollan constituyen un todo, el modelo sugiere que una vez que cada modelo esté listo, se proceda a implementarlos con la metodología “MSF For Agile Software Development” de Microsoft.

Al final del planteo los autores argumentan lo siguiente:

El modelo obtenido muestra que la gestión del conocimiento se implementa en los sistemas complejos porque para llegar a ser complejos se debe obtener, clasificar, distribuir, transmitir y guardar información sacando el mejor provecho de la organización, interacción, feedback, adaptación y evolución. (p. 9)

Se puede observar que no existe un detalle claro de cómo debe ser manejado el conocimiento en este modelo ni pasos detallados para este proceso, habla simplemente de interacciones entre miembros del equipo. Además, está atado a una metodología de desarrollo de software en particular, cuando una SF es libre de utilizar las metodologías que le resulten más prácticas.

No existen tampoco pruebas en el documento de métodos de validación de este modelo.

En la tabla 16 se muestran los desafíos de la GC en la ingeniería de software y SF en relación con este modelo:

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	NO	
Necesidad de buscar conocimiento fuera de la organización	PARCIAL	Habla de interacciones con miembros externos a la organización, sin embargo, no habla de recopilar conocimiento a partir de otro tipo de fuentes externas.
Contextos y procesos difieren en cada proyecto	PARCIAL	No nombra este problema en particular, pero divide a los equipos por paquete de software, y agrupa a los miembros de los mismos según habilidades y características del mismo, esto da algo de contexto a cada equipo.
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	NO	
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	PARCIAL	Habla de interacciones constantes entre miembros del equipo y retroalimentación, este conocimiento podría compartirse en estas interacciones. Sin embargo, no garantiza que este tipo de conocimiento se comparta.
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	NO	
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	SI	Habla de interacciones constantes entre miembros del equipo y retroalimentación, este conocimiento podría compartirse en estas interacciones
Conocimiento altamente especializado, difícil de formalizar.	PARCIAL	. Habla de interacciones constantes entre miembros del equipo y retroalimentación, este conocimiento podría compartirse en estas interacciones, sin embargo, no habla de formalizar el mismo.
Alta rotación del personal vuelve al conocimiento volátil.	NO	
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	SI	Mediante las etapas de feedback se puede controlar esto. Además, habla del mantenimiento de bitácoras para mejorar la toma de decisiones.
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	NO	
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	PARCIAL	No dice exactamente cómo compartir conocimiento entre roles diferentes y hacer este entendible, sin embargo, mediante las interacciones que propone podría compartirse.
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	PARCIAL	No dice exactamente cómo compartir conocimiento entre roles diferentes y hacer este entendible, sin embargo, mediante las interacciones que propone podría compartirse.

Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	PARCIAL	No dice exactamente cómo compartir conocimiento entre roles diferentes y hacer este entendible, sin embargo, mediante las interacciones que propone podría compartirse.
Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	SI	Habla contantemente de la reutilización de paquetes y por ende del conocimiento, Propone bitácoras para mejorar la toma de decisiones.

Tabla 16. Desafíos de la GC en el desarrollo de software en relación con el modelo de Bovea y García. Elaboración propia.

Según los criterios de clasificación de modelos de GC de Gómez (2006) explicados anteriormente, se puede categorizar este modelo bajo el apartado tecnológico, pues está orientado al desarrollo software, sin embargo, no queda claro si propone o no el uso de tecnologías para el soporte a la GC, no se puede catalogar tampoco como cultural, pues no propone un cambio a nivel organizacional ni como almacenamiento, acceso y transferencia, pues no están especificados del todo estos procesos.

- **Modelo ISECO**

Pons y otros (2014) crearon un modelo de GC para mejorar el desarrollo de equipos de proyectos informáticos. En la figura 7 se observa el esquema general de esta propuesta.

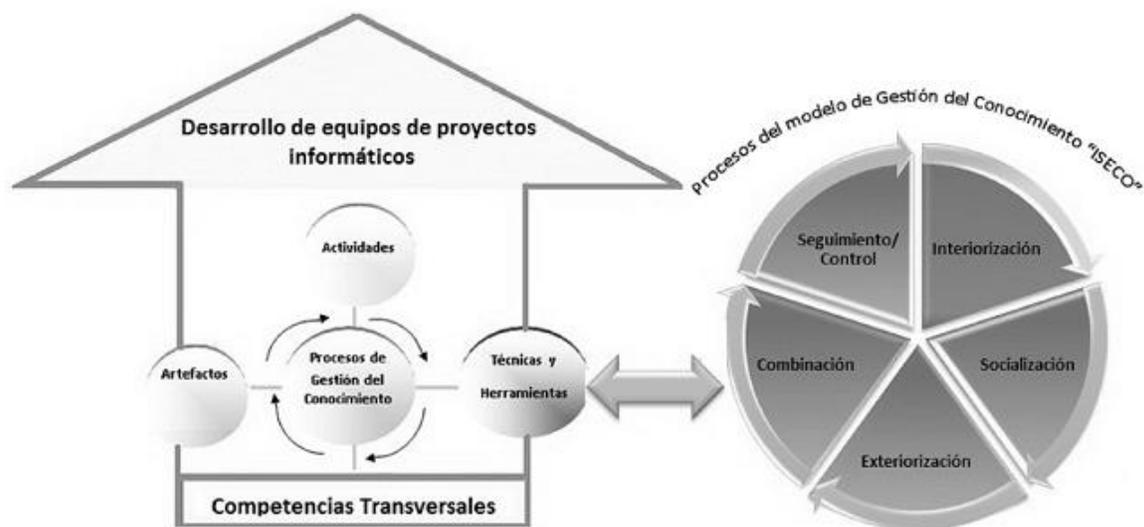


Figura 7: Modelo de Pons y otros. Fuente: Pons y otros (2014).

Las etapas que define este modelo son cíclicas, y se detallan a continuación:

1. **Interiorización:** se basa en convertir el conocimiento explícito en conocimiento implícito y agregarle experiencia. Propone como actividades: capturar, procesar y presentar el conocimiento implícito, y como herramientas de soporte: seminarios, talleres y mapas conceptuales. No explica detalladamente cómo se debe presentar el conocimiento tácito.
2. **Socialización:** se trata de compartir el conocimiento entre miembros del equipo, como herramientas propone actividades grupales como: planes de capacitación y talleres de discusión. También propone la tenencia de un sitio web de la organización.
3. **Exteriorización:** consiste en convertir el conocimiento de implícito a explícito. propone exteriorizar conocimiento a partir del intercambio o debate que se genera con la realización de técnicas como talleres, elaboración de guías para el auto aprendizaje y manuales de capacitación. Se sugiere como herramienta la utilización de mapas conceptuales.
4. **Combinación:** consiste en generar conocimiento explícito a partir de experiencias dentro del proyecto. Propone como herramientas: reuniones, mesas redondas, entrevistas y reuniones con expertos.
5. **Seguimiento y control:** es revisar el estado de aplicación del modelo para corregir lo que sea necesario.

Para soportar el modelo, propone la existencia de una colección de datos organizada por casos, la creación de una red de expertos donde se sepa quiénes son las personas con mayor conocimiento en áreas determinadas y el diseño de un sitio web para la difusión del conocimiento.

Se puede observar que este modelo es más específico en sus actividades que otros modelos anteriores.

En la siguiente tabla (ver tabla 17), se muestran los desafíos de la GC en SF e ingeniería de software en relación con el modelo ISECO.

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	NO	Aunque las actividades son cíclicas no especifica pasos para mantener actualizado el conocimiento.
Necesidad de buscar conocimiento fuera de la organización	NO	
Contextos y procesos difieren en cada proyecto	SI	Propone generar una colección de conocimiento por casos.
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	SI	Propone generar una colección de conocimiento por casos, y un sitio web donde esto se refleje, este sitio podría estar disponible para toda la organización.
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	PARCIAL	Habla de compartir conocimiento como "buenas prácticas", sin embargo, esto lo considera conocimiento explícito.
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	SI	Propone la creación de una red de expertos
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	NO	A pesar de que habla de convertir conocimiento implícito en explícito, no toma en cuenta los tiempos.
Conocimiento altamente especializado, difícil de formalizar.	SI	. En sus etapas se destaca el intento constante de convertir conocimiento implícito en explícito.
Alta rotación del personal vuelve al conocimiento volátil.	NO	
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	NO	A pesar de que propone organizar el conocimiento por casos, no habla de bitácoras o fracasos.
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	NO	
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	SI	Esto se puede lograr mediante las etapas y actividades que se proponen en este modelo.
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	SI	Mediante las actividades planteadas, se puede transmitir el conocimiento necesario a personas con menos experiencia.
Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	SI	Mediante la socialización se puede lograr solucionar este desafío.
Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	PARCIAL	No especifica nada de la reutilización, sin embargo, como propone generar conocimiento a partir de la experiencia y tener repositorios de código ordenados por caso, esto podría ser posible.

Tabla 17. Desafíos de la GC en el desarrollo de software en relación con el modelo ISECO. Fuente: Elaboración propia.

Según los criterios de clasificación de modelos de GC de Gómez (2006) explicados anteriormente, se puede categorizar este modelo bajo el apartado tecnológico, ya que propone el uso de herramientas tecnológicas como soporte al proceso de GC y bajo el apartado almacenamiento, acceso y transferencia, debido a que sus actividades se orientan a que este proceso ocurra.

3.1.4. Resultados del análisis

A lo largo de esta sección se analizan los modelos de GC más utilizados, así como también aquellos que están relacionados directamente con el desarrollo de software. Del análisis de estos 10 modelos, en relación con los desafíos de la GC en la ingeniería de software y las SF surge una matriz de vacancia que se puede observar en la tabla 18.

	No es tenido en cuenta.
	Si es tenido en cuenta.
	Es tenido en cuenta parcialmente.

Desafíos de la GC en la Ing. de software y en las SF.	Modelos de GC analizados.									
	Wiig	Nonaka/ Takeuchi	Sveiby	Earl	Kerschberg	Bustelo y Amarilla	Mc. Elroy	CEN	Bovea y García	ISECO
Cambios constantes en las tecnologías utilizadas.										
Necesidad de buscar conocimiento fuera de la organización										
Contextos y procesos difieren en cada proyecto										
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.										
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).										
Es necesario saber "Quién sabe qué cosa" para reducir tiempos										

Modelo de GC.	Puntajeobtenido.
Earl	18/30
ISECO	16/30
CEN	14/30
Bovea y García	13/30
Wiig	5/30
Sveiby	5/30
Kerschberg	5/30
Nonaka / Takeuchi	4/30
Mc. Elroy	4/30
Bustelo y Amarilla	2/30

Tabla 19: Calificación de los modelos analizados en relación con la matriz de vacancia. Fuente: Elaboración propia.

Es importante destacar que esta calificación es a modo de comparación entre los modelos analizados, para acercarse a determinar cuáles de estos podrían ser más exitosos si se aplican en una SF. En ningún momento hacen referencia a la calidad de dichos modelos, puesto que son estudios científicos que han sido validados previamente en otros rubros.

Se puede observar entonces, que el modelo con mayor puntuación es el modelo de Earl (18/30), seguido del modelo ISECO (16/30) y CEN (14/30) en respectivo orden. Sin embargo, el modelo de Earl fue analizado teniendo en cuenta todas las escuelas que plantea.

Con base en este análisis se puede concluir que la aplicación de cualquiera de los modelos de GC vistos, en la ingeniería de software y en especial en una SF, no resolvería todos los desafíos que implica la GC en este rubro, por lo cual al aplicar cualquiera de los mismos en una organización de este tipo no se podrían obtener las ventajas competitivas que ofrece el conocimiento.

Otro factor importante por destacar es que la alta rotación del personal existente en una SF no es tomada en cuenta por ninguno de los modelos analizados, así como también la distribución global de

un proyecto, siendo este desafío tomado en cuenta parcialmente por tan solo un modelo (Nonaka y Takeuchi).

Se propone entonces, para solucionar el problema, la elaboración y validación de un modelo de GC que tome en cuenta los desafíos antes identificados, para ser de esta manera aplicable en una SF, permitiéndole a la misma obtener las ventajas competitivas que derivan de dicha gestión

4. SOLUCIÓN

En este capítulo se propone un modelo de GC aplicable en SF que tiene en cuenta los desafíos de GC mencionados a lo largo de esta investigación. Para ello se presenta la estructura del modelo propuesto detallando sus fases, actividades y documentación necesaria (sección 4.1), un ejemplo de la aplicación de este (sección 4.2), y la realización del mismo análisis comparativo en relación con los desafíos de GC que fue aplicado para todos los modelos analizados durante esta investigación (sección 4.3).

4.1. Modelo propuesto

4.1.1. Generalidades

Este modelo tiene como base actividades y/o etapas propuestas en los modelos de EARL, Mc. Elroy ISECO y CEN, que se irán nombrando a medida que se explique la propuesta. Está diseñado para ser aplicado en SF teniendo en cuenta los desafíos de GC que existen en este rubro y han sido detallados durante capítulos anteriores.

Consta de 6 fases cíclicas que aplicarán los miembros de cada proyecto, más dos fases apartes que deben ser llevadas a cabo por personas designadas por la organización y serán explicadas más adelante.

El modelo está pensado para ser compatible con metodologías de desarrollo tanto derivadas del manifiesto ágil (ya que las reuniones que propone pueden ser una prolongación de determinadas reuniones o ceremonias que forman parte de este tipo de metodologías) como tradicionales.

La idea es que sea un modelo cíclico paralelo al proceso de desarrollo de software (al igual que en el modelo ISECO), sin interferir de manera agresiva en las actividades normales del equipo y la organización, es decir, sin consumir mucho tiempo y esfuerzo de los recursos.

Al igual que el modelo de Mc. Elroy, este modelo se centra en buscar espacios vacíos de conocimiento para crear, a partir de este hallazgo, conocimiento sólido, acompañando el

trabajo diario de los individuos. Se utiliza en parte este modelo como base, porque es la forma más fácil de adaptar un modelo a los tiempos reducidos que existen en una SF y generar conocimiento realmente importante para el trabajo de la organización. Sin embargo, de este modelo es tomada la idea principal solamente, puesto que está formado por muchas etapas, actividades y validaciones que podrían complicar el trabajo de los individuos de una SF.

Al igual que el modelo de Mc. Elroy, este modelo comienza con la detección de una necesidad de conocimiento y termina con el conocimiento formalizado en la base de conocimientos, sin embargo, no solo queda en la base de conocimientos, sino también interiorizado en otros miembros de la organización (esto reduce la volatilidad del conocimiento debido a la alta rotación de personal). Es importante mencionar que el modelo CEN también comienza con la identificación de la necesidad de conocimiento y la búsqueda de este evitando siempre la redundancia, sin embargo, no se centra en una necesidad en particular detectada por un individuo de la organización.

A diferencia de otros modelos como el modelo CEN, este modelo propone que el conocimiento sea compartido (al menos con el mismo equipo de trabajo) antes de formalizarlo. De esta manera el conocimiento puede ser validado por los demás miembros del equipo, ampliado y mejorado antes de ser llevado a la base de conocimientos.

4.1.2. Estructura del modelo

El modelo propuesto se divide en 6 fases principales que involucran a un equipo de desarrollo de software y se realizan de forma cíclica y continua. Estas fases son:

- Detectar una necesidad de conocimiento.
- Investigar (buscar el conocimiento).
- Organizar e interiorizar.
- Utilizar
- Compartir y validar.
- Formalizar

Además, existen otras dos fases que son paralelas a las ya mencionadas y son llevadas a cabo por personas designadas especialmente para esto por la organización, estas fases son:

- Fase de preparación de la organización para la aplicación del modelo.
- Fase de mantenimiento del modelo.

La siguiente figura (ver figura 8) ilustra la organización de las fases del modelo propuesto.

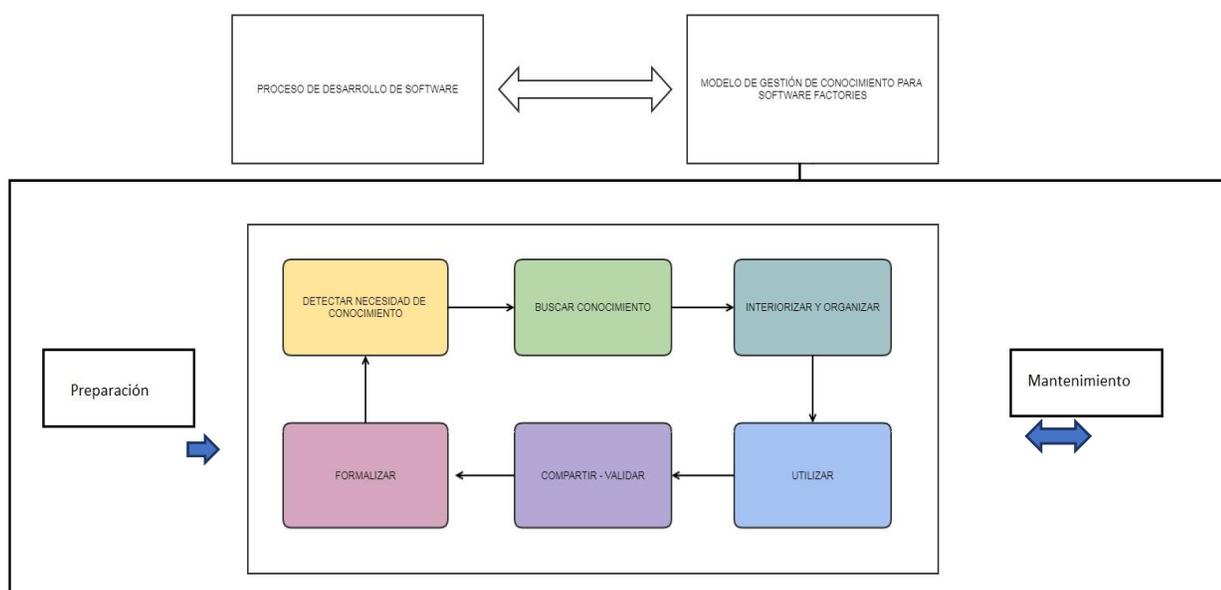


Figura 8. Fases del modelo propuesto

A continuación, se explican las fases tanto cíclicas como paralelas que integran el modelo propuesto.

4.1.3. Fases paralelas

Como se explicó anteriormente, en estas fases los protagonistas no son los integrantes de cada proyecto, sino personas designadas por la organización para llevarlas a cabo.

1- Fase de preparación de la organización para la aplicación del modelo.

Durante esta fase se designa el comité de GC de la organización, se determina qué elementos conforman la base de conocimientos y se busca ponerlos a disposición de todos los miembros de la organización, se diseña y crea la wiki de conocimiento y el directorio. Las actividades que componen esta fase son:

a. Seleccionar comité de Gestión del conocimiento:

La organización debe seleccionar a una o más personas (dependiendo de las necesidades de cada caso), para integrar este comité. Las funciones de este son:

- Supervisar la adecuada aplicación del modelo en cada uno de los proyectos.
- Llevar control de las charlas informativas (cronograma y temas) así como también evaluar las solicitudes de estas.
- Garantizar el espacio y materiales necesarios para que se lleven a cabo las charlas informativas.
- Tomar cualquier decisión pertinente a adaptar el presente modelo a la organización.
- Capacitar a los miembros de la organización en cuanto a la aplicación del presente modelo.
- Velar por el correcto funcionamiento de la wiki de conocimientos y el directorio
- Solventar en lo posible, cualquier duda y o problema que tenga algún equipo en relación con la gestión del conocimiento.
- Gestionar cualquier motivación que decida otorgar la organización a los empleados que generen más conocimiento.

Queda a criterio de la organización seleccionar el número de personas que integrarán este espacio y a las personas más aptas. Se recomienda que sean al menos 3 personas y siempre en número par. Así mismo no hace falta que se creen nuevos puestos exclusivos dentro de la organización para realizar estas tareas.

Roles: directivos de la organización, Comité de GC

Documentos: ninguno

Ejemplo: los directivos de la organización han decidido implementar este modelo de GC en su SF, para ellos se reúnen y eligen al comité de GC, el cual estará integrado por 3 de los Project Manager de la organización.

- b. Determinar qué elementos pertenecen a la base de conocimientos y garantizar una forma en la que los miembros de la organización tengan acceso a este material.

Roles: Comité de GC

Documentos: ninguno

Ejemplo: el Comité de GC investiga y realiza un inventario informal sobre los repositorios de conocimiento que existen actualmente en la organización. Deciden crear un repositorio de documentos virtuales, crear el directorio, listas de email y wiki del conocimiento, y deciden que estas serán las fuentes que integren la base de conocimientos.

- c. Crear o adquirir el directorio y/o las listas de correo electrónico para cumplir esta función.

Roles: Comité de GC

Documentos: ninguno

Ejemplo: el Comité de GC decide que utilizará un equipo de desarrollo para crear una herramienta electrónica que sirva como directorio, así mismo decide crear una lista de correo que servirá para consultar dudas entre los miembros de la organización.

- d. Crear o adquirir una herramienta que pueda ser utilizada como wiki del conocimiento. Si ya existe una, realizarle los cambios necesarios para adaptarla al modelo y a la organización.

Roles: Comité de GC

Documentos: ninguno

Ejemplo: el Comité de GC decide utilizará el mismo equipo de desarrollo que elaborará el directorio para adaptar una herramienta de GC que ya existe a la empresa y así aplicarla con el presente modelo. Tienen sucesivas reuniones con el equipo de desarrollo seleccionado para definir los requerimientos y manejan este caso como un proyecto más de la empresa.

- e. Capacitar a los equipos existentes en cuanto a la aplicación del modelo.

Roles: Comité de GC, miembros de la organización.

Documentos: Ninguno

Ejemplo: una vez listo el directorio y la wiki de conocimientos, el comité de GC elabora una serie de charlas informativas con los líderes técnicos y Project managers en la organización para capacitarlos en cuanto al modelo y las herramientas elaboradas, estos líderes deben a su vez capacitar a sus equipos.

La siguiente figura ilustra las actividades presentes en esta etapa (ver figura 9).

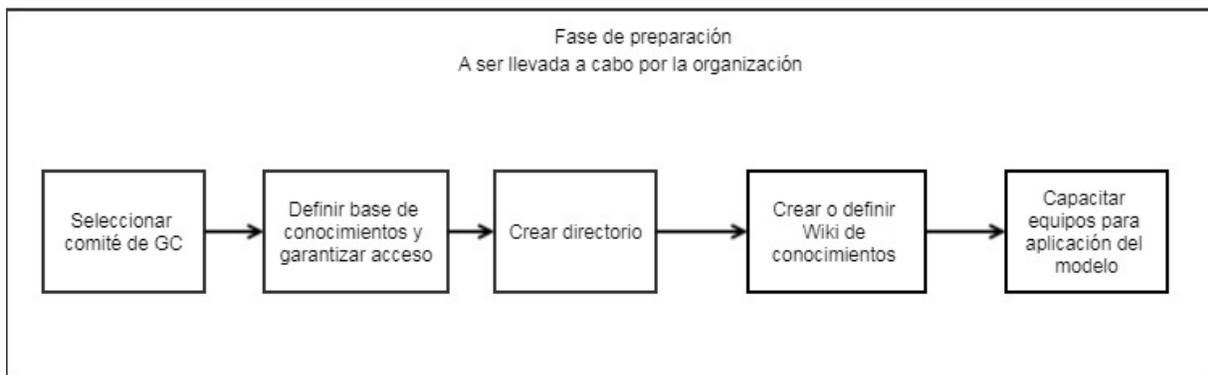


Figura 9. Actividades de la fase de preparación

Al igual que en el modelo de Nonaka y Takeuchi (1994) se recomienda que la organización apoye en todos los niveles el proceso de gestión del conocimiento, tomándolo como una de sus prioridades para que de esta manera se puedan obtener las ventajas que ofrece la GC.

Existen aspectos importantes para tener en cuenta durante esta fase, que son expuestos a continuación:

Al igual que la escuela de Sistemas de EARL, se debe contar con una base de conocimientos estructurada, y se pueden dar incentivos para generar conocimiento válido, de esta forma los miembros de la organización podrían comprometerse más con el proceso de GC.

Al igual que la escuela cartográfica del modelo de EARL, se debe tener un directorio específico de los miembros de la organización y el conocimiento que poseen, esto va a ser mucho más fácil de mantener actualizado, si se vincula directamente de forma automatizada con la base de conocimientos.

Se puede destacar, que el conocimiento en este modelo presenta un flujo, al igual que la escuela de ingeniería de Earl, en donde el conocimiento es compartido en una base de conocimientos de fácil acceso a todos los miembros de la organización, así mismo se busca que los miembros estén conectados por las formas de distribuir el conocimiento y colaboración entre los mismos para validar y compartir dicho recurso, similar a la escuela de redes de Earl.

- **Características de la base de conocimiento:**

En este modelo se considera como base de conocimientos, aquellos medios internos de los cuales se puede obtener conocimiento, como lo son: La wiki de conocimientos, correos electrónicos, intranet o web y documentos internos. Esta base de conocimientos es similar a la presentada en el modelo de Kerschberg, y se representa en la siguiente figura (ver figura 10):

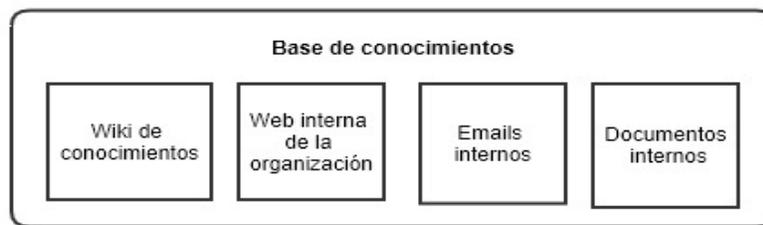


Figura 10. Elementos que conforman la base de conocimientos

La wiki de conocimientos es un sistema interno (web, aplicación de escritorio o aplicación móvil), en donde se centralizará en conocimiento que va siendo formalizado, en forma de artículos o tutoriales. Puede formar parte de la web interna de la organización.

Para que algunos de los desafíos planteados anteriormente en esta investigación sean resueltos por este modelo, es muy importante la wiki de conocimientos y se sugiera que tenga ciertas características que se presentan a continuación:

- a. Debe haber un espacio para cada proyecto, accesible al menos a los líderes de proyectos, en donde se detallan características propias del mismo, contexto, problemas importantes que se hayan presentado a lo largo de su duración y la forma en que fueron resueltos. De igual manera, al final del proyecto se debe colocar un apartado con recomendaciones para otros equipos y si el proyecto fue fallido, se deben indicar las posibles causas y cosas que no se deben hacer. Se puede determinar qué información estará visible para toda la organización según las necesidades de esta.
- b. Cada entrada en la wiki de conocimientos (cada artículo o pieza de conocimiento agregada) debe contener palabras claves que facilitarán la búsqueda, así mismo contener lo siguiente:
 - a. Autor o Autores.
 - b. Proyecto y fecha.
 - c. Tecnología(s).
 - d. Fuente (bien sea interna o externa).

- c. Se recomienda la utilización de etiquetas para cada artículo, de forma de clasificar de mejor manera la información. Por ejemplo, si es información sobre “Mejores Prácticas” se puede utilizar una etiqueta similar a #MejoresPrácticas, si la tecnología es un artículo sobre programación en c++ se puede utilizar una etiqueta como #c++ y así sucesivamente, esto va a ayudar a separar de mejor manera la información y facilitará la búsqueda de la misma.
- d. Se recomienda que los filtros de búsqueda sean al menos por texto, por autor, por proyecto, y por etiqueta.
- e. Se recomienda mostrar en la página de inicio los títulos de los últimos artículos agregados con su respectivo enlace.
- f. Se recomienda colocar al final de cada artículo una lista de otros artículos creados por el mismo autor y artículos con etiquetas similares.
- g. Una vez que un artículo es creado no estará inmediatamente visible, debe ser revisado y aceptado por el líder del proyecto primero.
- h. Se recomienda que la wiki de conocimientos esté vinculada con el directorio, es decir, en algún lugar del artículo debe haber un link que lleve al perfil del autor en el directorio.

- **Características del directorio:**

El directorio es otra herramienta que puede facilitar el proceso de compartir conocimiento dentro de la organización. Un directorio contiene información básica de todos los miembros de la organización y puede permitir que un miembro de esta consiga a otros que tengan información relacionada con sus dudas y pueda contactarlos para solventarlas. Se recomienda que el directorio sea un medio electrónico, por ejemplo, parte de la web interna de la organización. Se recomiendan las siguientes características para el directorio:

- a. La utilización de etiquetas en el directorio que reflejen el conocimiento de cada persona, esta información puede ser actualizada periódicamente por cada uno.
- b. Debe contener información básica de cada miembro de la organización: Sus conocimientos relacionados con la organización y su trabajo diario, su rol actual, proyectos en los cuáles ha

estado involucrado, información necesaria para contactarlo y una lista de los artículos que ha creado para la base de conocimientos con sus respectivos enlaces.

- c. Cuando una persona realiza una búsqueda en la base de conocimientos, debe aparecer un listado de las personas que podrían saber del tema.
- d. También se deben poder realizar búsquedas en el directorio, por etiquetas, proyectos o datos de los miembros de la organización.

Tanto la wiki de conocimientos como el directorio deben estar accesibles para miembros de la organización en todas las locaciones geográficas donde la misma opere, de esta manera y ubicando las reuniones relacionadas con la aplicación de este modelo en horarios donde todos los miembros del equipo estén disponibles (al igual que se suele hacer cuando se aplica scrum en proyectos distribuidos), se hace factible su aplicación en equipos globalmente distribuidos.

Es entendible que algunas organizaciones no lleven control de en qué proyectos está cada persona o un historial de este tipo de información, por lo que las características del directorio son recomendaciones, sin embargo, para que funcione adecuadamente, el directorio debe contener como mínimo el rol y tecnologías en las cuáles trabaja y ha trabajado cada miembro.

De no existir un directorio y no haber posibilidades de conseguirlo, se recomienda entonces para sustituirlo, listas de email organizadas a las que pertenezcan los miembros de la organización, a las cuales la persona que tiene una necesidad de conocimiento pueda enviar un email preguntando por el mismo.

2- Fase de mantenimiento del modelo.

El modelo debe llevar un mantenimiento y control periódico para que su aplicación pueda ser prolongada en el tiempo. Este mantenimiento debe ser llevado por la organización, la cual debe designar a las personas necesarias para cada una de las actividades que conlleva esta fase. La periodicidad de cada una de las actividades de esta fase debe ser definida por la

organización según sus necesidades. Las actividades de esta fase no dependen una de otra, por lo que pueden ser llevadas a cabo en cualquier orden:

- a. Back up de la wiki de conocimientos y el directorio: debe ser llevado a cabo un *back-up* periódico de la información almacenada en la wiki de conocimientos y el directorio, la periodicidad de esta actividad y las personas involucradas dependerá de las políticas de back-up de la organización.

Roles: depende de las políticas de la organización.

Documentos: depende de las políticas de la organización.

Ejemplo: el equipo responsable de la preservación y seguridad de los datos en la organización deberá hacer back up periódicamente de las bases de datos relacionadas con la base de conocimientos.

- b. Solicitud de feedback e implementación de mejoras: esta actividad se recomienda que la lleve a cabo el comité de GC designado por la organización y consiste en solicitar feedback periódicamente a los miembros de la organización relacionado con el proceso de GC, para determinar qué aspectos hay que reforzar o modificar y hacer que el modelo se adecúe cada vez más a las necesidades de la organización.

Roles: Comité de GC

Documentos: depende de cómo decida el comité de GC manejar el feedback.

Ejemplo: el Comité de GC decide realizar una encuesta al final de cada cuatrimestre a los miembros de la organización, para ello utilizan una herramienta virtual de encuestas en donde colocan varias preguntas y pasan el enlace a los miembros de la organización para que las llenen. Una vez obtenidos los resultados se reúnen para analizarlos y tomar decisiones al respecto.

- c. Reconocimiento a los generadores de conocimiento: como se nombró anteriormente, se recomienda que la aplicación del modelo sea incentivada con premios o reconocimientos a las personas que más artículos suban a la wiki de conocimientos en un período de tiempo

determinado por la organización, el tipo de reconocimiento y periodicidad de este deben ser definidos previamente.

Roles: Comité de GC

Documentos: ninguno

Ejemplo: el Comité de GC decide que al finalizar cada cuatrimestre entregarán un premio a los 3 miembros de la organización que más artículos hayan creado en la wiki de conocimiento, para ello realizarán un evento chico y notificarán la fecha por email.

La siguiente figura (figura 11) muestra de forma gráfica las actividades a ser llevadas a cabo en esta fase.

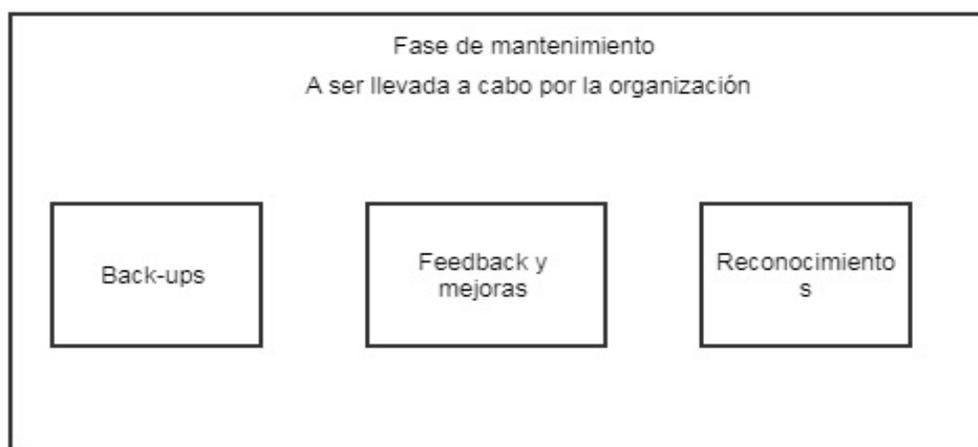


Figura 11. Actividades de la fase de mantenimiento del modelo.

4.1.4. Fases cíclicas

Estas fases son llevadas a cabo por los miembros de la organización y todo el proceso se desencadena cuando un individuo detecta una necesidad importante de conocimiento. Cada fase

incluye un conjunto de actividades ideadas para generar el menor impacto posible en cuanto a tiempo y esfuerzo del proceso de desarrollo de software.

Durante la explicación del modelo, se detallará cada actividad, los roles involucrados en la misma y un ejemplo relacionado. Al final de este capítulo se encuentra un ejemplo completo del modelo, de inicio a fin, para dar una visión más general del mismo.

1. Detectar necesidad de conocimiento

Esta etapa inicia el ciclo del modelo, se da cuando una persona del equipo detecta que existe una necesidad de conocimiento importante que puede ser un bloqueo para realizar su trabajo o para el trabajo de otros compañeros del equipo.

Se puede afirmar que a partir de esta etapa el modelo propuesto presenta similitudes con los procesos de mejora continua dentro de la gestión de calidad en las organizaciones.

Bhuiyan&Amit (2005) definen proceso de mejora continua, como una cultura de mejora sostenida que apunta a la eliminación del desperdicio en todos los sistemas y procesos de una organización. Involucra el trabajo de todos los miembros de la organización para hacer mejoras sin necesidad de grandes inversiones de capital.

Para los mencionados autores, el proceso de mejora continua puede ocurrir por cambios evolutivos, en cuyo caso las mejoras son incrementales. Se puede observar a lo largo de este modelo, que el ir sumando conocimiento a la organización mediante la detección de una necesidad de este, de forma cíclica, es similar a lo que el proceso de mejora continua supone.

Las actividades que involucran esta etapa serían:

a. Detectar necesidad

Es el momento en el que un miembro de un proyecto se da cuenta de que necesita utilizar un conocimiento que no posee.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: ninguno.

Ejemplo: la persona J es un desarrollador que pertenece a un proyecto de la organización, durante el desarrollo de sus actividades detecta que una parte de la aplicación en la que está trabajando está muy lenta debido a una errónea aplicación de patrones de arquitectura y necesita saber cuál es la mejor forma de diseñar una nueva arquitectura para cambiarla en un futuro.

b. Preguntas para determinar importancia:

Para determinar si realmente existe una necesidad de conocimiento el miembro del equipo debe responderse las siguientes preguntas:

- ¿Es esta necesidad de conocimiento un bloqueo para terminar mis tareas?
- ¿Puede ser esta necesidad de conocimiento un bloqueo para que otro miembro de mi equipo o de otro equipo termine sus tareas?
- ¿Puede surgir esta misma necesidad de conocimiento en un futuro, bien sea por alguien en mi equipo o por otra persona de la organización?
- ¿Adquirir este conocimiento puede ayudar a mejorar la calidad y legibilidad del resultado esperado por el equipo y el cliente?

Si la respuesta a alguna de estas preguntas es SI, entonces es una señal de que se debe seguir con las siguientes etapas. Si la respuesta es No a las cuatro preguntas la persona debe replantearse la importancia de esta necesidad de conocimiento para evitar perder tiempo y enfocarse en la resolución de sus tareas.

Responder la cuarta pregunta de la segunda actividad es necesario para determinar la necesidad de conocimiento de convenciones locales de desarrollo de software dentro del equipo. Es importante acotar también que, si un líder de proyecto detecta errores comunes y repetitivos en el proceso de desarrollo de software, esto también puede ser un detonante para generar conocimiento, ya que claramente, a pesar de que esta persona cuente con el mismo, el resto del equipo necesita tener a mano esta información en pro de lograr los objetivos deseados con los estándares de calidad deseados.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: ninguno

Ejemplo: luego de la actividad anterior J procede a hacerse las preguntas respectivas a esta actividad. Resultando que la respuesta a las preguntas 3 y 4 es un SI, por lo cual J debe completar los demás pasos del modelo.

Esta etapa fue tomada de la primera actividad del modelo de Mc. Elroy, como detonador del ciclo de gestión del conocimiento.

En la siguiente figura (ver figura 12) se puede entonces observar la estructura de esta fase del modelo:

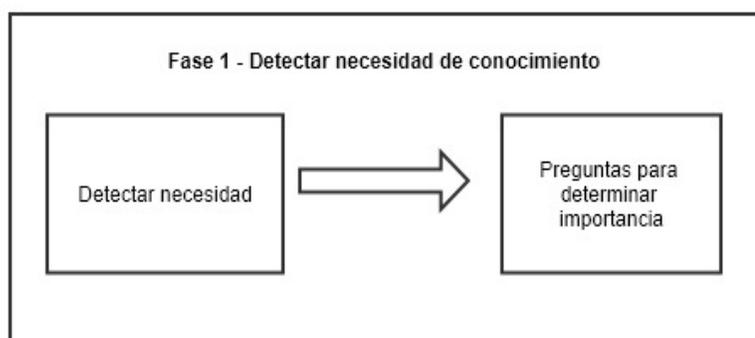


Figura 12. Actividades de la fase 1. Detectar necesidad de conocimiento

2. Buscar el conocimiento

La persona debe recurrir a las fuentes internas de conocimiento pertenecientes a la organización (base de conocimientos y directorio en primera instancia, descritos a lo largo de este capítulo) para buscar el conocimiento necesitado, si no lo consigue de esta manera, entonces debe recurrir a fuentes externas.

La idea de buscar el conocimiento en fuentes internas en primera instancia no es solo para ahorrar tiempo, sino que también de esta manera se puede reutilizar el conocimiento existente y evitar la redundancia. Estos recursos como la wiki de conocimiento (sitio donde se centralizará el conocimiento) y el directorio son tomados de las escuelas de sistemas y cartográfica del modelo de Earl. Sin embargo, éste utiliza el término Base de conocimientos para referirse al sitio donde se centraliza el mismo.

Las actividades que involucra esta fase son:

a. Buscar el conocimiento en la wiki de conocimientos

Si el conocimiento fue encontrado en la wiki de conocimientos (descrita más adelante), y fue de utilidad entonces no se llevan a cabo las últimas dos fases del modelo, ya el conocimiento está formalizado. Si no fue encontrado o no se encontró nada útil se debe continuar con la siguiente actividad de esta fase.

Roles: Todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: Ninguno

Ejemplo: La persona J procede entonces a buscar el conocimiento que necesita en la wiki de conocimientos, buscando mediante palabras claves.

b. Buscar el conocimiento en el directorio

Si no se consiguió nada útil en la wiki de conocimientos, se debe proceder a buscar una persona con conocimientos del tema en el directorio.

Si el conocimiento fue encontrado utilizando el directorio (es decir, en otra persona de la organización), pero no está formalizado, se debe considerar entre ambos la importancia de formalizar este conocimiento respondiendo las mismas preguntas de la etapa anterior. Si

es importante formalizarlo entonces se debe saltar a con la actividad e de este modelo, si no es importante, no se llevan a cabo las dos últimas fases del modelo.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: ninguno

Ejemplo: la persona J no consigue la información que necesita en la wiki de conocimientos, por lo que procede a buscar en el directorio, para ello va al directorio y busca de nuevo por palabras claves. Consigue a una persona que tiene conocimientos de arquitectura de software y procede a enviarle un email. Se reúne con esta persona para solventar sus dudas y ambos coinciden en la importancia de agregar este tema a la wiki de conocimientos.

c. Buscar el conocimiento en otras fuentes internas de la base de conocimientos

Si el conocimiento aún no ha sido encontrado en las actividades anteriores, se debe proceder a buscar en otro tipo de fuentes internas, como, por ejemplo: algún repositorio de documentos. Si se consigue la información, la persona debe preguntarse si es importante que esta información esté en la wiki de conocimientos y de ser así, saltar a la actividad e.

Roles: Todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: Ninguno

Ejemplo: A pesar de que J en la actividad anterior pudo juntarse con personas internas que sabían al respecto, no le ha quedado el tema del todo claro por lo que decide buscar en el repositorio de documentos de la organización.

d. Buscar el conocimiento en fuentes externas

Si el conocimiento aún no ha sido encontrado, es necesario entonces buscar en fuentes externas.

Si el conocimiento fue encontrado en una fuente externa a la organización, el empleado debe utilizar su criterio para determinar la importancia de formalizar este conocimiento dentro de la organización. Es decir, debe preguntarse: ¿Es este conocimiento difícil de encontrar en una fuente externa?, ¿Es importante realmente que la organización cuente con este conocimiento en la wiki de conocimientos?, ¿Cuándo otra persona tenga la misma necesidad de conocimiento, la puede resolver más rápido consultando fuentes internas? Si la respuesta a alguna de estas preguntas es SI, entonces se debe continuar con la siguiente actividad de esta fase, si la respuesta a todas las preguntas es NO, entonces no se continúa con las demás fases del modelo.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: ninguno

Ejemplo: J no consigue nada en el repositorio de documentos por lo que procede a buscar en internet. Consigue un artículo interesante que complementa la información que recibió por parte de los otros miembros de la organización. Se pregunta entonces si es necesario que esta información esté en la wiki de conocimientos y confirma su idea de que sí es necesario formalizarla.

e. Comunicar novedades al líder del proyecto

Si se determina que es necesario formalizar el conocimiento encontrado, entonces la persona debe enviar un email al líder del equipo para que este conocimiento sea considerado en las siguientes fases del modelo para el período actual, es decir, sea agregado a la lista de temas a compartir y evaluar en la reunión periódica de GC (ver sección 4.1.3 – e y f). Una vez que el líder del equipo recibe dicho email, si considera que es relevante según su criterio, debe agregarlo a la lista antes mencionada.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: solicitud para compartir/validar conocimiento (ver 4.1.3 e) y lista de temas a compartir/validar en reunión (ver 4.1.3 f)

Ejemplo: J ya decidió que es importante formalizar este tema, así que le envía un email a su líder de proyecto para que lo incluya en la lista de temas a discutir en la reunión de compartir/validar conocimiento (ver 4.1.3 e). Su líder técnico está de acuerdo y lo agrega a la lista de temas a discutir en la siguiente reunión (ver 4.1.3 f).

Es importante mencionar que el orden de estas actividades es sugerido, pero no determinante, queda a criterio del miembro del equipo en qué fuentes realizar su búsqueda primero para conseguir el conocimiento, pero se continúa siempre y cuando el mismo no esté formalizado y sea importante hacerlo.

La siguiente figura (ver figura 13) muestra las actividades resultantes para esta fase.

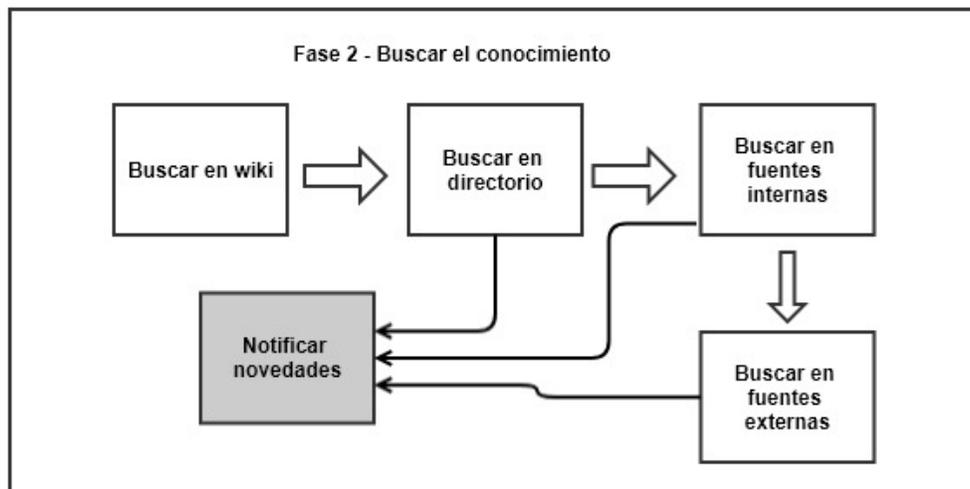


Figura 13. Fase 2. Buscar el conocimiento.

3. Interiorizar y organizar el conocimiento

La persona debe una vez encontrado el conocimiento, organizar la información adquirida e interiorizarla. Si el conocimiento no está formalizado en la wiki y tiene dudas de que este nuevo conocimiento esté correcto en primera instancia antes de continuar con su trabajo,

debe consultarlo con líderes y otros miembros de la organización (validación en primera instancia).

Esta etapa puede ser comparada con la etapa de Interiorización del modelo ISECO.

Las actividades propuestas para esta etapa son:

a. **Consultar con líderes**

En caso de que existan dudas sobre la veracidad de la información encontrada, la persona debe consultar con sus líderes más directos, los cuales, aunque no posean conocimiento del tema pueden validar las fuentes de donde fue adquirido. Si no existen dudas se debe saltar directamente a la siguiente actividad.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: ninguno

Ejemplo: J aún tiene dudas de si la información obtenida es correcta, por lo que lo discute con su líder técnico, quien verifica las fuentes de donde fue obtenido y valida las mismas.

b. **Organizar la información encontrada**

La persona debe organizar la información encontrada de modo que sea lo más fácil posible de interiorizar y convertir en conocimiento.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: ninguno

Ejemplo: J procede entonces a organizar la información, para ello utiliza un mapa conceptual ya que es la forma en que siempre le ha resultado más fácil comprender un tema.

c. **Interiorizar la información adquirida:**

La persona debe interiorizar la información para convertirla en conocimiento que pueda ser utilizado en la siguiente fase, esta actividad puede realizarse en forma simultánea a la anterior. Se recomienda conversar sobre el tema encontrado con otro miembro del equipo para facilitar el proceso de interiorización.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: ninguno

Ejemplo: mientras J realiza el mapa conceptual de la actividad anterior, va interiorizando la información.

La siguiente figura (ver figura 14) muestra las actividades que involucran esta fase.

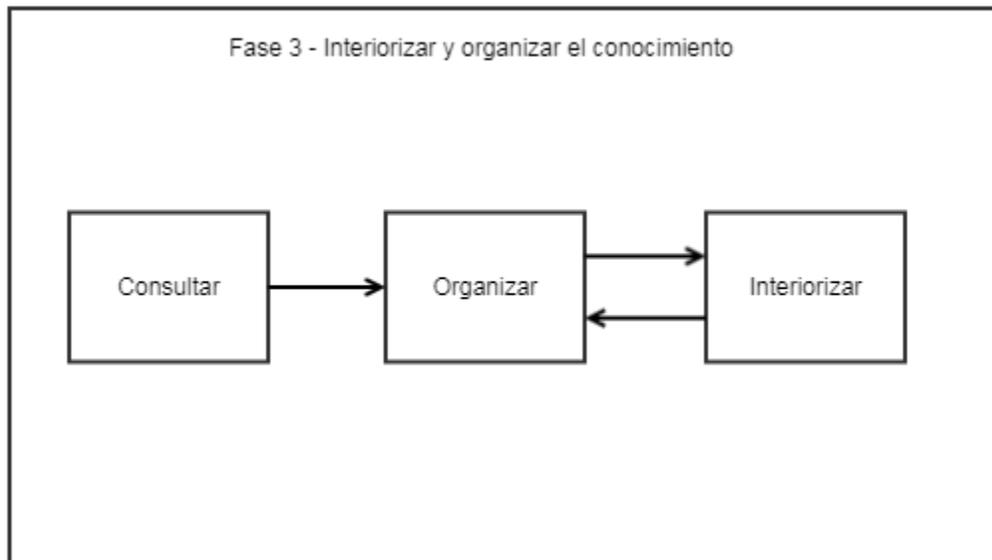


Figura 14. Actividades de la fase 3. Interiorizar y organizar el conocimiento.

4. Utilizar el conocimiento:

Una vez validado esto, debe utilizar entonces el conocimiento para seguir realizando sus tareas, tomando nota de los puntos más importantes de lo aprendido, para poder llevar a cabo las siguientes etapas de este modelo cuando sea el momento. Esta etapa se podría considerar una extensión de la etapa de interiorización del modelo ISECO, ya que la persona debe tomar registro de su experiencia utilizando este conocimiento para darle forma al mismo y tenerlo en cuenta a la hora de compartir y externalizar el conocimiento.

Las actividades que involucra esta fase serían:

a. Utilizar

Es la utilización del conocimiento como tal en las tareas diarias según el rol que tenga el individuo.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: ninguno

Ejemplo: J procede a rediseñar la nueva arquitectura según los conocimientos adquiridos.

b. Registrar

La persona debe tomar registro de cómo fue la experiencia de aplicar el conocimiento encontrado, para que al incluir esto en la etapa de formalización, éste resulte más útil para las personas que lo necesiten en un futuro.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: ninguno

Ejemplo: J va anotando ítems importantes sobre su experiencia al diseñar la nueva arquitectura.

Ambas actividades pueden realizarse en forma paralela.

La siguiente figura (ver figura 15) muestra las actividades que conforman esta etapa:

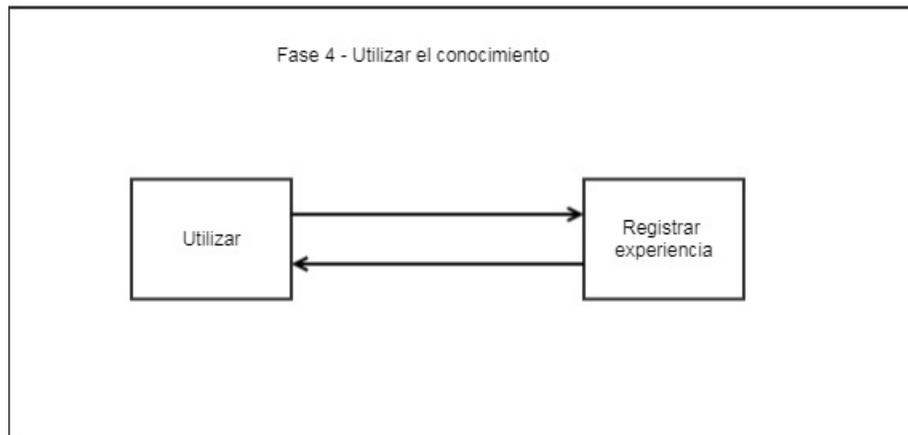


Figura 15. Actividades de la fase 4. Utilizar el conocimiento.

5. Compartir / Validar

Es en esta etapa donde se mezcla la gestión del conocimiento con las metodologías de desarrollo ágiles.

Según Manchanda et al. (2017) las metodologías ágiles para el desarrollo de software han sido implementadas para disminuir el tiempo de desarrollo a través de un modelo iterativo e incremental, en negocios donde los requerimientos son relativamente inestables, es decir que pueden cambiar sobre la marcha. Estas metodologías han sido desarrolladas además para incrementar la satisfacción del cliente, reducir la cantidad de errores y reflejar en el resultado final las necesidades de cambio en la organización que ocurren durante el desarrollo del producto.

Para Abrahamsson et al. (2002) las metodologías ágiles promueven las relaciones y comunicación entre los miembros de un equipo de desarrollo de software, generando relaciones cercanas, espíritu de equipo y buen ambiente de trabajo, además de promover la cooperación

entre el equipo y el cliente. Este tipo de relaciones y espíritu de equipo puede ayudar a facilitar el proceso de GC.

Una de las metodologías ágiles más famosas es Scrum, según Abrahamsson et al. (2002) esta metodología ha sido desarrollada para manejar el proceso de desarrollo de sistemas, aplicando ideas de procesos industriales al desarrollo de software, resultando en un enfoque que reintroduce la idea de la flexibilidad, adaptabilidad y productividad. Scrum no define ninguna técnica específica de desarrollo de software, sino que se concentra en cómo los miembros del equipo deben trabajar en función de producir software en un ambiente cambiante. Scrum ayuda a mejorar las prácticas ingenieriles existentes (ej. Testing) en una organización identificando constantemente deficiencias o impedimentos en el proceso de desarrollo.

Scrum cuenta con diferentes reuniones del equipo de desarrollo de un proyecto, también llamadas “ceremonias”. Según Abrahamsson et al. (2002) estas ceremonias incluyen, entre otras, el *Sprint Planning* (es una reunión al inicio del Sprint, considerando a un sprint como un período de trabajo que dura entre una semana y un mes, en la que se determinarán las actividades que se desarrollarán durante el siguiente período de trabajo, las prioridades y su estimación), el *Sprint Review* o *Sprint Retrospective* (el último día del sprint, el equipo se reúne y analiza los resultados del sprint tomando conciencia de qué se puede mejorar) y el *Daily Scrum Meeting* (el equipo se reúne diariamente para llevar control del progreso del sprint y los bloqueos que pueden existir).

La idea principal de esta etapa del modelo es que en una ceremonia periódica de las metodologías ágiles (por ejemplo, la reunión retrospectiva de scrum), cada miembro del equipo comparta con los demás el conocimiento adquirido y utilizado durante el ciclo de la metodología. Las metodologías ágiles son cíclicas y cada ciclo dura un período corto de tiempo, es por esto por lo que lo ideal es que un ciclo de este modelo de GC dure lo mismo que un ciclo de la metodología de desarrollo. Se recomienda la reunión retrospectiva (si se usa scrum), ya que su propósito es hablar de lo que se debe mejorar y de lo aprendido durante el sprint, sin

embargo, se puede sumar esta fase a cualquier ceremonia o reunión de una metodología ágil, a criterio de la organización.

Si no se utilizan este tipo de metodologías, se debe establecer una reunión periódica (no más de un mes de diferencia entre ellas, para evitar que se pierda el conocimiento adquirido) para realizar esta actividad.

Debe existir un calendario donde se detalle la fecha y hora tentativas de cada una de estas reuniones periódicas, el detalle de este calendario es explicado más adelante.

En esta reunión, cada persona compartirá lo aprendido y describirá forma en que se utilizó. Los demás miembros del equipo podrán intervenir para darle forma al conocimiento y añadir ideas de ser necesario. Así mismo, se validará definitivamente este conocimiento entre todos los miembros del equipo y se determinará la importancia de formalizarlo.

Se deben tratar todos los temas que tiene agendados el líder técnico en la lista de temas a tratar que será explicada más adelante, y al final se debe llenar una minuta con los resultados de la reunión, la cual debe ser enviada a todos los miembros del equipo por email.

Esta etapa podría compararse con las etapas de socialización, externalización y combinación del modelo ISECO. También con la etapa de compartir conocimiento del modelo CEN.

Se debe también durante esta reunión planificar la forma en que cada nuevo conocimiento será formalizado en la wiki de conocimientos, aportando ideas para facilitar y agilizar el proceso.

Las actividades que se deben realizar en esta reunión, por cada tema a compartir son:

a. Explicación del tema

La persona que buscó y utilizó el conocimiento debe compartir el mismo con el equipo de forma breve, debe también justificar el por qué sería útil formalizar el mismo.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización (En cada reunión de proyecto estarán solo las personas involucradas en dicho proyecto)

Documentos: ninguno

Ejemplo: en la reunión J les explica a sus compañeros las mejores formas de diseñar una arquitectura, como diseñó la arquitectura y justifica el por qué debe formalizarse este conocimiento en la wiki de conocimientos.

b. Opiniones del resto del equipo (enriquecimiento)

El resto del equipo debe opinar si le parece que el conocimiento adquirido es válido y agregar posibles cambios o añadidos a lo compartido.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización (En cada reunión de proyecto estarán solo las personas involucradas en dicho proyecto).

Documentos: ninguno

Ejemplo: el resto del equipo opina sobre lo expuesto por J y le sugieren un par de mejoras que puede tener en cuenta a la hora de formalizar el conocimiento.

c. Votación de formalización

Los miembros del equipo presentes en la reunión votarán si le parece relevante o útil que este conocimiento sea agregado a la wiki de conocimientos. En caso de empate, decide el líder del equipo.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización (En cada reunión de proyecto estarán solo las personas involucradas en dicho proyecto).

Documentos: ninguno

Ejemplo: se realiza una votación para determinar si es necesario o no formalizar el tema propuesto por J (J no participa en la votación), el resultado es que SI debe ser formalizado.

d. **Decisión de charla informativa**

Entre los miembros del equipo se procede a decidir si el tema es de tal importancia que amerite una charla informativa para el resto de la organización u otros equipos, esta decisión corresponde al líder del equipo, los demás miembros opinarán al respecto. Sin embargo, que el líder decida no es algo restrictivo, se puede decidir de acuerdo con las necesidades de cada equipo.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización (en cada reunión de proyecto estarán solo las personas involucradas en dicho proyecto).

Documentos: ninguno

Ejemplo: el líder del equipo decide que sería una buena idea llevar este tema al resto de la organización, por lo que da el SI para proceder con la solicitud de charla informativa.

En la siguiente figura (ver figura 16) se muestra el diagrama que explica la presente fase.

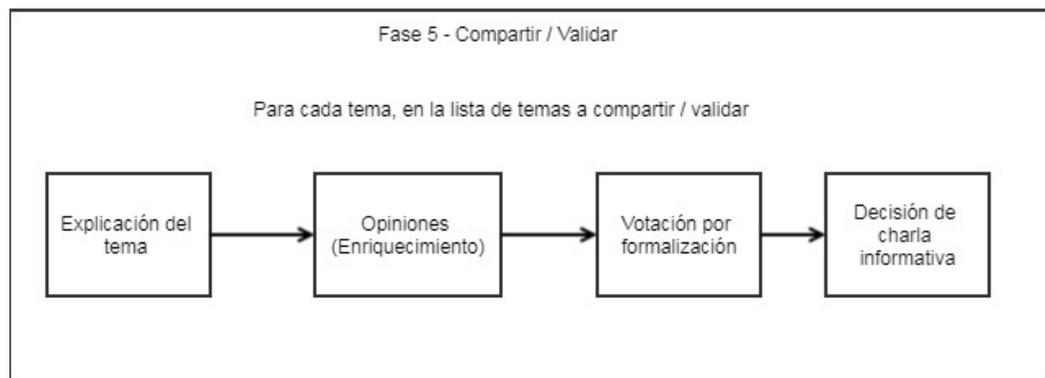


Figura 16. Actividades de la fase 5. Compartir / Validar el conocimiento.

6. Formalizar

Una vez compartido y validado el conocimiento por los miembros del equipo, el conocimiento debe ser formalizado, es decir, llevado a la wiki de conocimientos. Debe ser clasificado debidamente según las clasificaciones establecidas en la wiki de conocimiento. Así mismo, el directorio debe ser actualizado para que ahora refleje que la persona que generó el conocimiento sabe del tema. Siguiendo las fases de este modelo, al haber sido el conocimiento previamente compartido con el resto del equipo, se vuelve más fácil de formalizar puesto que ya ha sido externalizado y los miembros del equipo pueden colaborar con este proceso.

Luego de formalizarse el conocimiento, si éste es de suma importancia para el resto de la organización, se pueden organizar charlas que involucren a otros equipos (por ejemplo: charlas técnicas). Debe existir también un calendario disponible para todos los miembros de la organización con los detalles de cada una de las charlas planificadas, este calendario se debe actualizar cada vez que se planifica una nueva charla informativa, los detalles de este documento se explican más adelante.

Cuando un equipo decide dar una charla informativa, se debe enviar una solicitud al comité de gestión del conocimiento (o algún directivo de la organización designado para supervisar la GC), mediante una planilla que se detalla más adelante. Una vez aprobada la charla, el comité debe entonces actualizar el calendario.

Esta etapa puede ser comparada con la etapa de Almacenamiento del modelo CEN.

Las actividades de esta etapa entonces serían:

a. Formalización del conocimiento

Es volcar todo el conocimiento adquirido en la base de conocimientos, de forma ordenada y fácil de comprender para el resto de la organización. Antes de ser publicado, el contenido formalizado debe ser revisado por el líder del equipo.

Roles: todos los roles involucrados en cada proyecto de desarrollo de software de la organización.

Documentos: Ninguno

Ejemplo: J procede a elaborar un artículo en la wiki de conocimientos con el conocimiento adquirido, teniendo además en cuenta sus registros al haber diseñado la nueva arquitectura.

b. Actualización del directorio

Si la organización cuenta con un directorio con las características señaladas en este capítulo, entonces el líder de proyecto debe actualizar el perfil de la persona para indicar que posee ahora conocimiento sobre el tema. Se puede también notificar por email a los miembros de la organización de que existe un nuevo artículo en la wiki de conocimientos.

Roles: líder de proyecto

Documentos: ninguno

Ejemplo: el líder de proyecto agrega “diseño de arquitectura” al perfil de J en el directorio, así mismo, envía un email notificando de la existencia del nuevo artículo al resto de la organización.

c. Solicitud de charla informativa

Si se decidió en la fase anterior solicitar una charla informativa para el tema en cuestión, se debe enviar la solicitud al comité de GC.

Roles: líder de proyecto, Comité de GC

Documentos: solicitud de charla informativa (ver apartado 4.1.3 c), Respuesta a solicitud de charla informativa (ver apartado 4.1.3 d). Calendario de charlas informativas (ver apartado 4.1.3 b)

Ejemplo: el líder de proyecto envía una solicitud de charla informativa al comité de GC (ver apartado 4.1.3 c), el comité de GC enviará su respuesta (ver apartado 4.1.3 d) y agendará la charla en el calendario de charlas informativas (ver apartado 4.1.3 b)

En la siguiente figura (ver figura 17) se muestran de forma gráfica las actividades de esta fase.

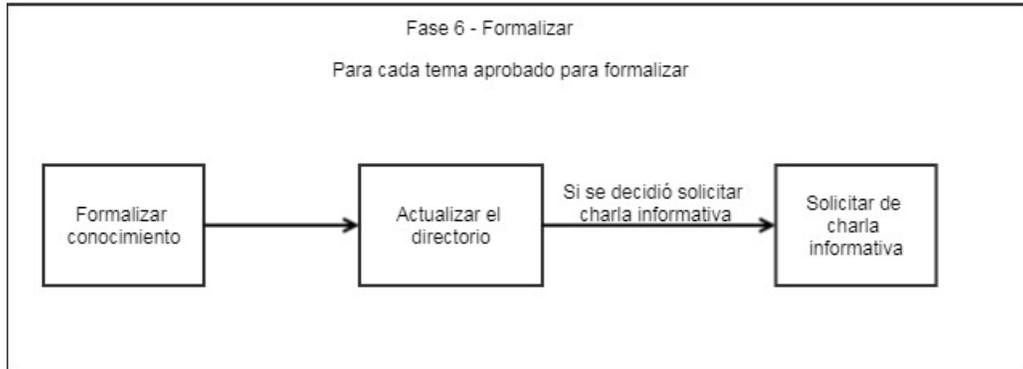


Figura 17. Actividades de la fase 6 – Formalizar.

En la siguiente imagen se muestra un diagrama de flujo en el que se especifica de forma gráfica la relación entre las fases. Es importante aclarar que los pasos expuestos no son determinantes y dependerán de la forma de trabajo de cada organización.

4.1.5. Documentación necesaria para la aplicación del modelo

La documentación necesaria para que este modelo sea aplicable, es sencilla y fácil de llenar, de esta manera no se pierde tiempo del proceso de desarrollo de software y es compatible con las metodologías ágiles.

Entre los documentos necesarios se encuentran:

a. Cronograma de reuniones para la fase de compartir-validar el conocimiento por equipo

Como fue explicado anteriormente, la fase 5 del presente modelo consta de una reunión periódica en donde se valida y comparte el conocimiento que fue adquirido a través del modelo desde la última reunión.

Para ello debe existir un calendario al que todos los miembros del equipo tengan acceso, se puede llenar la planilla planteada en el Anexo 1, enviarla por email a todos los

miembros del equipo y colocarla en un lugar visible, o bien utilizar una herramienta ofimática tipo calendario, en donde se agreguen todas las reuniones, invitando a los miembros del equipo. Si se está utilizando Scrum y se decide unir estas reuniones a las reuniones retrospectivas, bastará con notificar a los miembros del equipo y colocar la aclaración en los calendarios ya utilizados para la metodología de desarrollo.

b. Cronograma de charlas informativas para la organización

Luego de las reuniones para compartir-validar conocimiento, el equipo determina si existe algún conocimiento que vale la pena ser compartido en una charla con el resto de los miembros de la organización. Estas charlas informativas, luego de ser aprobadas, deben ser agregadas al calendario de charlas informativas de gestión del conocimiento. Dicho calendario (Ver Anexo 2) debe estar compartido en un lugar visible a todos los miembros de la organización. Se recomienda enviar por correo electrónico una invitación a los posibles asistentes mediante correo electrónico.

c. Solicitud de charla informativa

Como fue explicado anteriormente, cuando un equipo decide dar una charla informativa, debe enviar una solicitud para realizar la misma al comité de GC (o a algún directivo designado para supervisar la GC), una vez aprobada la solicitud, el comité actualizará el calendario anteriormente explicado. La solicitud debe enviarse por correo electrónico a la persona o personas pertinentes, teniendo como adjunto la planilla presentada en el Anexo 3 ya llena.

d. Respuesta de aprobación/rechazo de charla informativa

Una vez que el comité de GC o personas designadas revisan y consideran la solicitud de charla informativa enviada, deben enviar una respuesta al equipo con el resultado de la evaluación. Esta respuesta debe seguir la plantilla presentada en el Anexo 4

e. Solicitud de compartir-validar conocimiento en el equipo

Cuando un miembro del equipo, siguiendo el modelo explicado, ha encontrado conocimiento en fuentes externas o existe la necesidad de formalizar un conocimiento interno que no está en la wiki, debe enviar una solicitud por mail al líder responsable del equipo, para que de esta manera el tema sea agregado a la lista de temas a discutir en la próxima reunión. El correo debe tener el formato planteado en el Anexo 5.

f. Lista de temas a discutir en la reunión de compartir-validar conocimiento

Cada líder de proyecto debe llevar una lista con los temas a ser compartidos y validados en la próxima reunión periódica, el formato de esta lista se puede encontrar en el Anexo 6.

g. Minuta de reunión compartir-validar conocimiento

Luego de la reunión periódica, el líder del proyecto debe llenar una minuta con los resultados de la reunión (debe ser una minuta a parte si se combinó la reunión de GC con otra reunión de la metodología de desarrollo de software utilizada), la cual debe ser enviada por email a todos los miembros del equipo. El formato de la minuta se puede ver en el Anexo 7.

Es importante mencionar que el formato de estos documentos es una sugerencia del modelo, pero pueden ser adaptados a cada organización según las necesidades de esta.

4.1.6. Ejemplo completo de la aplicación del modelo propuesto

A continuación, se presenta un ejemplo hipotético de aplicación del modelo propuesto:

La empresa ALFA, es una SF multinacional con sede en diversos países del mundo que se dedica a brindar soluciones de software a medida de las necesidades de sus clientes. Tiene equipos de trabajo multidisciplinarios para cada proyecto en el que está involucrada. Normalmente los equipos están conformados por al menos un Project manager, un líder técnico, un analista funcional, un tester, y más

de un desarrollador. En un mismo proyecto se pueden manejar diferentes tecnologías y/o lenguajes de programación dependiendo de las necesidades del cliente. Como es una empresa multinacional, es común que las personas de un mismo proyecto estén distribuidas en puntos geográficos diferentes, en ocasiones, países con diferentes zonas horarias.

La empresa ALFA utiliza la metodología ágil SCRUM en la mayoría de sus proyectos, por lo que adaptarán la fase 5 (compartir/validar) a la reunión retrospectiva que involucra dicha metodología al final de cada ciclo de esta.

La empresa ALFA decide comenzar a implementar el modelo propuesto en esta tesis, ya que se enfrenta con una alta rotación del personal y tiempos reducidos que le impiden aplicar otros modelos de GC, sin embargo, siguen necesitando que el conocimiento quede dentro de la organización, para poder aprovechar todas las ventajas que ofrece el mismo.

Los directivos de ALFA se reúnen en primera instancia para cumplir con la fase de preparación o fase cero del modelo, para esto designan a un comité de GC que está conformado por 3 personas, dos de ellos son *project managers* y el otro es un líder técnico.

El comité seleccionado comienza a realizar reuniones periódicas para determinar en primera instancia qué elementos conformarán la base de conocimientos y que herramientas informáticas están a disposición de la empresa para utilizar como directorio y como wiki de conocimientos.

Luego de n reuniones el comité llega a la conclusión de que por razones de presupuesto adaptarán una herramienta que poseen y ya utilizan como wiki para que sea apta para la aplicación del modelo, para ello designarán un equipo que está disponible. Los cambios que decidieron aplicarle a la herramienta fueron:

- la posibilidad de categorizar los artículos subidos de forma que se puedan buscar por tema, proyecto y persona que lo elaboró,
- posibilidad de que al buscar un tema determinado en la herramienta aparezcan los perfiles de las personas que podrían tener información al respecto con sus datos de contacto (para utilizar esta característica como directorio)

- La posibilidad de buscar en la herramienta a una persona en específico para poder ver su perfil, el cual debe contener enlaces a los artículos que ha escrito y enlaces a los perfiles de los proyectos en los que ha trabajado.
- Una página de perfil para cada proyecto en donde se especifiquen las tecnologías utilizadas, razones de éxito y de fracaso y lecciones aprendidas (en cada proyecto se decidirá qué información puede ser pública y qué información no)

El comité también llega a la conclusión de que es necesario crear una lista de email para que la gente pueda preguntar en caso de dudas donde no se logra conseguir la información necesaria y no se sabe de una persona específica a la cual preguntar.

También se decide que la base de conocimientos estará integrada por un repositorio de documentos a los cuales todos los empleados podrán tener acceso. En la wiki de conocimientos se podrá también vincular artículos con documentos de este repositorio.

Así mismo, llegan a la conclusión de que se premiará al final cuatrimestre, a las 3 personas que más artículos validados hayan publicado durante el período.

Una vez adaptada la herramienta, comienza el proceso de capacitación sobre el modelo. Para ello, el comité genera un manual de instrucciones que envía por email a todos los miembros de la organización e inicia un ciclo de charlas informativas, primero con los *project managers* y líderes de proyectos para explicarles su rol en el proceso y la importancia de la aplicación del modelo, y luego con el resto de los miembros de la organización.

A partir de este momento, se reducirá este ejemplo a un equipo y proyecto en particular perteneciente a la empresa ALFA.

El equipo BAZINGA se ve en la obligación de comenzar con la aplicación del modelo. Es un equipo conformado por un *project manager*, un líder técnico, un analista funcional, 3 desarrolladores de C#, 3 desarrolladores de JAVA y dos *testers*. El analista funcional y 2 de los desarrolladores se encuentran en otro país cuya zona horaria está a 2 horas menos que la sede donde se encuentra el resto del equipo.

El equipo ya está acostumbrado a trabajar con las diferencias horarias, por lo que sus ceremonias de SCRUM son en horarios en donde todos están trabajando.

En primera instancia el PM y el líder técnico se reúnen para determinar de qué forma se adaptará el modelo en su equipo al ritmo de trabajo y procesos de este. De igual manera, deciden cómo serán de ahora en adelante las reuniones retrospectivas de SCRUM, a las que se le agregarán 20 minutos al final para realizar la fase 5 del modelo. Llenan entonces el calendario de reuniones (Anexo 1)

Paso siguiente, se reúnen con el resto del equipo para comunicarles lo decidido y motivarlos a aplicar el modelo, aclarándoles sin embargo que la prioridad es seguir entregando valor al cliente al mismo ritmo actual.

El *project manager* crea el perfil del proyecto en la wiki de conocimientos (dejando lugar para ir agregando las lecciones aprendidas a medida que se desarrolla el mismo). Cada miembro de la organización crea entonces su propio perfil en el que agrega sus datos de contacto profesional y sus áreas de experticia. El perfil de cada uno quedará vinculado al perfil del proyecto en el que trabaja. Los PM pueden también crear perfiles de proyectos ya finalizados y vincularlos con los miembros de éstos.

Comienza entonces un nuevo ciclo (sprint) de scrum para el equipo BAZINGA, en el cual se comenzará a aplicar el modelo, cada miembro comienza con sus labores diarias como de costumbre.

Uno de los desarrolladores (Miembro llamado A), necesita saber cuál es cómo solicitar un acceso determinado a uno de los servidores del cliente, que necesita para continuar con su trabajo, por su parte, el *tester* (miembro B), está trabajando para automatizar el proceso de testeo y tiene una duda muy particular relacionada a la tecnología que está utilizando, si no logra resolver esta duda no podrá continuar con una tarea que tiene asignada.

De esta forma comienza la aplicación de las fases principales de la metodología para A y B.

En el caso de A:

- i. Detectar la necesidad de conocimiento: A se da cuenta que necesita adquirir este conocimiento y procede a hacerse las 4 preguntas básicas que determinan si debe o no

continuar con las demás fases. En su caso, la respuesta a la pregunta ¿Es esta necesidad de conocimiento un bloqueo para terminar mis tareas? es un SI, por lo que debe continuar con el modelo.

- ii. **Buscar:** A procede a buscar en la Wiki de conocimientos, no consigue nada, pero navegando por la wiki, nota que dos desarrolladores ubicados en otra oficina de la empresa han trabajado en un proyecto que utilizó servidores similares del mismo cliente (ya que el perfil de dicho proyecto contenía información básica de las tecnologías y medios utilizados), automáticamente A busca en el directorio los datos de contacto de estas personas y envía un email a ambos con sus dudas. La respuesta a su email fue un detalle de los pasos que siguió una de las personas para solicitar los accesos cuando lo necesitó. A comunica a su líder de proyecto que ha encontrado este conocimiento y le parece importante que sea formalizado en la wiki de conocimiento para que sea útil en futuros proyectos, para esto A envía un email con una solicitud para compartir y validar conocimiento (Anexo 5) al líder técnico del equipo, el líder da el OK y agrega el tema a la lista de temas que se discutirán en la reunión que tendrá lugar al finalizar el sprint.
- iii. **Interiorizar:** A, al leer el email, consulta con sus líderes para ver si los pasos recibidos no violan ninguna política de la empresa o del cliente, al disipar esta duda, procede a hacer una lista ordenada de los pasos a seguir, de esta manera A logra organizar la información y asimilarla.
- iv. **Utilizar:** A aplica los pasos necesarios para solicitar el acceso al servidor y mientras lo hace va registrando su experiencia (por ejemplo, algún paso que no estaría registrado en la lista que recibió).

Ahora A sigue con sus tareas y espera a la reunión de Compartir/Validar conocimiento que se hará dentro de la reunión retrospectiva de scrum.

El caso de B es diferente al de A, puesto que se trata de una duda técnica y muy específica.

- 1- Detectar necesidad de conocimiento: B detecta que necesita este conocimiento para continuar, y procede a hacerse las 4 preguntas básicas. Como es un bloqueo tener esta necesidad para seguir con su trabajo, entonces continúa con el modelo.
- 2- Buscar: B procede a buscar la información, no la consigue en ninguna fuente interna, a pesar de haberse comunicado con personas que manejan esa tecnología (luego de buscar en el directorio), así que decide buscar más a fondo en internet. B consigue la respuesta, sin embargo, no considera que sea necesario formalizar este conocimiento debido a lo particular del caso. Lo consulta con su líder técnico y este opina lo mismo.
- 3- Interiorizar: B organiza con un gráfico en una hoja de papel la información encontrada y de esta forma la asimila.
- 4- Utilizar: B utiliza el conocimiento y prosigue con su tarea. Como se decidió que no era necesario formalizar este conocimiento, no envía ninguna solicitud a su líder técnico y el tema no será tratado en la reunión de compartir/validar.

Durante el sprint, hubo otros miembros del equipo con casos similares que siguieron el modelo y ahora sus temas están en la lista de temas a tratar en la reunión de compartir/validar conocimiento.

- 5- Compartir/Validar: llega el momento de la reunión, luego de que termina la retrospectiva común de scrum, proceden a discutir los temas de la lista.

A explica cómo fue el proceso y la información que consiguió, todos están de acuerdo en que el conocimiento debe formalizarse, sin embargo, también están de acuerdo en que no es necesario una charla para el resto de la organización. De igual forma se procede a tratar todos los demás temas en la lista, incluidos los temas correspondientes a personas del equipo que se encuentran en otras oficinas (utilizando diferentes medios de comunicación en tiempo real).

A tiene entonces el permiso del equipo para proceder a formalizar el conocimiento adquirido.

- 6- Formalizar: A, entonces agrega un artículo estructurado en la wiki de conocimientos sobre cómo pedir los permisos necesarios para acceder al servidor en cuestión. Así mismo, su perfil en el directorio se actualiza automáticamente, teniendo ahora el artículo vinculado. De haber estado de acuerdo todo el equipo en la necesidad de una charla informativa, el líder técnico debía enviar una solicitud de charla al comité de GC (Anexo 3), y el comité, luego de evaluar la petición, debe responder con el resultado de la solicitud (Anexo 4) y de ser afirmativa la respuesta, agregarlo al calendario de charlas informativas (Anexo 1).

Comienza entonces un nuevo sprint y el proceso se repite.

Mientras tanto el comité de GC se encarga de la fase paralela de mantenimiento, asegurándose de que se hagan los debidos respaldos de toda la información, y al final del cuatrimestre organizan un evento pequeño para premiar a las personas que más contribuyeron con la wiki de conocimientos. De igual forma envían por email una encuesta para recibir *feedback* y decidir qué mejoras se pueden aplicar a la implementación del modelo.

4.1.7. Análisis del modelo en relación con los desafíos de GC para SF

A continuación, se presenta el mismo análisis que fue realizado para los demás modelos en este caso en particular.

Desafío	¿Tomado en cuenta?	Comentario
Cambios constantes en las tecnologías utilizadas.	SI	Por la forma cíclica en la que está planteado el modelo, acompañando el proceso de desarrollo de software, permite adquirir conocimiento nuevo a la organización cuando sea detectada una necesidad, esta necesidad entre otras cosas podría ser a raíz de un cambio tecnológico.
Necesidad de buscar conocimiento fuera de la organización	SI	En la etapa de búsqueda de conocimiento, propone la búsqueda en fuentes externas cuando el conocimiento no se encuentra dentro de la organización.

Contextos y procesos difieren en cada proyecto.	SI	Entre las características de la base de conocimiento se propone que cada proyecto que ha existido en la organización tenga un espacio en donde se describa de ser posible el contexto y los resultados obtenidos, así como también causas de fallo de haber sido el caso, cada pieza de conocimiento debe vincularse con el proyecto de donde surgió, esto permite contextualizar el conocimiento por proyecto.
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	SI	El modelo permite agregar conocimiento cuando sea necesario realmente, teniendo en cuenta que puede estar relacionado a un proyecto a al dominio de la ingeniería, es por esto por lo que responder las preguntas propuestas en la fase 1 es necesario. A su vez el conocimiento debe estar bien clasificado en la base de conocimientos para que sea más fácil acceder al mismo.
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	SI	Las preguntas planteadas en la fase 1 del modelo (específicamente la pregunta 4) ayudan a detectar cuando es necesario incorporar a la organización el conocimiento de convenciones de desarrollo locales o "mejores prácticas" relacionadas a la calidad de los productos.
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	SI	La existencia y organización del directorio propuesto ayuda a los miembros de la organización a encontrar a personas que cuenten con el conocimiento necesario en el momento.
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	SI	El modelo está diseñado para acompañar el proceso de desarrollo de software sin consumir demasiado tiempo por parte de los miembros de la organización.
Conocimiento altamente especializado, difícil de formalizar.	SI	El hecho de compartir el conocimiento con los miembros del equipo en una reunión hace que el conocimiento sea más fácil de formalizar, ya que las ideas se pueden organizar en colaboración con los demás miembros del equipo.

Alta rotación del personal vuelve al conocimiento volátil.	SI	Con el modelo propuesto, el conocimiento no solo queda formalizado en un medio digital, sino que además queda interiorizado en diversos miembros de la organización (debido a la forma de compartirlo), esto reduce la volatilidad de este.
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	SI	El modelo propone que en la base de conocimiento exista una sección dedicada a cada proyecto, en la cual, entre otras cosas, se especifiquen las razones de fracaso si este fue el caso.
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	SI	En el modelo se habla de la posible existencia de equipos globales, en esos casos propone que en la etapa “compartir-validar” se utilicen medios tecnológicos para incorporar en las reuniones a todos los involucrados en el proyecto, independientemente de la posición geográfica, y en horarios en los que todos los miembros estén presentes (si se utilizan reuniones propias de la metodología de desarrollo para esta etapa, este problema ya estaría resuelto con anterioridad).
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	SI	Como el conocimiento se comparte y valida por todos los miembros del equipo antes de ser formalizado, este es explicado e interiorizado por los mismos sin importar su rol, al menos de forma general. La existencia del directorio y la base de conocimientos también ayuda a que el conocimiento sea accesible a todos los miembros de la organización sin importar su rol.
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	SI	Como el conocimiento se comparte y valida por todos los miembros del equipo antes de ser formalizado, este es explicado e interiorizado por los mismos sin importar su seniority, al menos de forma general. La existencia del directorio y la base de conocimientos también ayuda a que el conocimiento sea accesible a todos los miembros de la organización sin importar la experiencia.

Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	SI	El conocimiento es compartido con todos los miembros del equipo antes de ser formalizado, por lo cual es interiorizado por los mismos sin importar en que tecnología se especializan (al menos de forma general). La existencia del directorio y la base de conocimientos también ayuda a que el conocimiento sea accesible a todos los miembros de la organización sin importar la tecnología.
Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	SI	El modelo propone que el conocimiento sea buscado primero en fuentes internas (base de conocimiento y directorio), antes que, en fuentes externas, de esta forma se incentiva la reutilización del conocimiento. Como todo conocimiento pasa también por una etapa de validación, se reduce el riesgo de redundancia en el conocimiento formalizado.

Tabla 20. Análisis del modelo propuesto en relación con los desafíos de GC definidos. Elaboración propia.

5. VALIDACIÓN

Se entiende como validez del contenido al grado en que los ítems que constituyen un instrumento son una muestra representativa del dominio de contenido del tema y por lo general esta toma de validez se ha determinado mediante la comparación sistemática de los ítems del instrumento con el dominio del contenido estudiado. (Escobar y Cuervo, 2008).

Es decir, para validar el contenido de la metodología propuesta, es necesario comparar sus ítems con el dominio de conocimiento relacionado con la gestión de conocimiento en las fábricas de software.

Para ello es posible la utilización de un método de validación llamado “Juicio de expertos”, entendiéndose como experto tanto un individuo en si, como un grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema y hacer recomendaciones al respecto, teniendo siempre un máximo de competencia (Michalus et al 2015).

“La evaluación mediante el juicio de experto consiste, básicamente, en solicitar a una serie de personas la demanda de un juicio hacia un objeto, un instrumento, un material de enseñanza, o su opinión respecto a un aspecto concreto.” (Cabero & Llorente, 2013).

Por su parte, para Escobar y Cuervo (2008) el juicio de expertos se define como una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos cualificados en éste, y que pueden dar información, evidencia, juicios y valoraciones.

Escobar y Cuervo (2008) también afirman que el juicio de expertos en muchas áreas es una parte importante de la información cuando las observaciones experimentales están limitadas.

En el caso de este modelo, debido a limitaciones de tiempo y de instrumentos exactos para medir el éxito de un modelo de gestión de conocimiento en fábricas de software (las métricas comunes de un proyecto de software pueden depender de muchos factores, por lo que utilizar éstas no es acertado para medir el modelo en particular), se utilizará el juicio de expertos como instrumento principal de validación.

Cabero y Llorente (2013) han detallado las ventajas que posee la utilización de este instrumento, y se enuncian a continuación:

- la teórica calidad de la respuesta que se obtiene de la persona,
- el nivel de profundización de la valoración que se ofrece,
- su facilidad de puesta en acción,
- la no exigencia de muchos requisitos técnicos y humanos para su ejecución,
- el poder utilizar en ella diferentes estrategias para recoger la información es de gran utilidad para determinar el conocimiento sobre contenidos y temáticas difíciles, complejas y novedosas o poco estudiadas,
- la posibilidad de obtener información pormenorizada sobre el tema sometido a estudio, para lo cual es necesario poder contar con diferentes tipos de expertos.

Según Cabero y Llorente (2013) existen distintas formas de aplicar el juicio de expertos:

- Agregación individual de los expertos: consiste en obtener la información de manera individual de cada uno de ellos, sin que estos se encuentren en contacto.
- Método Delphi: consiste en recoger la opinión de los expertos de forma individual y anónima, devolviéndoles la propuesta de conjunto para su revisión y acuerdo, una leve dispersión llevará a afirmar que se ha llegado a un acuerdo.
- Técnica grupal nominal: los expertos aportan la información de manera individual, y después de forma grupal presencial se llega a un acuerdo.
- Método de consenso, donde de forma grupal y conjuntamente, los expertos seleccionados llegan a conseguir un acuerdo.

Para este trabajo, debido a la cantidad de expertos necesarios, se utilizará el método de agregación individual de expertos, llevándole la información a cada uno de ellos, recogiendo su opinión por separado y calculando luego la concordancia entre los jueces.

Los pasos que se deben seguir para aplicar el juicio de expertos son los siguientes (Escobar y Cuervo, 2008):

- **Definir el objetivo del juicio de expertos.** En este apartado los investigadores deben tener clara la finalidad del juicio
- **Selección de los jueces.** Se deben tener en cuenta diversos factores como la formación académica de los expertos y su experiencia y reconocimiento y evaluación.
- **Explicitar tanto las dimensiones como los indicadores que está midiendo cada uno de los ítems de la prueba.** Esto le permitirá al juez evaluar la relevancia, la suficiencia y la pertinencia del ítem.
- **Especificar el objetivo de la prueba.** El autor debe proporcionar a los jueces la información relacionada con el uso de la prueba, es decir, el objetivo de esta, esto incrementará el nivel de especificidad de la evaluación.
- **Establecer los pesos diferenciales de las dimensiones de la prueba.** Se lleva a cabo cuando algunas de las dimensiones tienen pesos diferentes.
- **Diseño de planillas.** La planilla se debe diseñar de acuerdo con los objetivos de la evaluación.
- **Calcular la concordancia entre jueces.** Para esto se utilizan los estadísticos Kappa y Kendall que se explicarán más adelante.
- **Elaboración de las conclusiones del juicio**

A continuación, se van a explicar cada uno de los pasos seguidos anteriormente definidos en función a la presente investigación.

5.1 Definir el objetivo del juicio de expertos.

El objetivo principal del juicio de expertos en esta investigación será: Validar el modelo de GC para SF propuesto mediante la opinión de expertos previamente seleccionados, para que de esta forma pueda ser aplicado en SF en futuras oportunidades.

5.2 Selección de los jueces

Existen diversos métodos para seleccionar a los expertos adecuados para la validación de un tema en particular. Para Cabero y Llorente (2013) existen métodos que van desde una forma

desarticulada que no dependen de ningún patrón, hasta métodos más estructurados como el biorama o el coeficiente K. Ambos autores explican como métodos estructurados al biorama y al coeficiente de competencia experta (coeficiente k) de la siguiente manera:

- **Biorama:** se elaboran biografías de los expertos incorporando factores de cada uno de ellos que puedan estar relacionados con el tema que se desea validar (lugar de trabajo, experiencia, actividades desarrolladas, entre otros). Se escogerán entonces los expertos que posean mayor número de justificaciones que le permitan justificar al evaluador los motivos de la selección.
- **Coficiente de competencia experta (coeficiente K):** se combina un valor asignado a la opinión propia de cada uno de los posibles expertos sobre su nivel de conocimiento en diferentes aspectos e indicadores relacionados con el tema a validar con otros factores que pueden influir en la opinión del experto como lo son: experiencia laboral, publicaciones científicas, estudios, entre otros.

Para la obtención del coeficiente de competencia experto se aplica la fórmula siguiente:

$K = \frac{1}{2} (Kc + Ka)$, donde:

Kc es el “Coeficiente de conocimiento” o información que tiene el experto acerca del tema. Es calculado a partir de la valoración que realiza el propio experto en la escala del 0 al 10, multiplicado por 0,1.

Ka es el denominado “Coeficiente de argumentación” o fundamentación de los criterios de los expertos. Este coeficiente se obtiene a partir de la asignación de una serie de puntuaciones a diferentes fuentes de argumentación que ha podido esgrimir el experto.

Se escogerán entonces aquellos expertos cuyo coeficiente K sea mayor.

Para esta investigación se ha decidido escoger el coeficiente K, ya que tiene una base matemática que considera tanto la propia opinión del experto como su experiencia relacionada a algún aspecto del tema a validar. A pesar de incluir la subjetividad que puede poner el experto en cuestión, elimina la subjetividad que puede incluir el investigador en la selección de expertos.

Cabrero & Llorente (2013), señalan que no existe un acuerdo entre los autores en el número de expertos necesarios para validar un tema en particular, sin embargo, han nombrado a varios autores

que indican que el número debe oscilar entre 7 y 35, **en esta investigación se utilizó por cuestiones de tiempo un el número de 7 expertos.**

Si un 80 % de los expertos han estado de acuerdo con la validez de un ítem en particular, este puede ser considerado como válido. (Voutilainen y Liukkonen, 1995, citado en Escobar y Cuervo 2008).

Para esta investigación se definieron las distintas fuentes de argumentación que se evaluarán en cada uno de los expertos, a cada fuente de argumentación se le asignó un peso de 1 punto. La cantidad de puntos que suma el experto de presentar n características, multiplicada por 0.1 (ya que son 10 características) genera el coeficiente de argumentación (k) de dicho experto. La siguiente tabla (ver tabla 21) muestra las fuentes de argumentación seleccionadas:

Fuente de argumentación
Tiene experiencia laboral en el área de GC
Tiene experiencia manejando el conocimiento en una SF o en un equipo de desarrollo
Tiene experiencia trabajando en equipos de desarrollo en SF
Tiene experiencia liderando equipos de desarrollo en SF
Tiene experiencia liderando al menos un área de una SF
Tiene experiencia administrando equipos y proyectos de desarrollo de software en una SF (Project manager)
Tiene experiencia trabajando en equipos de desarrollo de software con diferente distribución geográfica.
Tiene experiencia trabajando en equipos de desarrollo de software multidisciplinarios
Tiene estudios académicos y/o experiencia investigativa relacionados con el área de GC.
Tiene estudios académicos y/o experiencia investigativa relacionados con liderazgo y gestión de equipos y/o proyectos en una SF

Tabla 21. Fuentes de argumentación seleccionadas. Elaboración propia.

Para calcular el coeficiente de conocimiento (k_c), se le explicó a cada experto de forma breve la investigación, y se le pidió que enumere del 1 al 10, lo capacitado que se siente en cuanto a sus conocimientos para ser parte del grupo de jueces que validen el presente trabajo.

5.2.1. Expertos seleccionados

A continuación, se muestran los resultados del coeficiente de competencia experta (K), para cada uno de los expertos seleccionados. Los valores dados a cada fuente de argumentación han sido sacados de los CV de los expertos y comunicaciones con los mismos. Los CV de los expertos seleccionados se pueden ver del ANEXO I al anexo O.

$K = \frac{1}{2} (Kc + Ka)$ en donde:

Kc: Coeficiente de conocimiento (Auto evaluación * 0,1)

Ka: Coeficiente de argumentación (Suma fuentes de argumentación * 0,1)

Andrei Rukavina

Cálculo del Ka:

Fuente de argumentación	
Tiene experiencia laboral en el área de GC	0
Tiene experiencia manejando el conocimiento en una SF o en un equipo de desarrollo	1
Tiene experiencia trabajando en equipos de desarrollo en SF	1
Tiene experiencia liderando equipos de desarrollo en SF	1
Tiene experiencia liderando al menos un área de una SF	1
Tiene experiencia administrando equipos de desarrollo de software en una SF (Project manager)	1
Tiene experiencia trabajando en equipos de desarrollo de software con diferente distribución geográfica.	1
Tiene experiencia trabajando en equipos de desarrollo de software multidisciplinarios	1
Tiene estudios académicos y/o experiencia investigativa relacionados con el área de GC.	0
Tiene estudios académicos y/o experiencia investigativa relacionados con liderazgo y gestión de equipos y/o proyectos de software	0
Total	$Ka = 7 * 0.1 = 0.7$

Tabla 22. Cálculo de Ka. AndreiRukavina. Elaboración propia.

Resultado de auto evaluación: 8

$$Kc: 8 * 0,1 = 0,8$$

$$K = \frac{1}{2} (0,7+0,8) = 0,75$$

Sebastián Lopreto

Cálculo del Ka:

Fuente de argumentación	
Tiene experiencia laboral en el área de GC	0
Tiene experiencia manejando el conocimiento en una SF o en un equipo de desarrollo	1
Tiene experiencia trabajando en equipos de desarrollo en SF	1
Tiene experiencia liderando equipos de desarrollo en SF	1
Tiene experiencia liderando al menos un área de una SF	0
Tiene experiencia administrando equipos de desarrollo de software en una SF (Project manager)	1
Tiene experiencia trabajando en equipos de desarrollo de software con diferente distribución geográfica.	1
Tiene experiencia trabajando en equipos de desarrollo de software multidisciplinarios	1
Tiene estudios académicos y/o experiencia investigativa relacionados con el área de GC.	0
Tiene estudios académicos y/o experiencia investigativa relacionados con liderazgo y gestión de equipos y/o proyectos de software	1
Total	$Ka = 7 * 0,1 = 0,7$

Tabla 23. Cálculo de Ka. Sebastián Lopreto. Elaboración propia.

Resultado de auto evaluación: 7,5

$$Kc: 7,5 * 0,1 = 0,75$$

$$K = \frac{1}{2} (0,7 + 0,75) = 0,72$$

Ramiro Garbarini

Cálculo del Ka:

Fuente de argumentación	
Tiene experiencia laboral en el área de GC	0
Tiene experiencia manejando el conocimiento en una SF o en un equipo de desarrollo	0
Tiene experiencia trabajando en equipos de desarrollo en SF	1
Tiene experiencia liderando equipos de desarrollo en SF	1
Tiene experiencia liderando al menos un área de una SF	0
Tiene experiencia administrando equipos de desarrollo de software en una SF (Project manager)	1
Tiene experiencia trabajando en equipos de desarrollo de software con diferente distribución geográfica.	1
Tiene experiencia trabajando en equipos de desarrollo de software multidisciplinarios	1
Tiene estudios académicos y/o experiencia investigativa relacionados con el área de GC.	1
Tiene estudios académicos y/o experiencia investigativa relacionados con liderazgo y gestión de equipos y/o proyectos de software	1
Total	Ka = 7 * 0,1 = 0,7

Tabla 24. Cálculo de Ka. Ramiro Garbarini. Elaboración propia.

Resultado de auto evaluación: 7

$$Kc: 7,0 * 0,1 = 0,7$$

$$K = \frac{1}{2} (0,7 + 0,7) = 0,7$$

Martin Riera

Cálculo del Ka:

Fuente de argumentación	
Tiene experiencia laboral en el área de GC	1
Tiene experiencia manejando el conocimiento en una SF o en un equipo de desarrollo	0
Tiene experiencia trabajando en equipos de desarrollo en SF	1
Tiene experiencia liderando equipos de desarrollo en SF	1
Tiene experiencia liderando al menos un área de una SF	1
Tiene experiencia administrando equipos de desarrollo de software en una SF (Project manager)	1
Tiene experiencia trabajando en equipos de desarrollo de software con diferente distribución geográfica.	1
Tiene experiencia trabajando en equipos de desarrollo de software multidisciplinarios	1
Tiene estudios académicos y/o experiencia investigativa relacionados con el área de GC.	0
Tiene estudios académicos y/o experiencia investigativa relacionados con liderazgo y gestión de equipos y/o proyectos de software	1
Total	Ka = 8 * 0,1 = 0,8

Tabla 25. Cálculo de Ka. Martín Riera. Elaboración propia.

Resultado de auto evaluación: 7

$$Kc: 7,0 * 0,1 = 0,7$$

$$K = \frac{1}{2} (0,7 + 0,8) = 0,75$$

Marcelo Masci

Cálculo del Ka:

Fuente de argumentación	
Tiene experiencia laboral en el área de GC	0
Tiene experiencia manejando el conocimiento en una SF o en un equipo de desarrollo	0
Tiene experiencia trabajando en equipos de desarrollo en SF	1
Tiene experiencia liderando equipos de desarrollo en SF	1
Tiene experiencia liderando al menos un área de una SF	0
Tiene experiencia administrando equipos de desarrollo de software en una SF (Project manager)	1
Tiene experiencia trabajando en equipos de desarrollo de software con diferente distribución geográfica.	1
Tiene experiencia trabajando en equipos de desarrollo de software multidisciplinarios	1
Tiene estudios académicos y/o experiencia investigativa relacionados con el área de GC.	0
Tiene estudios académicos y/o experiencia investigativa relacionados con liderazgo y gestión de equipos y/o proyectos de software	1
Total	$Ka = 6 * 0,1 = 0,6$

Tabla 26. Cálculo de Ka. Marcelo Masci. Elaboración propia.

Resultado de auto evaluación: 8

$$Kc: 8,0 * 0,1 = 0,8$$

$$K = \frac{1}{2} (0,6 + 0,8) = 0,7$$

Pablo Cigliuti

Cálculo del Ka:

Fuente de argumentación	
Tiene experiencia laboral en el área de GC	0
Tiene experiencia manejando el conocimiento en una SF o en un equipo de desarrollo	1
Tiene experiencia trabajando en equipos de desarrollo en SF	1
Tiene experiencia liderando equipos de desarrollo en SF	1
Tiene experiencia liderando al menos un área de una SF	1
Tiene experiencia administrando equipos de desarrollo de software en una SF (Project manager)	0
Tiene experiencia trabajando en equipos de desarrollo de software con diferente distribución geográfica.	1
Tiene experiencia trabajando en equipos de desarrollo de software multidisciplinarios	1
Tiene estudios académicos y/o experiencia investigativa relacionados con el área de GC.	0
Tiene estudios académicos y/o experiencia investigativa relacionados con liderazgo y gestión de equipos y/o proyectos de software	1
Total	$Ka = 7 * 0,1 = 0,7$

Tabla 27. Cálculo de Ka. Pablo Cigliuti. Elaboración propia.

Resultado de auto evaluación: 7

$$Kc: 7,0 * 0,1 = 0,7$$

$$K = \frac{1}{2} (0,7 + 0,7) = 0,7$$

Rafael Natera

Cálculo del Ka:

Fuente de argumentación	
Tiene experiencia laboral en el área de GC	0
Tiene experiencia manejando el conocimiento en una SF o en un equipo de desarrollo	1
Tiene experiencia trabajando en equipos de desarrollo en SF	1
Tiene experiencia liderando equipos de desarrollo en SF	1
Tiene experiencia liderando al menos un área de una SF	0
Tiene experiencia administrando equipos de desarrollo de software en una SF (Project manager)	1
Tiene experiencia trabajando en equipos de desarrollo de software con diferente distribución geográfica.	1
Tiene experiencia trabajando en equipos de desarrollo de software multidisciplinarios	1
Tiene estudios académicos y/o experiencia investigativa relacionados con el área de GC.	0
Tiene estudios académicos y/o experiencia investigativa relacionados con liderazgo y gestión de equipos y/o proyectos de software	0
Total	$Ka = 6 * 0,1 = 0,6$

Tabla 28. Cálculo de Ka. Rafael Natera. Elaboración propia.

Resultado de auto evaluación: 8

$$Kc: 8,0 * 0,1 = 0,8$$

$$K = \frac{1}{2} (0,8 + 0,6) = 0,7$$

La siguiente tabla (Ver tabla 29) resume el resultado de la selección de los expertos junto con su coeficiente K.

Experto	Coeficiente K
AndreiRukavina	0,75
Sebastián Lopreto	0,72
Ramiro Garbarini	0,70
Martin Riera	0,75
Marcelo Masci	0,70
Pablo Cigliuti	0,70
Rafael Natera	0,70

Tabla 29. Resultados de la selección de expertos, expertos seleccionados. Elaboración propia

Se escogieron a todos los expertos con coeficiente K superior o igual a 0,7.

5.3. Explicar las dimensiones e indicadores a medir (definición de los ítems de evaluación)

Las dimensiones por medir están vinculadas directamente a los 15 desafíos de la GC en SF que se definieron durante esta investigación, y los indicadores servirán para determinar qué tanto este modelo resuelve los mencionados desafíos.

En la siguiente tabla se presentan las dimensiones e indicadores para cada dimensión:

Dimensiones	Indicadores
Cambios constantes en las tecnologías utilizadas.	<ul style="list-style-type: none"> - Permite a los miembros de la organización mantenerse actualizados con las tecnologías a medida que sea necesario
Necesidad de buscar conocimiento fuera de la organización	<ul style="list-style-type: none"> - Permite buscar conocimiento fuera de la organización - Permite que el conocimiento adquirido desde fuera de la organización sea introducido en la misma
Contextos y procesos difieren en cada proyecto	<ul style="list-style-type: none"> - Permite que el conocimiento sea compartido entre diferentes proyectos a pesar de que los contextos sean diferentes - Permite que una persona que estaba en otro proyecto pueda adaptarse y adquirir el conocimiento del nuevo proyecto en el que participará - Permite guardar información pertinente a cada proyecto (contexto, procesos, lecciones aprendidas...)
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	<ul style="list-style-type: none"> - Permite a los miembros de la organización adquirir y compartir conocimiento del dominio de cada proyecto. - Permite a los miembros de la organización adquirir y compartir conocimiento de temas generales de ingeniería de sistemas y desarrollo de software.

<p>Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).</p>	<ul style="list-style-type: none"> - Permite que el conocimiento sobre convenciones de desarrollo de software como “buenas prácticas” sea adquirido por los miembros de la organización - Permite que el conocimiento sobre convenciones de desarrollo de software como “buenas prácticas” sea compartido entre los miembros de la organización
<p>Es necesario saber “Quién sabe qué cosa” para reducir tiempos</p>	<ul style="list-style-type: none"> - Permite ubicar a las personas dentro de la organización por sus conocimientos en temas específicos. - Permite contactar a las personas que tienen conocimiento sobre temas específicos para compartir conocimiento
<p>El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.</p>	<ul style="list-style-type: none"> - No afecta de forma considerable el tiempo destinado al desarrollo de software. - Permite que el conocimiento sea formalizado a pesar de los tiempos reducidos en una SF. - Motiva a las personas a formalizar el conocimiento.
<p>Conocimiento altamente especializado, difícil de formalizar.</p>	<ul style="list-style-type: none"> - Facilita el proceso de formalización del conocimiento, al brindar herramientas y soporte en otros miembros de la organización.
<p>Alta rotación del personal vuelve al conocimiento volátil.</p>	<ul style="list-style-type: none"> - Permite que el conocimiento quede distribuido entre los miembros de la organización y no en una sola persona - Facilita el proceso de adquisición de conocimiento de nuevos miembros en la organización o en un proceso en particular - Permite que el conocimiento sea formalizado y persistido en un medio de fácil acceso a los miembros de la organización
<p>Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.</p>	<ul style="list-style-type: none"> - Permite guardar información sobre lecciones aprendidas en un proyecto en particular. - Permite compartir información sobre lecciones aprendidas en un proyecto entre los miembros de la organización
<p>En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.</p>	<ul style="list-style-type: none"> - Permite que el modelo pueda ser seguido de forma remota por personas en zonas geográficas u horarias diferentes.

	<ul style="list-style-type: none"> - Permite que el conocimiento sea compartido entre miembros de la organización a pesar de las diferencias geográficas, horarias y culturales. - Permite que miembros de la organización que estén en locaciones geográficas u horarias diferentes tengan acceso al conocimiento de la organización.
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	<ul style="list-style-type: none"> - Permite que el conocimiento sea compartido entre diferentes roles de un mismo proyecto.
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	<ul style="list-style-type: none"> - Permite que el conocimiento sea compartido entre personas con el mismo rol pero diferente seniority en el mismo proyecto. - Permite que el conocimiento sea compartido entre personas con el mismo rol pero diferente seniority en diferentes proyectos.
Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	<ul style="list-style-type: none"> - Permite que el conocimiento sobre diferentes tecnologías sea compartido entre miembros de la organización. - Permite que una persona experta en una tecnología pueda acceder a conocimiento relacionado con otra tecnología.
Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	<ul style="list-style-type: none"> - Permite acceder a conocimiento generado en proyectos anteriores. - Permite compartir conocimiento generado en proyectos anteriores

Tabla 30. Dimensiones e indicadores por medir. Elaboración propia.

5.4. Especificar el objetivo de la prueba

El objetivo de la prueba es adquirir la opinión de diferentes expertos relacionados con la GC y SF para validar cada una de las dimensiones e ítems especificados y de esta manera confirmar que el presente modelo de GC ayuda a resolver los desafíos que presenta la GC en SF definidos anteriormente.

5.5. Establecer los pesos diferenciales de las dimensiones de la prueba

En este caso, todas las dimensiones tendrán el mismo peso, ya que se busca ayudar a solucionar todos los desafíos de la GC en SF definidos anteriormente sin orden de importancia.

5.6. Diseño de plantilla

Se diseñó una plantilla de entrevista con 31 preguntas, una por cada dimensión a evaluar. Los entrevistados respondían del 1 al 5 (donde 1 es nada y 5 mucho) que tan de acuerdo estaban con el modelo propuesto en cuanto al indicador a evaluar. Además, por cada indicador se le preguntó a cada entrevistado sobre ideas de mejora en el modelo para abarcar de forma completa cada indicador en particular.

La plantilla de entrevista se puede ver en el anexo H.

5.7. Conclusiones del juicio de expertos

En la siguiente tabla (ver tabla 31) se muestran los resultados de las entrevistas a los expertos para cada indicador. Han sido coloreados en gris aquellos indicadores que tienen dos o más votos con posibilidades de mejoras sin mayores modificaciones al modelo (3 puntos).

Dimensiones	Indicadores	Andrei Rukavina	Sebastián Loprete	Ramiro Garbarini	Martín Riera	Marcelo Masci	Pablo Cigliuti	Rafael Natera
Cambios constantes en las tecnologías utilizadas.	Permite a los miembros de la organización mantenerse actualizados con las tecnologías a medida que sea necesario	5	4	3	4	3	5	5
Necesidad de buscar conocimiento fuera de la organización	Permite buscar conocimiento fuera de la organización	5	5	5	4	5	5	5
	Permite que el conocimiento adquirido desde fuera de la organización sea introducido en la misma	4	5	5	5	5	3	5
Contextos y procesos difieren en cada proyecto	Permite que el conocimiento sea compartido entre diferentes proyectos a pesar de que los contextos sean diferentes	5	5	4	5	4	5	5
	Permite que una persona que estaba en otro proyecto pueda adaptarse y adquirir el conocimiento del nuevo proyecto en el que participará	4	5	4	5	5	5	3
	Permite guardar información pertinente a cada proyecto (contexto, procesos, lecciones aprendidas...)	5	5	5	5	5	5	4
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	Permite a los miembros de la organización adquirir y compartir conocimiento del dominio de cada proyecto.	5	3	5	4	4	5	5
	Permite a los miembros de la organización adquirir y compartir conocimiento de temas generales de ingeniería de sistemas y desarrollo de software.	5	5	4	5	5	5	5
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	Permite que el conocimiento sobre convenciones de desarrollo de software como "buenas prácticas" sea adquirido por los miembros de la organización	5	4	5	5	5	5	4
	Permite que el conocimiento sobre convenciones de desarrollo de software como "buenas prácticas" sea compartido entre los miembros de la organización	4	4	5	5	5	4	5
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	Permite ubicar a las personas dentro de la organización por sus conocimientos en temas específicos.	5	4	4	5	5	5	5
	Permite contactar a las personas que tienen conocimiento sobre temas específicos para compartir conocimiento	5	4	4	5	5	5	5

El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	No afecta de forma considerable el tiempo destinado al desarrollo de software.	5	3	5	5	4	4	5
	Permite que el conocimiento sea formalizado a pesar de los tiempos reducidos en una SF.	5	4	5	5	4	4	5
	Motiva a las personas a formalizar el conocimiento.	5	4	4	5	5	5	5
Conocimiento altamente especializado, difícil de formalizar.	Facilita el proceso de formalización del conocimiento, al brindar herramientas y soporte en otros miembros de la organización.	5	5	5	5	5	5	4
Alta rotación del personal vuelve al conocimiento volátil.	Permite que el conocimiento quede distribuido entre los miembros de la organización y no en una sola persona	5	4	4	5	5	5	5
	Facilita el proceso de adquisición de conocimiento de nuevos miembros en la organización o en un proceso en particular	4	5	4	4	5	5	5
	Permite que el conocimiento sea formalizado y persistido en un medio de fácil acceso a los miembros de la organización	5	5	4	5	5	5	5
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.	Permite guardar información sobre lecciones aprendidas en un proyecto en particular.	5	5	5	4	5	5	4
	Permite compartir información sobre lecciones aprendidas en un proyecto entre los miembros de la organización	5	5	5	4	4	5	5
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.	Permite que el modelo pueda ser seguido de forma remota por personas en zonas geográficas u horarias diferentes.	5	5	5	3	4	5	5
	Permite que el conocimiento sea compartido entre miembros de la organización a pesar de las diferencias geográficas, horarias y culturales.	4	5	5	4	5	5	5
	Permite que miembros de la organización que estén en locaciones geográficas u horarias diferentes tengan acceso al conocimiento de la organización.	5	4	5	5	4	5	5
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.	Permite que el conocimiento sea compartido entre diferentes roles de un mismo proyecto.	5	5	5	4	5	5	5
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).	Permite que el conocimiento sea compartido entre personas con el mismo rol pero diferente seniority en el mismo proyecto.	5	5	5	4	5	5	5
	Permite que el conocimiento sea compartido entre personas con el mismo rol pero diferente seniority en diferentes proyectos.	5	4	4	4	4	5	5

Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)	Permite que el conocimiento sobre diferentes tecnologías sea compartido entre miembros de la organización.	5	4	5	4	5	5	5
	Permite que una persona experta en una tecnología pueda acceder a conocimiento relacionado con otra tecnología.	5	5	5	5	5	5	5
Las SF re-utilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	Permite acceder a conocimiento generado en proyectos anteriores.	5	4	5	5	5	5	5
	Permite compartir conocimiento generado en proyectos anteriores	5	4	4	5	5	5	5

Tabla 31. Resultados de las entrevistas con los expertos para cada indicador. Elaboración propia.

Como se puede observar en la tabla anterior, en todos los indicadores, más del 80% de los jueces han estado de acuerdo, por lo que se consideran como válidos bajo los criterios expuestos anteriormente.

Un solo indicador ha sido marcado como punto a mejorar, con dos calificaciones de 3 puntos, mientras que un total de 6 indicadores han tenido al menos un voto como posibilidad de mejora (3 puntos). Ninguno de los indicadores obtuvo menos de 3 puntos como calificación por ninguno de los expertos involucrados. Entonces según el juicio de expertos que indica que el 80% de los jueces deben estar de acuerdo para que un indicador se considere válido, todos los indicadores han sido considerados como válidos.

En la siguiente tabla (ver tabla 32) se muestran las posibilidades de mejora sugeridas por los expertos para cada indicador (hubo algunas sugerencias menores, con notas superiores a 3 puntos que también son tomadas en cuenta y listadas).

Dimensiones	Indicadores	Sugerencias de mejora.
Cambios constantes en las tecnologías utilizadas.	Permite a los miembros de la organización mantenerse actualizados con las tecnologías a medida que sea necesario	Han sido sugeridas dos oportunidades de mejora: <ul style="list-style-type: none"> - El modelo debería ser más proactivo en cuanto a la actualización del conocimiento en nuevas tecnologías. Darle un poco más de agilidad a este punto. - El modelo motiva la generación del conocimiento más no el consumo, podría haber una forma de controlar el consumo de información de la wiki y motivarlo.
Necesidad de buscar conocimiento fuera de la organización	Permite buscar conocimiento fuera de la organización	
	Permite que el conocimiento adquirido desde fuera de la organización sea introducido en la misma	Ha sido sugerida una oportunidad de mejora por 2 expertos: <ul style="list-style-type: none"> - Permite que el conocimiento sea introducido, sin embargo, la decisión de formalizar termina siendo subjetiva (a pesar de las preguntas) podrían agregarse más validaciones a esta parte.
Contextos y procesos difieren en cada proyecto	Permite que el conocimiento sea compartido entre diferentes proyectos a pesar de que los contextos sean diferentes	Han sido sugeridas dos oportunidades de mejora para este indicador: <ul style="list-style-type: none"> - El modelo podría tener en cuenta a las personas que están trabajando en las oficinas de los clientes (consultores externos), que no solo están en otro proyecto sino también en otra organización. - Podría haber una reunión periódica donde todos los proyectos expongan sobre temas relacionados al mismo (socialización entre proyectos)
	Permite que una persona que estaba en otro proyecto pueda adaptarse y adquirir el conocimiento del nuevo proyecto en el que participará	Ha sido sugerida una oportunidad de mejora para este indicador: <ul style="list-style-type: none"> - A pesar de que da la base inicial para que la persona se integre al proyecto, no hay un proceso de transferencia de conocimiento definido, podría definirse alguno.
	Permite guardar información pertinente a cada proyecto (contexto, procesos, lecciones aprendidas...)	

Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.	Permite a los miembros de la organización adquirir y compartir conocimiento del dominio de cada proyecto.	Tres sugerencias de mejoras han sido propuestas para este punto: <ul style="list-style-type: none"> - Incluir en el directorio a clientes clave para poder contactarlos y adquirir conocimientos del dominio del proyecto. - El modelo no obliga a incluir información del dominio de negocios, por lo que sería bueno obligar a que cada proyecto tengo información de este tipo en la wiki. - La formalización del conocimiento debería incluir otros formatos diferentes al escrito para agilizar el acceso al conocimiento.
	Permite a los miembros de la organización adquirir y compartir conocimiento de temas generales de ingeniería de sistemas y desarrollo de software.	Se ha propuesto una mejora en este indicador: <ul style="list-style-type: none"> - Puede que este tipo de conocimiento se lleve mal a la práctica, aunque este formalizado y claro, se recomienda validar la práctica del que consume el conocimiento.
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).	Permite que el conocimiento sobre convenciones de desarrollo de software como "buenas prácticas" sea adquirido por los miembros de la organización	
	Permite que el conocimiento sobre convenciones de desarrollo de software como "buenas prácticas" sea compartido entre los miembros de la organización	Una sugerencia de mejora ha sido propuesta para este indicador: <ul style="list-style-type: none"> - Las buenas prácticas pueden variar de un proyecto a otro, se recomienda que haya información de buenas prácticas del proyecto atada al perfil del proyecto en la wiki.
Es necesario saber "Quién sabe qué cosa" para reducir tiempos	Permite ubicar a las personas dentro de la organización por sus conocimientos en temas específicos.	
	Permite contactar a las personas que tienen conocimiento sobre temas específicos para compartir conocimiento	

El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.	No afecta de forma considerable el tiempo destinado al desarrollo de software.	Dos expertos coincidieron en este punto en que, si bien el modelo no afecta de forma significativa el proceso de desarrollo, se debe seguir buscando alternativas para que el impacto sea menor. Ambos expertos estuvieron de acuerdo en que reducir el impacto es complicado y que el modelo presenta muchos más beneficios que hacen que valga la pena aplicarlo.
	Permite que el conocimiento sea formalizado a pesar de los tiempos reducidos en una SF.	
	Motiva a las personas a formalizar el conocimiento.	Una propuesta de mejora ha sido sugerida para este indicador: <ul style="list-style-type: none"> - Las motivaciones para formalizar conocimiento pueden variar de una persona a otra por lo que se recomienda que exista más de una opción de motivación
Conocimiento altamente especializado, difícil de formalizar.	Facilita el proceso de formalización del conocimiento, al brindar herramientas y soporte en otros miembros de la organización.	Dos propuestas de mejora ha sido sugerida para este indicador: <ul style="list-style-type: none"> - Si bien el indicador es válido, si existe un estándar en cuanto a los formatos de los artículos de la wiki el proceso de formalización sería más sencillo. - Identificar los skills de los potenciales beneficiarios para que el artículo lleve la información necesaria para ser bien aprovechada.
Alta rotación del personal vuelve al conocimiento volátil.	Permite que el conocimiento quede distribuido entre los miembros de la organización y no en una sola persona	
	Facilita el proceso de adquisición de conocimiento de nuevos miembros en la organización o en un proceso en particular	Dos propuestas de mejora han sido sugeridas en este punto: <ul style="list-style-type: none"> - El comité de GC debería decidir qué conocimientos deben formar parte del proceso de inducción. - Las charlas informativas podrían registrarse en video, y ser parte del material de inducción.
	Permite que el conocimiento sea formalizado y persistido en un medio de fácil acceso a los miembros de la organización	Una propuesta de mejora ha sido sugerida en este punto: <ul style="list-style-type: none"> - El comité de GC debería establecer plantillas para regular el formato de los artículos de la wiki.

<p>Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.</p>	<p>Permite guardar información sobre lecciones aprendidas en un proyecto en particular.</p>	<p>Dos propuestas de mejora han sido sugeridas para este indicador:</p> <ul style="list-style-type: none"> - Si bien el modelo permite que esto ocurra, debería garantizar de alguna manera que las lecciones aprendidas existan, y se formalicen en el perfil de cada proyecto. - Las lecciones aprendidas a incluir en la base de conocimientos deberían ser aquellas genéricas que podrían aplicar a otros proyectos, pues hay casos aprendidos donde las lecciones son muy específicas.
	<p>Permite compartir información sobre lecciones aprendidas en un proyecto entre los miembros de la organización</p>	<p>Una propuesta de mejora ha sido sugerida para este indicador:</p> <ul style="list-style-type: none"> - Una reunión periódica en donde todos los proyectos que terminaron expliquen sus lecciones aprendidas va a ayudar a compartir de forma social este conocimiento.
<p>En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.</p>	<p>Permite que el modelo pueda ser seguido de forma remota por personas en zonas geográficas u horarias diferentes.</p>	<p>Una propuesta de mejora ha sido sugerida para este indicador:</p> <ul style="list-style-type: none"> - Colocar la zona horaria y el país de cada experto en el directorio va a permitir contactar a dicho experto en horario adecuado.
	<p>Permite que el conocimiento sea compartido entre miembros de la organización a pesar de las diferencias geográficas, horarias y culturales.</p>	<p>Tres propuestas de mejora han sido sugeridas para este indicador:</p> <ul style="list-style-type: none"> - Colocar la zona horaria y el país de cada experto en el directorio va a permitir contactar a dicho experto en horario adecuado. - Generar un glosario de términos importantes y sus equivalentes en otros países en la wiki, de manera el conocimiento podría estar más globalizado - La wiki debe estar escrita en un idioma en común (ingles si hay más de un idioma en la organización).
	<p>Permite que miembros de la organización que estén en locaciones geográficas u horarias diferentes tengan acceso al conocimiento de la organización.</p>	<p>Una propuesta de mejora ha sido sugerida para este indicador:</p> <ul style="list-style-type: none"> - Colocar la zona horaria y el país de cada experto en el directorio va a permitir contactar a dicho experto en horario adecuado. -

<p>En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.</p>	<p>Permite que el conocimiento sea compartido entre diferentes roles de un mismo proyecto.</p>	<p>Una propuesta de mejora ha sido sugerida en este indicador:</p> <ul style="list-style-type: none"> - Se debe limitar el alcance del conocimiento a personas con el perfil adecuado pues tratar de hacer entendible algo muy técnico puede generar sobreesfuerzo. Una forma es definiendo los conocimientos preexistentes necesarios para entender un artículo en específico.
<p>Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).</p>	<p>Permite que el conocimiento sea compartido entre personas con el mismo rol, pero diferente seniority en el mismo proyecto.</p>	<p>Una propuesta de mejora ha sido sugerida para este indicador:</p> <ul style="list-style-type: none"> - Reuniones periódicas donde los proyectos compartan información relevante sobre su trabajo o tecnologías ayuda también a potenciar este punto.
	<p>Permite que el conocimiento sea compartido entre personas con el mismo rol pero diferente seniority en diferentes proyectos.</p>	
<p>Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)</p>	<p>Permite que el conocimiento sobre diferentes tecnologías sea compartido entre miembros de la organización.</p>	
	<p>Permite que una persona experta en una tecnología pueda acceder a conocimiento relacionado con otra tecnología.</p>	

Las SF reutilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.	Permite acceder a conocimiento generado en proyectos anteriores.	Dos propuestas de mejora han sido sugeridas para este indicador: <ul style="list-style-type: none"> - Reuniones periódicas donde los proyectos compartan información relevante sobre su trabajo o tecnologías ayuda también a potenciar este punto. - Todos los elementos que se lleven a producción deben tener documentadas en la wiki sus componentes reutilizables (quizá bajo un tag de reutilización)
	Permite compartir conocimiento generado en proyectos anteriores	Una propuesta de mejora ha sido sugerida para este indicador: <ul style="list-style-type: none"> - Establecer tags genéricos que son estándares en el mercado para poder conseguir el conocimiento en cuanto a reutilización se refiere.

Tabla 32. Posibilidades de mejoras sugeridas por los expertos. Elaboración propia.

Sobre esta lista se puede destacar que hubo coincidencias entre las oportunidades de mejoras en los expertos (a pesar de no haberse tratado de la misma entrevista en el mismo lugar y hora). Se pueden identificar entonces los siguientes puntos de mejora a lo largo del modelo.

Mejoras en la Wiki:

- Podrían incluirse, además del texto, otros formatos de contenido en la wiki para agilizar el proceso de formalización y consumo del conocimiento. por ejemplo: audio o video.
- Al inicio de cada artículo de la wiki, especificar los conocimientos previos necesarios para consumir el conocimiento de este.

- Como las buenas prácticas pueden estar atadas a cada proyecto, los perfiles de los proyectos en la wiki deberían incluir información de buenas prácticas, además deben incluir información de solamente las lecciones aprendidas genéricas, es decir, que puedan ser útiles en otros proyectos. Los perfiles de proyectos también deben incluir información clave del dominio de negocio.
- Establecer estándares de formato de los artículos incluidos en la wiki, para hacerlos más legibles y fáciles de comprender.
- Registrar las charlas informativas en video y volverlas parte de la wiki, o ponerlas disponibles en alguna fuente de fácil acceso.
- La wiki debe ser escrita en un idioma en común entre los miembros de la organización, por ejemplo: inglés.
- Los elementos reutilizables puestos en producción por un proyecto en específico deben estar documentados en la wiki y referenciados con links en el perfil del proyecto. En estos casos se deben utilizar tags genéricos para poder ser conseguidos fácilmente.
- Agregar a la wiki un glosario de términos claves junto a su equivalente en distintos países donde la organización esté presente para facilitar el proceso de interiorización del conocimiento.

Mejoras en el directorio:

- Colocar en el perfil de cada experto, la ubicación geográfica y zona horaria (para facilitar el contacto).
- Agregar al directorio no solo expertos de la organización sino a clientes clave que puedan brindar información relevante cuando sea necesario.

Mejoras generales al modelo:

- Buscar la forma de que el proceso de introducir conocimiento de nuevas tecnologías sea más proactivo (por ejemplo: motivar esto).
- A pesar de que el modelo indica a buscar primero conocimiento en la wiki, se debe intentar motivar no solo la generación del conocimiento sino el consumo de este.

- El modelo podría tener en cuenta a aquellas personas que no trabajan directamente en las instalaciones de la organización sino en un cliente, ya que estas personas por lo general se encuentran más aisladas.
- Para extender el proceso de socialización y fortalecer el compartir conocimientos, se recomienda que también exista una charla periódica, definida por el comité de GC, en donde cada proyecto comparta con los demás novedades, nuevos conocimientos adquiridos y lecciones aprendidas.
- Como las motivaciones son muy diferentes de una persona a otra, se puede recomendar a las organizaciones ofrecer más de una forma de motivación para generar conocimiento.
- Sugerir a la organización definir un proceso detallado de transferencia de conocimiento, incorporando a la wiki, para aplicar cuando una persona entra a la organización o cambia de proyecto.
- A pesar de que el modelo no tiene un impacto importante en el proceso de desarrollo de software, los tiempos en las fábricas de software son cada vez más reducidos, por eso se recomienda buscar alternativas para reducir aún más el impacto.

6. CONCLUSIONES

6.1. Aportaciones de la Tesis

Como se menciona anteriormente la Gestión del Conocimiento les permite a las organizaciones aprovechar de forma significativa las ventajas competitivas que se derivan del conocimiento, cosa que no es posible si no se administra este activo de forma adecuada.

La ingeniería de software, en especial el rubro de las fábricas de software presenta desafíos que hacen que gestionar el conocimiento sea complicado y que los modelos de gestión del conocimiento ya existentes no sean suficientes para lidiar con los mismos.

Por medio de investigación documental y cruzando esta información con entrevistas a los expertos que validaron el modelo propuesto, se determinan 15 desafíos que hacen que la GC sea un proceso diferente en las fábricas de software. Estos desafíos son los siguientes (ver tabla 33):

Cambios constantes en las tecnologías utilizadas.
Necesidad de buscar conocimiento fuera de la organización
Contextos y procesos difieren en cada proyecto
Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.
Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).
Es necesario saber “Quién sabe qué cosa” para reducir tiempos
El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.
Conocimiento altamente especializado, difícil de formalizar.
Alta rotación del personal vuelve al conocimiento volátil.
Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.
En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socioculturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.
En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.
Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).
Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)
Las SF re-utilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.

Tabla 33. Desafíos de la GC en las Fábricas de Software.

Existen diversos modelos de GC que han sido creados para administrar el conocimiento de forma adecuada y aprovechar las ventajas competitivas derivadas del mismo.

Durante este trabajo de investigación no solo se determinaron y listaron los desafíos que hacen que la Gestión del Conocimiento en fábricas de software sea distinta a otros rubros, sino que se analizaron los principales modelos existentes en función de dichos desafíos y se determinó que ninguno de los modelos de GC los toma en cuenta a todos, además la alta rotación del personal no es tratada por ninguno de los modelos analizados (ver tabla 18).

Los modelos de GC analizados durante esta investigación fueron los siguientes:

- Earl
- Iseco
- CEN
- Bovea y García
- Wiig
- Sveiby
- Kerschberg
- Nonaka – Takeuchi
- Mc. Elroy
- Bustelo y Amarilla

Debido a la presencia de este problema, se decide entonces elaborar un modelo de gestión del conocimiento adaptado a fábricas de software y que tome en cuenta los desafíos identificados.

El modelo propuesto consta de 5 fases cíclicas que aplicarán los miembros de cada proyecto, más dos fases apartes que deben ser llevadas a cabo por personas designadas por la organización.

El modelo está pensado para ser compatible con metodologías de desarrollo tanto derivadas del manifiesto ágil (ya que las reuniones que propone pueden ser una prolongación de determinadas reuniones o ceremonias que forman parte de este tipo de metodologías) como tradicionales.

Este modelo tiene como base actividades y/o etapas propuestas en los modelos de EARL, Mc. Elroy ISECO y CEN.

Al igual que el modelo de Mc. Elroy, este modelo comienza con la detección de una necesidad de conocimiento y termina con el conocimiento formalizado en la base de conocimientos, sin embargo, no solo queda en la base de conocimientos, sino también interiorizado en otros miembros de la organización (esto reduce la volatilidad del conocimiento debido a la alta rotación de personal).

La figura 8 muestra la estructura general de la propuesta.

Este modelo se divide en 6 fases principales que involucran a un equipo de desarrollo de software y se realizan de forma cíclica y continua (ver punto 4.1.4). Estas fases son:

- Detectar una necesidad de conocimiento: esta etapa inicia el ciclo del modelo, se da cuando una persona del equipo detecta que existe una necesidad de conocimiento importante que puede ser un bloqueo para realizar su trabajo o para el trabajo de otros compañeros del equipo. La persona debe responder entonces una serie de preguntas para determinar la importancia del conocimiento necesitado y saber así si se debe o no continuar con la aplicación del modelo para esta necesidad.
- Buscar el conocimiento: la persona debe recurrir a las fuentes internas de conocimiento pertenecientes a la organización (base de conocimientos y directorio en primera instancia, descritos a lo largo de este capítulo) para buscar el conocimiento necesitado, si no lo consigue de esta manera, entonces debe recurrir a fuentes externas. En esta instancia también se debe responder una serie de preguntas para determinar la importancia de formalizar el conocimiento, de ser así, se le debe notificar al líder del proyecto.
- Organizar e interiorizar: la persona debe una vez encontrado el conocimiento, organizar la información adquirida e interiorizarla. Si el conocimiento no está formalizado en la wiki y tiene dudas de que este nuevo conocimiento esté correcto en primera instancia antes de continuar con su trabajo, debe consultarlo con líderes y otros miembros de la organización (validación en primera instancia).
- Utilizar el conocimiento: una vez validado en primera instancia, se debe utilizar entonces el conocimiento para seguir realizando sus tareas, tomando nota de los puntos más importantes de lo aprendido, para poder llevar a cabo las siguientes etapas de este modelo cuando sea el momento.
- Compartir y validar: la idea principal de esta etapa, es que durante una ceremonia periódica los miembros del equipo se reúnan para compartir y validar los conocimientos generados y aplicados hasta el momento (Si se utiliza scrum se recomienda hacerlo en la reunión retrospectiva, si no se utilizan metodologías ágiles, debe definirse una reunión periódica a conveniencia del equipo). Debe existir un calendario donde se detalle la fecha y hora

tentativas de cada una de estas reuniones periódicas. En esta reunión, cada persona compartirá lo aprendido y describirá forma en que se utilizó. Los demás miembros del equipo podrán intervenir para darle forma al conocimiento y añadir ideas de ser necesario. Así mismo, se validará definitivamente este conocimiento entre todos los miembros del equipo y se determinará la importancia de formalizarlo. Si el conocimiento agregado es considerado como muy importante por parte de los miembros del equipo, se organizará una charla para compartirlo con el resto de la organización.

- Formalizar: una vez compartido y validado el conocimiento por los miembros del equipo, el conocimiento debe ser formalizado, es decir, llevado a la wiki de conocimientos. Debe ser clasificado debidamente según las clasificaciones establecidas en la wiki de conocimiento y validado por el líder del equipo previo a su publicación oficial. Así mismo, el directorio debe ser actualizado para que ahora refleje que la persona que generó el conocimiento sabe del tema. Siguiendo las fases de este modelo, al haber sido el conocimiento previamente compartido con el resto del equipo, se vuelve más fácil de formalizar puesto que ya ha sido externalizado y los miembros del equipo pueden colaborar con este proceso.

Además, existen otras dos fases que son paralelas a las ya mencionadas y son llevadas a cabo por personas designadas especialmente para esto por la organización, estas fases son (ver punto 4.1.3):

- Fase de preparación de la organización para la aplicación del modelo: durante esta fase se designará el comité de GC de la organización (encargado de llevar a cabo la implementación y control), se determinará qué elementos conforman la base de conocimientos y se buscará ponerlos a disposición de todos los miembros de la organización, se diseñará y creará la wiki de conocimiento y el directorio.
- Fase de mantenimiento del modelo: el modelo debe llevar un mantenimiento y control periódico para que su aplicación pueda ser prolongada en el tiempo. Durante esta fase el comité de gestión del conocimiento se encargará de hacer back up de la base de conocimientos, pedir *feedback* para aplicar mejoras continuas a la implementación del modelo y se llevará a cabo el reconocimiento a los creadores de conocimiento.

Existen dos elementos fundamentales para que la aplicación del modelo sea posible, estos son: La wiki de conocimientos y el directorio.

La wiki de conocimientos es un sistema interno (web, aplicación de escritorio o aplicación móvil), en donde se centralizará en conocimiento que va siendo formalizado, en forma de artículos o tutoriales. Puede formar parte de la web interna de la organización. En ella debe haber un espacio para cada proyecto y la información debe estar organizada con palabras claves y tags. El directorio es otra herramienta que puede facilitar el proceso de compartir conocimiento dentro de la organización. Un directorio contiene información básica de contacto y conocimientos de todos los miembros de la organización y puede permitir que un miembro de esta consiga a otros que tengan información relacionada con sus dudas y pueda contactarlos para solventarlas. Las características de la wiki y el directorio están detalladas en el punto 4.1.3.

El modelo propuesto es analizado en base a los desafíos de gestión del conocimiento en las fábricas de software que fueron descriptos anteriormente, los resultados de este análisis se pueden observar en la tabla 20.

Así mismo, el modelo es validado utilizando el método de “Juicio de expertos”, mediante entrevistas realizadas a 7 expertos previamente seleccionados según las instrucciones de dicho método.

Cada uno de los ítems o dimensiones a evaluar son confirmadas como válidas por los expertos (ver tabla 31), algunos ítems con posibilidades de mejoras sugeridas (ver tabla 32).

Sin embargo, la investigación debe continuar y las principales líneas futuras se listan en el punto siguiente (ver punto 6.2).

6.2. Futuras Líneas de Investigación

Este proyecto no solo deja líneas abiertas de investigación relacionadas directamente con el modelo, sino también con el área de gestión del conocimiento en fábricas de software. Es importante mencionar que la investigación de aplicación de este modelo sigue abierta y en desarrollo por el grupo

GEMIS de la Universidad Tecnológica Nacional, de hecho, se está planeando en este momento la aplicación en una pequeña fábrica de software del Área Metropolitana de Buenos Aires.

A continuación, se listan una serie de ítems que pueden ser evaluados en futuras oportunidades.

- Agregar al modelo las mejoras sugeridas por los expertos y listadas al final del punto 5.8. antes de aplicarlo a una organización.
- Los 15 desafíos de gestión del conocimiento obtenidos mediante investigación documental y validados con los expertos que evaluaron el modelo, podrían cambiar con el tiempo e incluso ser diferentes según la localidad o país en donde se encuentre la fábrica de software, por lo que se recomienda una investigación más detallada de los mismos dependiendo de la localidad.
- Aplicar este modelo en una fábrica de software pequeña o mediana para evaluar posibles mejoras y beneficios otorgados a la misma (este punto ya está siendo llevado a cabo por el grupo GEMIS), y aplicar mejoras cíclicas al modelo.
- Si bien el modelo se hizo pensando en Fábricas de Software genéricas (es decir, en base a la documentación de los desafíos encontrados y la definición per se de lo que es una Fábrica de Software, sin tener en cuenta el tamaño de esta), los expertos seleccionados tienen mayor experiencia en pequeñas y medianas Fábricas de Software, por lo que se puede decir que el modelo está validado para este tamaño de organizaciones. Por este motivo y por el hecho de que las grandes organizaciones pueden tener otro tipo de problemas o desafíos relacionados con el tamaño de la organización, validar este modelo con expertos cuya experiencia sea en grandes Fábricas de Software, así como también aplicarlo en este tipo de organización, es una futura línea de investigación abierta.
- Una de las limitantes en cuanto a la evaluación de este modelo, fue la inexistencia de algún tipo de métricas que permitan evaluar el nivel o madurez de gestión del conocimiento en una fábrica de software. Definir métricas que permitan evaluar la gestión del conocimiento antes y después

de la aplicación de este u otros modelos de gestión del conocimiento en fábricas de software es una línea de investigación abierta.

- Otra línea de investigación abierta es evaluar qué tecnologías podrían ser mejores para llevar a cabo este modelo según el tamaño de la organización (por ejemplo: tecnologías recomendadas para el directorio, tecnologías recomendadas para la wiki).
- Buscar la forma de cuantificar los beneficios que le puede brindar este modelo a una organización, bien sea en ahorro de tiempo en el proceso de desarrollo de software o monetario.

7. REFERENCIAS BIBLIOGRÁFICAS

- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002) Agile software development methods: Review and analysis, VTT publication 478, Espoo, Finland, 107p
- Argimón J. (2004). "Métodos de Investigación Clínica y Epidemiológica". Elsevier España, S.A. ISBN 9788481747096.
- Aurum, A., Jeffery, R., Wohlin, C., Handzic, M. (Eds.). (2013). Managing software engineering knowledge. Springer Science y Business Media.
- Bhuiyan N, Baghel Amit (2005), "An overview of continuous improvement: from the past to the present", Management Decision, Vol. 43 Iss 5 pp. 761 – 771
- Bovea, T. L. C., & García, V. H. M. (2011). Modelo de gestión del conocimiento aplicado a un sistema complejo: Desarrollo de fábricas de software. Ninth LACCEI Latin American and Caribbean Conference. Colombia
- Braude, E. J., & Bernstein, M. E. (2016). *Software engineering: modern approaches*. Waveland Press.
- Briceño M., Bernal C. (2010). Estudios de caso sobre la gestión del conocimiento en cuatro organizaciones colombianas líderes en penetración de mercado. Estudios Gerenciales, Vol 26. Nro 117
- Bueno, E. (2000). Dirección del Conocimiento y Aprendizaje: Creación, distribución y mediación de Intangibles. Recuperado de <http://www.sedic.es/bueno.pdf>
- Bustelo Ruesta, C., & Amarilla Iglesias, R. (2001). Gestión del conocimiento y gestión de la información. Inforárea 2001; 8 (34): 226-230
- Cabero J, Llorente, M. (2013) La aplicación del Juicio de Experto Como Técnica de Evaluación De Las Tecnologías de la Información y Comunicación, Revista de Tecnología de Información y Comunicación en Educación • Volumen 7, N° 2 Julio-diciembre 2013.
- Canals, A., Boisot, M., & Cornella, A. (2003). *Gestión del conocimiento*. Gestión 2000.
- Cárdenas, M. L., & Rivera, J. F. (2004). La teoría de la complejidad y su influencia en la escuela. *Revista de teoría y didáctica de las Ciencias Sociales*, (9).
- CEN (2004) CEN CWA 14924 European GuidetogoodPractice in Knowledge Management - Part 1 to 5, Brussels.

- Choo, Chun Wei y Alvarenga Neto, R. C. D. (2010). "Beyond the ba: Managing enabling contexts in knowledge organizations". *Journal of Knowledge Management*
- Cox, M.T. (2005). Metacognition in computation: A selected research review. *Artif. Intell.*, 169, 104-141.
- Creswell, J. W. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
- Crnkovic Ivica, Larsson Magnus. (2000). A Case Study: Demands on Component-based Development. ACM. International Conference on Software Engineering: Proceedings of the 22nd international conference on Software engineering.
- Cruz, Y. R. (2018). Gestión de Información y del Conocimiento para la toma de decisiones organizacionales. *Bibliotecas. Anales de Investigación*, (11), 150-163.
- Daft, R. (2010). *Organization Theory and Design (10ª ed.)*. USA: South-Western, Cengage Learning
- Davenport, T. (1998). Knowledge Management and the broader firm: Strategy, advantage, and performance. En J. Liebowitz, *Knowledge Management Handbook*.
- Davenport, T. y Prusak, L. (1998). *Working Knowledge: How organizations manage what they know*. Cambridge, MA: Harvard Business School Press.
- Díaz, M. T. R., & Millán, J. J. G. (2013). Gestión del Conocimiento y Capital Intelectual, a través de modelos universitarios. *Económicas CUC*.
- Dingsoyr, T., & Smite, D. (2014). Managing knowledge in global software development projects. *IT Professional*.
- Donald (2010). "Knowledge management as an ephemeral management fashion?". *Journal of Knowledge Management*.
- Drucker, P. F (1994). *Post-capitalist society*. Routledge.
- Duffy, Jan (2000). "Knowledge management: What every information professional should know". *Information Management Journal*.
- Earl, M. (2001). Knowledge management strategies: Toward a taxonomy. *Journal of Management Information Systems*, 18(1), 215-233.
- Escobar J, Cuervo A (2008). Validez de contenido y juicio de expertos: una aproximación a su utilización. *Avances en medición*, 6(1), 27-36.

- Evans M, Dalkir K and Bidian C. (2013) "A Holistic View of the Knowledge Life Cycle: The Knowledge Management Cycle (KMC) Model" *The Electronic Journal of Knowledge Management* Volume 12 Issue 2 (pp85-97)
- Ezponda, J. E. (2008). Transferencia de conocimiento entre comunidades científicas. *Arbor*, 184(731), 539-548.
- O'Dell, Carla y Grayson, C. Jackson (1998). "If only we knew what we know: Identification and transfer of internal best practices". *California Management Review*.
- Fernández, F. G., & Borjas, A. E. C. (2008). Los equipos de trabajo: una práctica basada en la gestión del conocimiento. *Visión gerencial*, (1), 45-58.
- Fernstrom, C., Narfelt, K. H., & Ohlsson, L. (1992). Software factory principles, architecture, and experiments. *IEEE Software*.
- Flores-Urbáez, M. y Peña-Cedillo, J. (2008). Gerencia del conocimiento y capacidades de innovación. Un estudio en laboratorios de investigación petrolera. Venezuela: Universidad del Zulia.
- Gao, Fei; Meng, Li y Clarke, Steve (2008). "Knowledge, management, and knowledge management in business operations". *Journal of Knowledge Management*
- Garvin, D. A. (1993). Building a learning organization. *Harvard business review*, 71(4), 78-91.
- Gergen, K. (2007). La ciencia psicológica en el contexto posmoderno. *Construccionismo Social: Aportes para el debate y la práctica*, 93-115.
- Gibbs, B., Hadley-Kershaw, E., Nerlich, B., Pearce, W., Salvadurai, H., Spencer, A., & Tsouvalis, J. (2013). The 2013 Conference Organising Committee.
- Gómez, D. R. (2006). Modelos para la creación y gestión del conocimiento: una aproximación teórica. *Educación* 37 (25 - 39).
- González-García, Alejandro y Parés-Ferrer, Marianela (2012). "Gestión del Conocimiento en Cuba: Diseminación de sus resultados de investigación, de 1997- 2010". *Ciencias de la Información*
- Greenfield, J., & Short, K. (2003). Software factories: assembling applications with patterns, models, frameworks and tools. In *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*. ACM.
- Greiner, Martina E.; Böhmman, Tilo y Krcmar, Helmut (2007). "A strategy for knowledge management". *Journal of Knowledge Management*.
- Guevara, J. C., Cavanzo, G. A., & Pérez, M. (2017). FRAMEWORK DE GESTIÓN DEL CONOCIMIENTO (FGC) BASADO EN CAPAS. *Visión Electrónica: algo más que un estado sólido*, 10(1).

- Heisig, P. (2009). Harmonisation of knowledge management—comparing 160 KM frameworks around the globe. *Journal of knowledge management*, 13(4), 4-31.
- Hernández, L. E. P. (2018). Gestión del conocimiento. *Letras Con* Ciencia Tecno* Lógica*, 91-102.
- Hernández, R., Fernández, C., y Baptista, M.P. (2010) Metodología de la Investigación (5ª Ed.). México: McGraw Hill Educación.
- Instituto de Ingeniería del Software (2000) *CMMISM for Systems Engineering/Software Engineering, Version 1.02, Staged Representation* (p.15). CMU/SEI-2000-TR-018 ESC-TR-2000-018, Carnegie Mellon University, Pittsburgh, PA.
- Jones, Kiku y Leonard, Lori N. K. (2009). "From tacit knowledge to organizational knowledge for successful KM". *Annals of Information Systems*
- Kerschberg, L. (2001, September). Knowledge management in heterogeneous data warehouse environments. In *International Conference on Data Warehousing and Knowledge Discovery* (pp. 1-10). Springer Berlin Heidelberg.
- Levison, Y. O., González, M. E., & Salguero, L. A. (2013). Propuesta para construcción de un modelo de gestión del conocimiento en una unidad de planificación y desarrollo de la upel. UPEL, Venezuela
- Manchanda, H., Agarwal, A., Bhati, D., & Ilango, P. (2017). Agile Methods for Software Development. *Imperial Journal of Interdisciplinary Research*, 3(6).
- Manucci, M. (2008). Impacto corporativo. Editorial La Crujía Ediciones. Buenos Aires, Argentina.
- Marshall, Chris; Prusak, Laurence y Shpilberg, David (1996). "Financial risk and the need for superior knowledge management". En: PRUSAK, Knowledge in organizations. Boston: Butterworth-Heinemann
- Maulini, A.; Straccia, L; Pollo-Cattaneo, M.F. (2018) Una aproximación a un modelo de gestión de conocimiento aplicable a las pequeñas y medianas fábricas de software. InNGENIO. ISBN 978-958-59127-9-3. Pág 53- 63.
- Maulini, A.; Straccia, L; Pollo-Cattaneo, M.F. (2019) Un modelo de gestión de conocimiento aplicable a las pequeñas y medianas fábricas de software. En "Desarrollo e Innovación en Ingeniería - Cuarta Edición" (Ed. Prof. Edgar Serna M). Capítulo 1, Pág. 5-16. Editorial Instituto Antioqueño de Investigación DOI: <http://doi.org/10.5281/zenodo.3387679>.
- McElroy, M. (2003), *The new knowledge management: complexity, learning and sustainable innovation*, Butterworth-Heinemann, Boston.

- Michalus, J. SaracheC, William A, Hernández P (2015), Método de expertos para la evaluación ex-ante de una solución organizativa. versión impresa ISSN 1668-8708
- Mowery, David C. (2010). "Alfred Chandler and knowledge management within the firm". *Industrial and Corporate Change*
- Nicholson, B., & Sahay, S. (2004). Embedded knowledge and offshore software development. *Information and organization*, 14(4), 329-365.
- Nomura, L., Spínola, M. M., Tonini, A. C., & Hikage, O. K. (2007). A model for defining software factory processes. In *19th International Conference on Production Research*.
- Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford university press
- Owen, John Mackenzie (1999). "Knowledge management and the information professional". *Information Services & Use*
- Pascual, Marcelo y SUNDARDAS, Arjan (2003). "El "mentoring" como transmisor del conocimiento tácito". *Harvard Deusto Business Review*
- Pawlowski, J. M., & Bick, M. (2015). The global knowledge management framework: Towards a theory for knowledge management in globally distributed settings. *Lead. Issues Knowl. Manag. Vol. Two*, 2, 134.
- Pedraja-Rejas, L. (2006). Sociedad del conocimiento y dirección estratégica: Una propuesta integradora. *Interciencia*, 31(8), 570-576.
- Pérez Montoro, M. (2004). Identificación y representación del conocimiento organizacional: la propuesta epistemológica clásica. *Documentos de proyecto DP04-001, Universidad Oberta de Cataluña*.
- Pérez, V. A., & Urbáez, M. F. (2016). Modelos teóricos de gestión del conocimiento: descriptores, conceptualizaciones y enfoques. *Entreciencias: diálogos en la Sociedad del Conocimiento*
- Pons, N. L., Pérez, Y. P., Stiven, E. R., & Quintero, L. P. (2014). Diseño de un modelo de Gestión del Conocimiento para mejorar el desarrollo de equipos de proyectos informáticos. *Revista Española de Documentación Científica*, 37(2), 044.

- Portal, S. G., & Jaramillo, M. D. J. M. (2017). Diseño de un data warehouse para medir el desarrollo disciplinar en instituciones académicas. *Investigación Bibliotecológica. Archivonomía, Bibliotecología e Información*, 31(72), 161-181.
- Ramos, C. A. (2015). Los paradigmas de la investigación científica. *Avances en Psicología*, 23(1), 9-17.
- Real Academia Española. (2014). *Diccionario de la lengua española* (23.ªed.). Consultado en <http://www.rae.es/rae.html>
- Reyes Meleán, C. F. (2005). Análisis de la relación entre la ingeniería del conocimiento y la gestión del conocimiento en base al modelo de Nonaka y Takeuchi.
- Ropain, E., Amaya, L., & Arroyo, E. (2014). FACTORES DE LA GESTIÓN DEL CONOCIMIENTO EN EL DESARROLLO DE SOFTWARE. Encuentro de Ciencias y Tecnología URBE 2014. Venezuela.
- Rodriguez, D. (2006). Modelos para la creación y gestión del conocimiento: una aproximación teórica. Recuperado de <http://educar.uab.cat/article/view/187/168>
- Rueda Martínez, M. I. (2014). La gestión del conocimiento y la ciencia de la información: relaciones disciplinares y profesionales. Tesis Doctoral, Universidad Carlos III, Getafe
- Rus, I., & Lindvall, M. (2002). Knowledge management in software engineering. *IEEE software*.
- Santos, S. C. D., & Soares, F. S. (2013). Authentic assessment in software engineering education based on PBL principles: a case study in the telecom market. In Proceedings of the 2013 International Conference on Software Engineering (pp. 1055-1062). IEEE Press
- Shongwe M. (2016) "An Analysis of Knowledge Management Lifecycle Frameworks: Towards a Unified Framework" *The Electronic Journal of Knowledge Management Volume 14 Issue 3(pp140-153)ISSN 1479-4411*
- Short, Jack Greenfield and Keith. (2004). *Software Factories-Assembling Applications with Patterns, Models, Frameworks, and Tools*. s.l. : Wiley Publishing.
- Son, L.K., & Terrace, H.S. (2009). Comparative metacognition. *Current opinion in neurobiology*, 19 1, 67-74.
- Straccia, L; Maulini, A; Pytel, P; Masci, M; Vegega, C; Pollo-Cattaneo, Ma. F (2017). *La Gestión del Conocimiento en Pequeñas y Medianas Fábricas de Software en el Área Metropolitana de Buenos Aires*. Proceedings XIX Workshop de Investigadores de Ciencias de la Computación, WICC 2017. Instituto Tecnológico de Buenos Aires ITBA. Pág.575-579. ISBN: 978-987-42-5143-5.
- Sveiby, K. (1997). *The New Organizational Wealth: Managing and Measuring Knowledge based Assets*. USA: San Francisco: Barrett-Kohler Publishers.

- Sveiby, K. E. (2001). A knowledge-based theory of the firm to guide in strategy formulation. *Journal of intellectual capital*, 2(4), 344-358.
- Swan, Jacky; Newell, Sue y Robertson, Maxine (2000). "Knowledge management: When will people management enter the debate?". SPRAGUE, Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS-33) (Maui, 4-7 de enero de 2000). Los Alamitos: Computer Society Press
- Todd, Ross J. (2001). "The smart school: Knowledge management working for your future [SCIS Oration]". Forging New Directions: Australian School Library Association Bienal.
- Uribe, G. C., Jiménez, S. L. R., & Jiménez, H. S. (2008). *Diseño de un modelo de gestión del conocimiento para la Escuela Interamericana de Bibliotecología*. Universidad de Antioquia.
- Wiig, K. (1993). Knowledge Management Foundations: Thinking about thinking – How people and organizations create, represent, and use knowledge. Arlington, TX: Schema.

ANEXO A

Calendario de reuniones periódicas para validar-compartir el conocimiento

CALENDARIO DE REUNIONES DE GESTIÓN DEL CONOCIMIENTO.

Nombre de la organización:

Nombre del equipo:

Nombre del proyecto:

Líder responsable del equipo:

Fecha	Hora	Lugar

ANEXO B**Calendario de charlas informativas de gestión del conocimiento****CALENDARIO DE CHARLAS INFORMATIVAS.****Nombre de la organización:****Fecha de inicio del ciclo de charlas:****Fecha fin del ciclo de charlas:**

Fecha	Hora	Lugar	Proyecto	Ponentes (Nombre y apellido)	Título de la charla

ANEXO C

Planilla de solicitud de charla informativa.

SOLICITUD DE APROBACIÓN DE CHARLA INFORMATIVA.

Nombre del proyecto:

Líder responsable del proyecto:

Nombre de los ponentes (nombre, apellido, rol actual):

Fecha tentativa:

Título de la charla:

Resumen:

Breve explicación de la importancia de dar esta charla en la organización:

ANEXO D

Planilla de resultado de solicitud de charla informativa.

RESULTADO DE SOLICITUD DE CHARLA INFORMATIVA

Nombre del Evaluador:

Fecha de revisión de la solicitud:

Proyecto:

Líder responsable:

Tema:

Resultado: Aprobado / Rechazado

Fecha y hora de la charla: (En caso de ser aprobado)

Motivo del rechazo: (En caso de ser rechazado)

Comentarios:

ANEXO E

Planilla de solicitud de compartir-validar conocimiento en la reunión periódica de GC.

SOLICITUD PARA COMPARTIR-VALIDAR CONOCIMIENTO

Fecha:

Nombre:

Rol:

Tema:

Fuente (s) de donde se obtuvo el conocimiento:

Breve explicación de importancia de compartirlo:

ANEXO F

Lista de temas a tratar en la siguiente reunión periódica de GC.

LISTA DE TEMAS PARA REUNIÓN DE COMPARTIR-VALIDAR CONOCIMIENTO

Nombre del proyecto:

Líder del proyecto:

Fecha de la reunión:

Nombre de la persona	Rol	Tema para discutir

ANEXO G

Minuta de reunión de compartir-validar conocimiento.

MINUTA DE REUNIÓN DE COMPARTIR-VALIDAR CONOCIMIENTO

Fecha:

Lista de asistentes:

Para cada tema discutido, se debe llenar lo siguiente:

Tema:

Comentarios importantes:

¿Va a ser formalizado en la Wiki? SI/NO

¿Se va a hacer la solicitud de charla informativa? – Fecha tentativa.

ANEXO H**Plantilla de las entrevistas realizadas a los expertos**

Del 1 al 5, en donde 1 es “nada” y 5 es “mucho” que tanto cree que el modelo contribuye con indicadores:

1- Cambios constantes en las tecnologías utilizadas.

- Permite a los miembros de la organización mantenerse actualizados con las tecnologías a medida que sea necesario _____
¿Qué cree que podría mejorarse en relación con este aspecto?

2- Necesidad de buscar conocimiento fuera de la organización.

- Permite buscar conocimiento fuera de la organización. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
- Permite que el conocimiento adquirido desde fuera de la organización sea introducido en la misma. ____
¿Qué cree que podría mejorarse en relación con este aspecto?

3- Contextos y procesos difieren en cada proyecto

- Permite que el conocimiento sea compartido entre diferentes proyectos a pesar de que los contextos sean diferentes. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
- Permite que una persona que estaba en otro proyecto pueda adaptarse y adquirir el conocimiento del nuevo proyecto en el que participará. ____
¿Qué cree que podría mejorarse en relación con este aspecto?
- Permite guardar información pertinente a cada proyecto (contexto, procesos, lecciones aprendidas...) _____
¿Qué cree que podría mejorarse en relación con este aspecto?

4- Se necesita conocimiento sobre el dominio propio de la ingeniería de software y sobre el dominio para el cual se crea el producto.

- Permite a los miembros de la organización adquirir y compartir conocimiento del dominio de cada proyecto. _____
¿Qué cree que podría mejorarse en relación con este aspecto?

- Permite a los miembros de la organización adquirir y compartir conocimiento de temas generales de ingeniería de sistemas y desarrollo de software. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
- 5- Se necesita conocimiento muy específico sobre convenciones locales de desarrollo de software (es normalmente informal).**
- Permite que el conocimiento sobre convenciones de desarrollo de software como “buenas prácticas” sea adquirido por los miembros de la organización. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
 - Permite que el conocimiento sobre convenciones de desarrollo de software como “buenas prácticas” sea compartido entre los miembros de la organización. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
- 6- Es necesario saber “Quién sabe qué cosa” para reducir tiempos**
- Permite ubicar a las personas dentro de la organización por sus conocimientos en temas específicos. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
 - Permite contactar a las personas que tienen conocimiento sobre temas específicos para compartir conocimiento. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
- 7- El conocimiento es mayormente implícito y no se vuelve explícito por falta de tiempo. Los tiempos en las SF son muy reducidos.**
- No afecta de forma considerable el tiempo destinado al desarrollo de software. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
 - Permite que el conocimiento sea formalizado a pesar de los tiempos reducidos en una SF. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
 - Motiva a las personas a formalizar el conocimiento. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
- 8- Conocimiento altamente especializado, difícil de formalizar.**
- Facilita el proceso de formalización del conocimiento, al brindar herramientas y soporte en otros miembros de la organización. _____

¿Qué cree que podría mejorarse en relación con este aspecto?

9- Alta rotación del personal vuelve al conocimiento volátil.

- Permite que el conocimiento quede distribuido entre los miembros de la organización y no en una sola persona. _____

¿Qué cree que podría mejorarse en relación con este aspecto?

- Facilita el proceso de adquisición de conocimiento de nuevos miembros en la organización o en un proceso en particular. _____

¿Qué cree que podría mejorarse en relación con este aspecto?

- Permite que el conocimiento sea formalizado y persistido en un medio de fácil acceso a los miembros de la organización. _____

¿Qué cree que podría mejorarse en relación con este aspecto?

10- Altas posibilidades de fracaso o cancelación de un proyecto hacen necesario almacenar conocimiento sobre razones del fracaso para no volver a repetirlo.

- Permite guardar información sobre lecciones aprendidas en un proyecto en particular. _____

¿Qué cree que podría mejorarse en relación con este aspecto?

- Permite compartir información sobre lecciones aprendidas en un proyecto entre los miembros de la organización. _____

¿Qué cree que podría mejorarse en relación con este aspecto?

11- En equipos globales las distancias temporales y geográficas afectan la habilidad para acceder y compartir el conocimiento, mientras que las distancias socio-culturales introducen desafíos para unificar las formas en que el conocimiento se comparte y se mantiene.

- Permite que el modelo pueda ser seguido de forma remota por personas en zonas geográficas u horarias diferentes. _____

¿Qué cree que podría mejorarse en relación con este aspecto?

- Permite que el conocimiento sea compartido entre miembros de la organización a pesar de las diferencias geográficas, horarias y culturales. _____

¿Qué cree que podría mejorarse en relación con este aspecto?

- Permite que miembros de la organización que estén en locaciones geográficas u horarias diferentes tengan acceso al conocimiento de la organización. _____
¿Qué cree que podría mejorarse en relación con este aspecto?

12- En un mismo proyecto de software, existen diferentes roles, con conocimientos y experticias diferentes, y el conocimiento debe ser accesible y entendible por todos los roles.

- Permite que el conocimiento sea compartido entre diferentes roles de un mismo proyecto. _____
¿Qué cree que podría mejorarse en relación con este aspecto?

13- Pueden existir dentro de un mismo rol diferentes niveles de experticia (seniority).

- Permite que el conocimiento sea compartido entre personas con el mismo rol pero diferente seniority en el mismo proyecto. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
- Permite que el conocimiento sea compartido entre personas con el mismo rol pero diferente seniority en diferentes proyectos. _____
¿Qué cree que podría mejorarse en relación con este aspecto?

14- Pueden existir dentro de un mismo rol, experticias en diferentes tecnologías (ejemplo: los desarrolladores de un mismo proyecto pueden estar especializados en tecnologías diferentes)

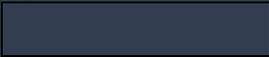
- Permite que el conocimiento sobre diferentes tecnologías sea compartido entre miembros de la organización. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
- Permite que una persona experta en una tecnología pueda acceder a conocimiento relacionado con otra tecnología. _____
¿Qué cree que podría mejorarse en relación con este aspecto?

15- Las SF re-utilizan código y productos o partes de los mismos, por lo cual debe existir conocimiento acerca de desarrollos anteriores y repositorios, y además ser compartido.

- Permite acceder a conocimiento generado en proyectos anteriores. _____
¿Qué cree que podría mejorarse en relación con este aspecto?
- Permite compartir conocimiento generado en proyectos anteriores. _____
¿Qué cree que podría mejorarse en relación con este aspecto?

ANEXO I

C.V. Andrei Rukavina

Contactar
 Ciudad Autónoma de Buenos Aires, Argentina

www.linkedin.com/in/andreirukavina (LinkedIn)

Aptitudes principales
 Data Science
 Data Mining
 Predictive Modeling

Languages
 Español (Native or Bilingual)
 Inglés (Full Professional)
 Italiano (Elementary)

Certifications
 Hyperion Essbase 7.1.2 Certified Consultant
 Hyperion Planning 9.3 Administrator I & II Certified Expert
 Certificación Quiterian Dynamic Data Web: Administración, Análisis y ETL (Herramienta de análisis dinámico)
 Examination for the Certificate of Competency in English (ECCE)

Honors-Awards
 Achievements in Excellence (Individual Award)
 1st Annual IMS Challenge
 Honorable Mention for Detailed Discovery (Where's Rachel?)
 Iniciativa y trabajo en equipo
 Mejor promedio: Universidad Tecnológica Nacional

Andrei Rukavina

Director - Applied Analytics

Extracto

Me considero una persona proactiva, me gustan los desafíos y siempre busco superarme.

Tengo un amplio perfil en TI compuesto por experiencia en:

- 1) Applied Analytics y Data Mining
- 2) Planificación y Gestión de proyectos.
- 3) Construcción de Data Marts y Data Warehouses
- 4) Desarrollo y Automatización de procesos ETL.
- 5) Desarrollo e implementación de aplicaciones web.
- 6) Automatización de flujo de datos sistemas.

Mi objetivo es el perfeccionamiento de un perfil polifacético con base en minería de datos y análisis predictivo.

Experiencia

AlixPartners

5 años 5 meses

Director - Digital

diciembre de 2015 - Present

Buenos Aires

- Ensure comprehensiveness of data collection post client interviews
- Understand which analytical solution to use to best solve client business problem and ensure alignment of analytical solutions with client issues
- Validate all model results and ensure applicability to client requirements
- Develop deep expertise in technical area(s): clustering, predictive model, or optimization
- Develop and guide analytical outputs integration into client's environment/ business
- Provide thought leadership for analytical plan and execution
- Understand business drivers and best mechanism for presenting this information to enable client decision making
- Develop the implementation framework for report / visualization and determine best delivery mechanism (Excel, SQL, Tableau, D3)

Page 1 of 6

Develop materials and present findings to senior executives

Vicepresidente – Information Management Services
septiembre de 2013 - noviembre de 2015 (2 años 3 meses)
Argentina

1) Modelado Predictivo

- * Construcción de modelos predictivos de alto rendimiento que generalicen bien nuevos datos
- * Programación en lenguajes de aprendizaje estadístico, como R, scikit-learn, Matlab o SAS
- * Aplicar los siguientes algoritmos, con un enfoque directo en el aumento de los ingresos y costos decrecientes: regresión, random forest, modelos generalizados con boosting, modelos aditivos generalizados, support vector machine, redes neuronales, y de series de tiempo.

2) Data Munging

- * Recolección de datos de una amplia variedad de bases de datos corporativas, incluyendo varias bases de datos SQL (Microsoft, Oracle, Netezza, etc), bases de datos no-SQL, bases de datos Access y archivos de Excel
- * Analizar los datos de los documentos HTML y XML válido mal estructurados
- * Utilización de expresiones regulares para extraer información de documentos de texto sin estructurar
- * Tratamiento de datos faltantes a través de imputación múltiple o el uso de modelos avanzados (por ejemplo, GBM en R)
- * Construcción de nuevas e interesantes características que mejoran la precisión del modelo
- * Automatización de tareas con Python, R, o secuencias de comandos Linux

3) Visualización de Datos

- * Exposición frente a ejecutivos de nivel-C
- * Presentar los hallazgos técnicos a un público no técnico
- * Uso de herramientas de visualización como: Tableau, QlikView, o ggplot2

SAS

ETL & Data Mining Engineer
julio de 2012 - septiembre de 2013 (1 año 3 meses)
Buenos Aires

Implementación, desarrollo y mantenimiento de diversas soluciones SAS

Coordinación del equipo de desarrollo.

Implementación de buenas prácticas y metodologías ágiles para asegurar calidad y mantenibilidad.

SAS Base 9.2

SAS Data Integrator 5.4

Unix

Consultor IT independiente

Instructor & Consultor en ETL

agosto de 2009 - septiembre de 2013 (4 años 2 meses)

Buenos Aires

Consultoría e Instrucción en fundamentos, herramientas y buenas prácticas de ETL.

Herramientas Involucradas:

1) Oracle Data Integrator

2) Kettle

3) SQL Server DTS

TECSO

1 año 8 meses

Desarrollador SSr-D

enero de 2012 - julio de 2012 (7 meses)

Rosario

Bajo la modalidad de consultor técnico, desarrollo actualmente actividades de:

1. Instalación, migración, preparación de entornos y tuning para:

IBM Maximo 4.2, 5.2, 6.x, 7.1, 7.5.0.x

2. Desarrollo e Implementación On-Site de ETL de carga y migración de datos.

3. Automatización de los procesos de carga y exportación de datos para IBM Maximo.

Principales tecnologías: SQL Server 2008, IBM WebSphere Application Server 7.0.0.x, IBM HTTP Server 7.0.0.x

Desarrollador Jr-C

octubre de 2011 - diciembre de 2011 (3 meses)

Rosario

1. Construcción y mantenimiento evolutivo en proyectos municipales: SIAT-MR y SAT-MSF.

Construcción de procesos automáticos masivos en motor backend.

2. Instalación, Preparación de entornos para Customización Máximo IBM.

Desarrollador PI

diciembre de 2010 - septiembre de 2011 (10 meses)

Rosario

1. Construcción y mantenimiento evolutivo en proyectos municipales: SIAT-MR y SAT-MSF.

Participación en el desarrollo de generador de reportes en JS, personalizable, usando Rhino & Flying Saucer.

Principales tecnologías: Struts 1, Hibernate, FOP.

2. Desarrollo y mantenimiento evolutivo en eDelivery Latam y Global.

Desarrollo de generador de .tiff a partir de .pdf bajo demanda de motor backend.

Principales tecnologías: Spring, Struts 2, ActiveMQ, Axis2

3. Soporte a producción.

BPD Solutions

1 año 9 meses

BI Analyst

septiembre de 2010 - noviembre de 2010 (3 meses)

Coordinación de tareas con equipos de trabajos y clientes.

Encargado desarrollo de DLM: Data Load Methodology para unificación de Data Marts en ambientes dispares. Inclusión de ETL interno para desarrollo global de equivalencias.

Desarrollo de un punto de entrada en JSP para automatización remota con Java, comunicandose mediante Web Services implementados sobre Axis2 y ODI.

Certificaciones:

Certificación Quiterian Dynamic Data Web: Administracion, Analisis y ETL
(Herramienta de analisis dinámico)

BI Consultant

marzo de 2009 - agosto de 2010 (1 año 6 meses)

Tareas:

Implementación de sistemas financieros y de planificación presupuestaria.

- Soporte y desarrollo On-Site.
- Desarrollo de metodologia de carga de datos para unificación de datos.
- Encargado de automatizacion en Data Warehouse y Cubos.
- Trabajo en paralelo con clientes y equipos de desarrollo.
- Desarrollo en Oracle Hyperion 9.3.1 , 9.3.5
- Desarrollo en Oracle Hyperion 11.1.3

Herramientas:

Oracle Data Integrator
Essbase 7.x, 9.3.x, 11.x
Hyperion Planning
Hyperion Financial Reporting
Hyperion Web Analysis
SQL Server 2005

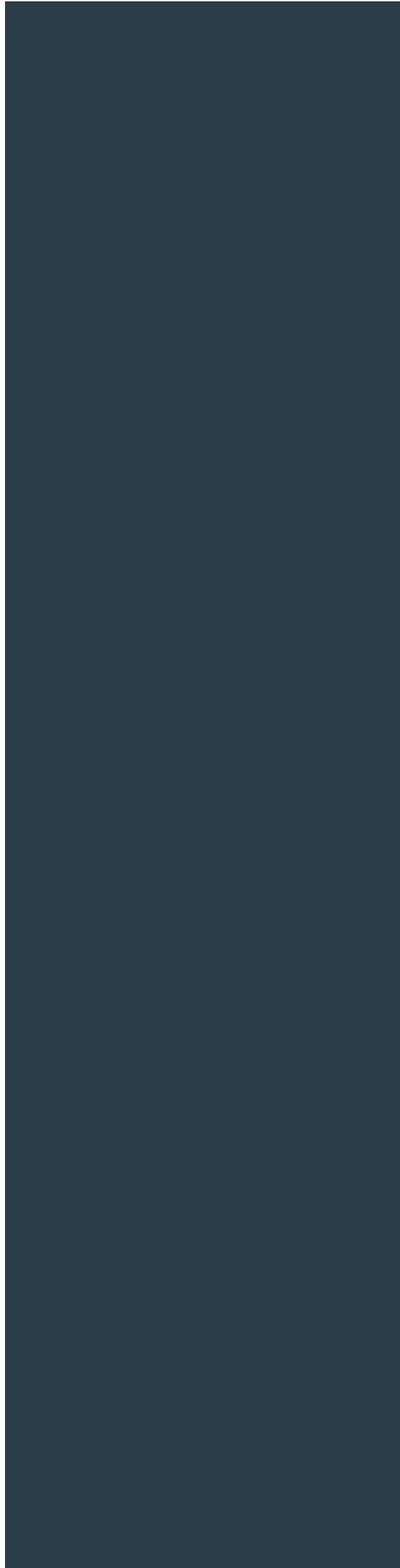
Certificaciones:

Hyperion Essbase 7.1.2 Certified Consultant
Hyperion Planning 9.3 Administrator I & II Certified Expert

Educación

Massachusetts Institute of Technology
Machine Learning for Big Data and Text Processing, Ciencias de la
computación · (2016 - 2016)

Universidad de Buenos Aires

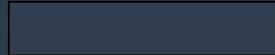


Magister en Explotación de Datos y Descubrimiento del Conocimiento, Data Mining, Posgrado, Statistics & Computer Science, Statistics & Computer Science · (2013 - 2014)

Ingeniero, Sistemas de información, Sistemas de información · (2004 - 2009)

ANEXO J

C.V. Sebastián Lopreto

Contactar

www.linkedin.com/in/sebastian-lopreto (LinkedIn)

Aptitudes principales

Java

JavaScript

Metodologías ágiles

Languages

Inglés (Full Professional)

Español (Native or Bilingual)

Certifications

Professional Scrum Master I (PSM I)

Certified Scrum Product Owner (CSPO)

Sebastián Lopreto

Ingeniero Electrónico, Especialista en Sistemas

Extracto

Soy un Ingeniero Electrónico con más de 10 años de experiencia en Sistemas. He trabajado tanto en desarrollos de software como de hardware, siempre con la misma pasión. Amante de la ciencia y la tecnología, busco constantemente desafíos y fomentar un buen ambiente de trabajo.

Mis compañeros me definen como una persona que sabe transmitir ideas y que sabe escuchar.

Dedico mucho tiempo a entender las necesidades de cada negocio. Puedo trabajar bien solo, pero me siento mejor si lo hago en colaboración con los demás.

Siempre estoy interesado en escuchar a compañeros, colegas, o simplemente gente interesante y creativa, así que no dude en ponerse en contacto conmigo si desea conectar.

Experiencia

Accusys Technology

Product Owner

agosto de 2018 - Present

Argentina

redbee Studios

Senior Software Engineer Team Lead

junio de 2017 - agosto de 2018 (1 año 3 meses)

Argentina

GlobalLogic Latinoamerica

Senior Software Engineer Team Lead

septiembre de 2015 - junio de 2017 (1 año 10 meses)

Argentina

Inmental Interactive

Full-stack Developer

mayo de 2013 - septiembre de 2015 (2 años 5 meses)

Page 1 of 2

QB9 Entertainment
Full-stack Developer
enero de 2010 - mayo de 2013 (3 años 5 meses)
Argentina

IBM
Full-stack Developer
enero de 2009 - diciembre de 2009 (1 año)
Argentina

Capgemini
Full-stack Developer
enero de 2008 - diciembre de 2008 (1 año)
Argentina

Educación

Universidad Tecnológica Nacional
Posgrado en Gestión de Proyectos, Gestión de proyectos · (2018 - 2018)

Universidad Tecnológica Nacional
Especialista en Ingeniería de Sistemas de la
Información, Ingeniería · (2015 - 2017)

Universidad Tecnológica Nacional
Ingeniero Electrónico, Ingeniería · (2004 - 2011)

Colegio PIO IX
Técnico Electrónico · (1998 - 2003)

ANEXO K

C.V. Ramiro Garbarini

CURRICULUM VITAE

Ramiro Walter Garbarini Ingeniero en Sistemas de Información

TITULACIONES

- TÍTULOS UNIVERSITARIOS:

1. Ingeniero en Sistemas de Información, Universidad Tecnológica Nacional, Facultad Regional Buenos Aires. Diciembre 2002.
2. Analista Universitario de Sistemas, Universidad Tecnológica Nacional, Facultad Regional Buenos Aires. Diciembre 2000.

- ESTUDIOS UNIVERSITARIOS EN CURSO

1. Magister en Tecnología Informática aplicada en la Educación - Universidad Nacional de la Plata, Facultad de Informática.
Tema de Tesis: "Diseño Conceptual de Espacios Virtuales para el desarrollo de proyectos en materias de carrera de grado".

CALIFICACIONES PROFESIONALES Y OTROS ESTUDIOS

- CURSOS DE POSGRADO

1. Universidad Nacional de La Plata, Facultad de Ciencias Informáticas, Secretaría de PostGrado – "Taller de Elaboración de Tesis" - 50 horas, Segundo Semestre 2011.
2. Universidad Nacional de La Plata, Facultad de Ciencias Informáticas, Secretaría de PostGrado – "Entornos de Aprendizaje en Hipermedia. Desarrollo de Material Educativo" - Programa Magister en Tecnología Informática aplicada en la Educación – 140 horas, Segundo Semestre 2010.
3. Universidad Nacional de La Plata, Facultad de Ciencias Informáticas, Secretaría de PostGrado – "Representación, Almacenamiento y Recuperación de Conocimiento" - Programa Magister en Tecnología Informática aplicada en la Educación – 170 horas, 2010.

4. Universidad Nacional de La Plata, Facultad de Ciencias Informáticas, Secretaría de PostGrado – “Psicología Cognitiva Aplicada a Informática en Educación” - Programa Magister en Tecnología Informática aplicada en la Educación – 120 horas, Segundo Semestre 2009.
 5. Universidad Nacional de La Plata, Facultad de Ciencias Informáticas, Secretaría de PostGrado – “Tecnología Informática, Evolución y Aplicaciones” - Programa Magister en Tecnología Informática aplicada en la Educación – 150 horas, Segundo Semestre 2009.
 6. Universidad Nacional de La Plata, Facultad de Ciencias Informáticas, Secretaría de PostGrado – “Educación a Distancia” - Programa Magister en Tecnología Informática aplicada en la Educación – 150 horas, Primer Semestre 2009.
 7. Universidad de Belgrano, Departamento de Posgrado y Capacitación Continua – Curso “Project Management” – Duración: 96hs - Finalizado Marzo 2004.
- OTRAS CAPACITACIONES
 1. Scaled Agile, SAFe DevOps Validation (Trainer Enablement 4.6) – Diciembre 2018.
 2. Scaled Agile, SAFe DevOps Validation (Trainer Enablement 4.5) – Diciembre 2018.
 3. Scaled Agile, SAFe 4 Scrum Master (Trainer Enablement 4.6) – Diciembre 2018.
 4. Scaled Agile, SAFe for Teams Validation (Trainer Enablement 4.6) – Noviembre 2018.
 5. Scaled Agile, SAFe 4 Scrum Master (Trainer Enablement 4.5) – Abril 2018.
 6. Scaled Agile, SAFe 4 Advanced Scrum Master (Trainer Enablement 4.5) – Enero 2018.
 7. Scaled Agile, SAFe 4 Product Owner/Product Manager (Trainer Enablement 4.5) – Octubre 2017.
 8. Scaled Agile, SAFe for Teams (Trainer Enablement 4.5) – Septiembre 2017.
 9. Scaled Agile, Leading SAFe (Trainer Enablement 4.5) – Agosto 2017.
 10. Scaled Agile, SAFe 4.0 Scrum Master (Instructor - Training Validation) – Agosto 2017.
 11. Scaled Agile, Implementing Scaled Agile Framework (SAFe) Course (4.5), Certificate Id: 32794733-3150 – 32 horas – Junio 2017. Virginia, EEUU.
 12. Kleer S.R.L., Scrum Alliance Registered Education Provider – “Agile Software Development” - 24 horas – Diciembre 2011.
 13. Kleer S.R.L., Scrum Alliance Registered Education Provider – “Scrum Developer Engineering Practices Assessment” - 24 horas – Diciembre 2011.
 14. Kleer S.R.L., Scrum Alliance Registered Education Provider – “Scrum Estimating & Planning” - 8 horas – Noviembre 2011.
 15. Kleer S.R.L., Scrum Alliance Registered Education Provider – “Introduction to Scrum” - 8 horas – Noviembre 2011.
 16. Aeroterra S.A. – “ArcGIS Server: Web Administration Using Microsoft .Net” - 24 horas – Abril 2010.
 17. Aeroterra S.A. – “ArcGIS Server Configuración para Oracle” - 16 horas - Abril 2010.

18. Aeroterra S.A. – “GeoDataBase Construcción” – 24 horas - Abril 2010.
 19. Aeroterra S.A. – “ArcGis III” – 16 horas – Marzo 2010.
 20. Aeroterra S.A. – “ArcGis II” – 24 horas - Marzo 2010.
 21. Universidad Tecnológica Nacional – Facultad Regional Buenos Aires – Secretaría de Cultura y Extensión Universitaria – “Auditor Interno de Sistemas de Gestión de Calidad bajo normas ISO” – 16 horas – Julio 2009.
 22. Universidad Tecnológica Nacional – Facultad Regional Buenos Aires – Secretaría de Cultura y Extensión Universitaria – “Academia SAP MM: Material Management – Módulos: TSCM50 - TSCM52” – 140 horas, Noviembre 2008.
 23. IBM Argentina – “Seminario: Testeo efectivo de Software – Herramientas y estrategias para proyectos exitosos” – 8 horas. Agosto 2007.
 24. IBM Argentina – “Modelado de Aplicaciones con Rational XDE” - 16 horas - Febrero 2005.
 25. SADIO – “Tutorial: Modelización de Arquitecturas de Software con UML” - 8 horas - Septiembre 2002.
 26. SADIO – “Introducción a la Auditoría de Sistemas” – 20 horas - Junio 2000.
 27. Universidad de Buenos Aires – Facultad de Ciencias Económicas – Secretaría de Asistencia Técnica y Pasantías - “Conducción de Equipos de Trabajo” – 8 horas – Marzo 2000.
 28. Instituto Tecnológico de Buenos Aires - Escuela de Posgrado - Centro de Actualización Permanente en Ingeniería del Software – “Taller de Ingeniería en Software: Seguimiento de Proyectos Asistida por PC” - 24 horas – Noviembre 1999.
 29. Instituto Tecnológico de Buenos Aires - Escuela de Posgrado - Centro de Actualización Permanente en Ingeniería del Software – “Taller de Ingeniería en Software: Estimación de Proyectos Asistida por PC” - 24 horas – Noviembre 1999.
 30. Instituto Tecnológico de Buenos Aires - Escuela de Posgrado - Centro de Actualización Permanente en Ingeniería del Software – “Curso: Diseño de Bases de Datos Relacionales y Conceptos Clave de la Orientación a Objetos” - Octubre 1999.
 31. SADIO – “Como Planificar y Controlar Proyectos” – 16 horas - Junio 1999.
 32. ORACLE Education – “Curso: Designer/2000: Diseño de Aplicaciones” – Marzo 1998.
 33. ORACLE Education – “Curso: Designer/2000: Diseño de Datos” – Marzo 1998.
 34. ORACLE Education – “Curso: Designer/2000: Técnicas de Modelización de Sistemas” – Marzo 1998.
 35. ORACLE Education – “Curso: Designer/2000: Técnicas de Análisis” – Marzo 1998.
- CERTIFICACIONES INTERNACIONALES
 1. Scaled Agile – “SAFe 4 Certified Program Consultant” ” – Certificate ID: 98337333-1552 – Fecha de certificado: 04 de Agosto 2017.
 2. SAP – “Certificate Solution Consultant SCM – Procurement with mySAP ERP 2005” – Certificate ID: 0005954980 – Fecha de certificado: 18 de Noviembre 2008.

IDIOMAS

- Inglés: Lee, Escribe y Habla.

ANTECEDENTES ACADÉMICOS:

- DOCENCIA

- JEFE DE LABORATORIO

1. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires – Departamento: Ingeniería en Sistemas de Información – Jefe de Laboratorios de Sistemas de Información – Grado Académico: Jefe de Laboratorio Adjunto Interino con dedicación Semi-Exclusiva - Desde: 2010 y continúa.

Actividades destacadas:

- Dirección del proyecto de implementación de un Sistema de Gestión de Calidad en Servicios. Proyecto innovador basado en la integración de los siguientes marcos de referencia de servicios: ISO 9001, ITIL, ISO 20000.
 - i. Investigación
 - ii. Análisis
 - iii. Diseño
 - iv. Implementación
 - v. Certificación:
 1. ISO 9001 – Certificado en 2011.
 2. ISO 20000 – Certificado en 2013.
- Diseño e implementación de 2 (dos) aulas-laboratorios orientadas al trabajo en equipo.
- Diseño e implementación de 1 (una) aula interactiva.
- Cambio de paradigma de uso de laboratorios, virtualización laboratorios para que el servicio se independice del espacio físico donde se utiliza.

- POSICIONES COMO PROFESOR

1. Universidad Nacional del Noroeste de la Provincia de Buenos Aires, Escuela de Tecnología, Departamento de Informática y Tecnología, Área: Teoría de Sistemas – Grado Académico: Profesor Adjunto Ordinario – Expte. 587/2014 Resolución (CS) Nro. 810/2014 - Desde: 01/02/15 y continúa.
2. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires – Departamento: Ingeniería en Sistemas de Información - Asignatura: Análisis de Sistemas – Grado Académico: Profesor Adjunto Interino - Desde: 14/11/2007 y continúa.
3. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires – Departamento: Ingeniería en Sistemas de Información - Asignatura: Sistemas de Información Geográfica – Grado Académico: Profesor Adjunto Interino – coordinador de cátedra - Desde: 01/08/2009 y continúa.

- POSICIONES COMO AUXILIAR DOCENTE

1. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires – Departamento: Ingeniería en Sistemas de Información - Asignatura: Proyecto – Grado Académico: Ayudante de Trabajos Prácticos de Primera Interino Ad-Honorem - Desde: 01/04/2012 hasta 31/03/2014.
 2. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires – Departamento: Ingeniería en Sistemas de Información - Asignatura: Análisis de Sistemas – Grado Académico: Ayudante de Trabajos Prácticos de Primera Interino - 01/04/2003 hasta 14/11/2007.
 3. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires – Departamento: Ingeniería en Sistemas de Información - Asignatura: Análisis de Sistemas – Grado Académico: Ayudante de Trabajos Prácticos de Segunda Interino - Desde: 01/04/1997 hasta 30/03/2002.
 4. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires – Departamento: Ingeniería en Sistemas de Información - Asignatura: Sistemas y Organizaciones – Grado Académico: Ayudante de Trabajos Prácticos de Segunda Interino – Desde: 01/04/1996 hasta 30/03/1997.
- POSICIONES DE DOCENTE DE EXTENSIÓN
 1. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires – Secretaría de Cultura y de Extensión Universitaria – Academia SAP MM (TSCM50 / TSCM52) – Cargo: Instructor - 2009.
- INVESTIGACIÓN
 1. Tesista investigador en el Proyecto “Metodología para la Especificación de Requisitos en Proyectos de Explotación de Información”. Área: Investigación Aplicada - Universidad Tecnológica Nacional, Facultad Regional Buenos Aires - Departamento de Ingeniería en Sistemas de Información.

FORMACIÓN DE RECURSOS HUMANOS:

- FORMACIÓN DE AUXILIARES DOCENTES
 - Universidad Tecnológica Nacional, Facultad Regional Buenos Aires.
 1. Auxiliar: Laura Recchini. Área Disciplinar: “Metodología orientada a objetos”. Desde 01/04/2007 hasta 30/11/2007. Ayudante de Trabajos Prácticos de Primera de la asignatura Análisis de Sistemas desde 01/04/2004.
 2. Auxiliar: Fabio Friedlander. Área Disciplinar: “Sistemas de Posicionamiento Globales”. Desde 01/04/2010 hasta 30/11/2010. Ayudante de Trabajos Prácticos de Primera de la asignatura Sistemas de Información Geográfica desde 01/04/2010.
 3. Auxiliar: Cecilia Sánchez. Área Disciplinar: “Explotación de Información Geográfica”. Desde 01/04/2010 hasta 30/11/2010. Ayudante de Trabajos Prácticos de Primera de la asignatura Sistemas de Información Geográfica desde 01/04/2010.

- FORMACIÓN DE AUXILIARES DE LABORATORIO

- Universidad Tecnológica Nacional, Facultad Regional Buenos Aires.
- 1. Auxiliar: Gabriela Liste. Área de Conocimiento: “Sistemas de Gestión de Calidad y de Infraestructura”. Desde 01/06/2010 hasta 30/06/2011. Auxiliar del Laboratorio de Sistemas de Información desde 01/04/2009.
- 2. Auxiliar: Pablo Lorenzatto. Área de Conocimiento: “Sistemas de Gestión de Infraestructura”. Desde 01/08/2010 hasta 30/11/2010. Auxiliar del Laboratorio de Sistemas de Información durante 01/04/2008 hasta 30/11/2010.

PUBLICACIONES

CAPÍTULOS DE LIBROS

1. Garbarini, R.; Saavedra Martínez, P. (2016). Metodología de desarrollo de software. En Straccia, L.; Zanetti, P. (eds.) (2016). Tecnología Informática. Alcances y usos en las organizaciones. 1era Edición. Ciudad Autónoma de Buenos Aires: Editorial CEIT. ISBN 978-987-1978-30-4. 176 pp.

Artículo en Revista

1. Garbarini, R., Cigliuti, P., Saavedra-Martínez, P., Pollo-Cattaneo, M. F., Recchini L. & Rivero-Peña Y. Proyecto plataforma e-learning para el aprendizaje colaborativo de la inteligencia de negocios. Revista Tecnología Educativa. Editor Responsable: Universidad de Holguín. 2018. Edición en línea, Vol. 3, No. 1. ISSN 2519-9463.
2. Garbarini, R., Gaspes, E., Osuna, G., Cigliuti, P., Saavedra-Martínez, P., Pollo-Cattaneo, M. F., Recchini L. & Rivero-Peña Y. Construcción y validación de un sistema web/móvil como herramienta de intervención tecnológica para la evaluación dinámica en asignatura de grado en ingeniería en sistemas de información. Revista Tecnología y Ciencia. Editor Responsable: Secretaria de Ciencia, Tecnología y Posgrado Universidad Tecnológica Nacional. 2018. Edición en línea, Año 16 N° 33 - ISSN 1666-6933. 059 pp.

COMUNICACIONES A CONGRESOS

1. Garbarini, R., Cigliuti, P., Saavedra-Martínez, P., Pollo-Cattaneo, M. F., Rivero-Peña Y. 2018. “Proyecto Plataforma E-Learning para el aprendizaje colaborativo de la Inteligencia de Negocios”. XVII Congreso Internacional de Informática en la Educación “INFOREDU” 2018. Habana, Cuba. Memorias de la XVII Convención y Feria Internacional Informática 2018, Editorial Joven Club, ISBN: 978-959-7255-00-0.

2. Garbarini, R., Cigliuti, P., Saavedra-Martínez, P. 2017. Propuesta de estrategia de gestión de datos maestros en el contexto de la migración a un sistema ERP. Simposio “Informática, Matemática y Ciencias de la Información”, Conferencia Científica Internacional, 8va Edición – Universidad de Holguín, Cuba. Memorias del evento, ISBN 978-959-16-3272-2. <http://eventos.uho.edu.cu/index.php/ccm/cci2017/paper/view/564>
3. Garbarini, R., Cigliuti, P., Osuna, G. 2017. Desarrollo de un sistema de puntos verdes inteligentes como instrumentos de educación ambiental. Simposio “Informática, Matemática y Ciencias de la Información”, Conferencia Científica Internacional, 8va Edición – Universidad de Holguín, Cuba. Memorias del evento, ISBN 978-959-16-3272-2. <http://eventos.uho.edu.cu/index.php/ccm/cci2017/paper/view/586>
4. Garbarini, R., Gaspes, E., Osuna, G., Cigliuti, P., Saavedra-Martínez, P., Pollo-Cattaneo, M. F., Recchini L. & Rivero-Peña Y. 2016. Construcción y validación de un sistema web/móvil como herramienta de intervención tecnológica para la evaluación dinámica en asignatura de grado en ingeniería en sistemas de información. 4to. Congreso Nacional de Ingeniería Informática / Sistemas de Información. Área Temática: Educación en Ingeniería (EI). ISSN 2347-0372.
5. Garbarini, R., Gaspes, E., Osuna, G. 2016. Sistema de atención temprana de contingencias ambientales para áreas urbanas. IX CONGRESO INTERNACIONAL GEOMÁTICA 2016 – La Habana, Cuba. Memorias de la XVI Convención y Feria Internacional Informática 2016, Editorial Joven Club (959-289), ISBN: 978-959-289-122-7.
6. Garbarini, R., Gaspes, E., Osuna, G., Cigliuti, P., Saavedra-Martínez, P. & Pollo-Cattaneo, M. F. 2015. Construcción de un Sistema de Composteras Inteligentes como instrumentos de educación ambiental. 3er. Congreso Nacional de Ingeniería Informática / Sistemas de Información. Área Temática: Aplicaciones Informáticas y de Sistemas de Información. ISBN 978-987-1896-47-9.
7. Garbarini, R., Gaspes, E., Macri, M., Osuna, G., Cigliuti, P., Saavedra-Martínez, P. & Pollo-Cattaneo, M. F. 2015. Propuesta de Sistema de Composteras Inteligentes como instrumentos de educación ambiental. Proceedings XVII Workshop de Investigadores en Ciencias de la Computación (Tecnología Informática Aplicada en Educación). Artículo 6934. ISBN 978-987-633-134-0.
8. Garbarini, R; Gaspes E.; Macri M.; Osuna G.; Larran G. 2014. Desarrollo de una metodología de apoyo para el ordenamiento territorial de los bosques nativos de la provincia de Buenos Aires mediante la utilización de sistemas de información geográfica. 2do Congreso Nacional de Ingeniería Informática / Sistemas de Información. Área Temática: Aplicaciones Informáticas y de Sistemas de Información. Pág. 98-105. ISSN 2346-9927.
9. Garbarini, R., Cigliuti, P., Saavedra-Martínez, P., Pollo-Cattaneo, M. F. & Burstyn, A. 2014. Elementos para una Estrategia de Laboratorio Virtual orientado a un Laboratorio Universitario de Ingeniería en Sistemas de Información. Proceedings XVI Workshop de Investigadores en Ciencias de la Computación. Pág. 994-998. ISBN 978-950-34-1084-4.
10. Garbarini, R; Haim, A; Cigliuti, P; Sánchez C; Saavedra Martínez, P. 2013. Evaluación de áreas geográficas adecuadas para la implementación de plantas solares termoeléctricas de torre mediante la utilización de sistemas de información geográfica. 1er Congreso Nacional de Ingeniería Informática / Sistemas de Información. Área Temática: Aplicaciones Informáticas y de Sistemas de Información.

11. Garbarini, R., Cigliuti, P., Pollo Cattaneo, Ma.F, Bursztyn, A. 2013. Implementación de un Sistema de Gestión de Calidad y Servicios en Laboratorio Universitario de Ingeniería en Sistemas de Información. VIII Congreso de Tecnología en Educación y Educación en Tecnología. ISBN 978-987-1676-04-0 Artículo: #5490.
12. Pollo-Cattaneo. M., Pytel, P., García-Martínez, R., Vegega, C., Amatriain, H., Ramón, H., Mansilla, D., Deroche, A., Cigliuti, P., Saavedra-Martínez, P., Garbarini, R., Rodríguez, D., Britos, P., Tomasello, M. (2013). Prácticas y Aplicaciones de Ingeniería de Requisitos en Proyectos de Explotación de Información. Proceedings del XV Workshop de Investigadores en Ciencias de la Computación, Pág. 171-175. ISBN 978-9-872-81796-1
13. Rodríguez, D., Charczuk, N., Garbarini, R., García-Martínez, R. 2012. Trabajo Colaborativo basado en Espacios Virtuales. Proceedings II Jornadas de Enseñanza de la Ingeniería (JEIN 2012). Pág. 192-199. ISSN 2313-9056. Programa de Tecnología Educativa y Enseñanza de la Ingeniería (TEyEI). Universidad Tecnológica Nacional.
14. Garbarini, R., Rodríguez, D., Pollo-Cattaneo, F., García-Martínez, R. 2012. Elementos para una Arquitectura de Espacio Virtual para Trabajo Colaborativo Orientado a Trabajos Finales Grupales en Carreras de Grado. Proceedings del XVIII Congreso Argentino de Ciencias de la Computación. Pág. 619-626. ISBN 978-987-1648-34-4.

CONGRESOS:

- PARTICIPACIÓN COMO EXPOSITOR

1. Conferencia Científica Internacional, 8va Edición, Simposio “Informática, Matemática y Ciencias de la Información” – Holguín, Cuba. Conferencia Pre-evento: Implementación de un Sistema de Gestión de Calidad y Servicios en Laboratorio Universitario de Ingeniería en Sistemas de Información.
2. Conferencia Científica Internacional, 8va Edición, Simposio “Informática, Matemática y Ciencias de la Información” – Holguín, Cuba. Conferencia Pre-evento: La importancia de la Gestión de Datos Maestros (MDM) en la era del Big Data.
3. Conferencia Científica Internacional, 8va Edición, Simposio “Informática, Matemática y Ciencias de la Información” – Holguín, Cuba. Conferencia: Desarrollo de un sistema de puntos verdes inteligentes como instrumentos de educación ambiental.
4. Conferencia Científica Internacional, 8va Edición, Simposio “Informática, Matemática y Ciencias de la Información” – Holguín, Cuba. Conferencia: Propuesta de estrategia de gestión de datos maestros en el contexto de la migración a un sistema ERP.
5. IX Congreso Internacional Geomática. La Habana, Cuba. Marzo 2016. Exposición: Sistema de atención temprana de contingencias ambientales para áreas urbanas.
6. 3er Congreso Nacional de Ingeniería Informática / Sistemas de Información. Área Temática: Aplicaciones Informáticas y de Sistemas de Información. Noviembre 2015. Exposición: Construcción de un Sistema de Composteras Inteligentes como instrumentos de educación ambiental.

7. 2do Congreso Nacional de Ingeniería Informática / Sistemas de Información. Área Temática: Aplicaciones Informáticas y de Sistemas de Información. Noviembre 2014.
Exposición: Desarrollo de una metodología de apoyo para el ordenamiento territorial de los bosques nativos de la provincia de Buenos Aires mediante la utilización de sistemas de información geográfica.
 8. XVI Workshop de Investigadores en Ciencias de la Computación (WICC 2014). Área Temática: Tecnología Informática Aplicada en Educación. Artículo 6193.
Poster: Elementos para una Estrategia de Laboratorio Virtual orientado a un Laboratorio Universitario de Ingeniería en Sistemas de Información.
 9. 1er Congreso Nacional de Ingeniería Informática / Sistemas de Información. Área Temática: Aplicaciones Informáticas y de Sistemas de Información. Universidad Tecnológica Nacional, Facultad Regional Córdoba - Noviembre 2013.
Exposición: Evaluación de áreas geográficas adecuadas para la implementación de plantas solares termoeléctricas de torre mediante la utilización de sistemas de información geográfica.
 10. IV Congreso Nacional de Sistemas de Gestión y Mejora Continua. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Secretaría de Cultura y Extensión Universitaria - Septiembre 2013.
Exposición: Medición de la calidad en organizaciones prestadoras de servicios.
 11. VIII Congreso de Tecnología en Educación y Educación en Tecnología. UNSE – Junio 2013.
Exposición: Implementación de un Sistema de Gestión de Calidad y Servicios en Laboratorio Universitario de Ingeniería en Sistemas de Información.
 12. XVIII Congreso Argentino de Ciencias de la Computación – Workshop Tecnología Informática Aplicada a la Educación (WTIAE). Octubre 2012.
Exposición: Elementos para una Arquitectura de Espacio Virtual para Trabajo Colaborativo Orientado a Trabajos Finales Grupales en Carreras de Grado.
 13. 3er Congreso Nacional de Sistemas de Gestión y Mejora Continua. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Secretaría de Cultura y Extensión Universitaria - Septiembre 2011.
Exposición: Proyecto Calidad FRBA – DISILAB ISO 9001:2008.
 14. 2do Congreso Nacional de Sistemas de Gestión y Mejora Continua. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Secretaría de Cultura y Extensión Universitaria – Agosto 2010.
Exposición: Calidad en Servicios de IT: Relación entre ISO 9001, ITIL, ISO 20000.
 15. 1er Congreso Nacional de Sistemas de Gestión y Mejora Continua. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Secretaría de Cultura y Extensión Universitaria – Agosto 2009.
- EVALUADOR DE TRABAJOS
 1. 4to. Congreso Nacional de Ingeniería Informática / Sistemas de Información, CONAISI 2016 - Argentina.
 2. 3er. Congreso Nacional de Ingeniería Informática / Sistemas de Información, CONAISI 2015 - Argentina.
 - PARTICIPACIÓN COMO PERSONAL TÉCNICO
 1. INFOCOM '96 – La Tecnología al Servicio del Hombre – del 10 al 16 de Junio 1996.

- ASISTENCIA A:

1. Congreso de Producción del Bicentenario - Instituto Argentino del Petróleo y Gas (IAPG) – 32 horas - Mayo 2010.
2. 1ra Jornadas gvSIG de Latinoamérica y el Caribe. Cooperar es Crecer - Instituto Geográfico Nacional (IGN) – 24 horas - Septiembre 2009.

TRANSFERENCIA

UTN FRBA - Programa de las Naciones Unidas para el Desarrollo, Proyecto PNUD ARG09/G46 FREPLATA II – 2013 - Diseño conceptual de un Sistema de Gestión de Información Ambiental para la Cuenca del Arroyo del Gato.

UTN FRBA – Dirección Nacional SINTyS (2013). Desarrollo e Implementación PKI Licenciada de Sistema Nacional Tributario y Social. Relevamiento, diseño y elaboración de la metodología de construcción de software. Resolución (D) N° 2083/13 – Proyecto 02-00000294-13-1.

ANTECEDENTES LABORALES:

ECOLOGIX S.R.L., Socio Fundador (abr '12 – actualidad)

Nuevo emprendimiento que tiene como fin desarrollar una empresa que brinde soluciones tecnológicas al mercado de bienes y servicios ambientales y de sustentabilidad.

Proyectos:

Desarrollo de un sistema para la gestión y protección de los bosques nativos de Argentina (MinCyT - FONTAR ANR BIO NANO TICS NA 034/12) – Director del Proyecto. (2013-2015)

Sistema de Atención Temprana de Contingencias Ambientales en Ámbitos Urbanos, sistema de información geográfico que brinda información clave para la toma de decisión de los responsables de dar respuesta a una contingencia al momento de su detección. El sistema permite visualizar el impacto de la contingencia geográficamente, pudiendo estimar la afectación de población, escuelas, hospitales, industrias, otros. Puede identificar los puntos más cercanos a los cuales evacuar la población afectada. (2014).

Sistema de Autoevaluación Ambiental, sistema que permita a las empresas integrar y evaluar los diferentes aspectos ambientales involucrados en sus negocios, tanto en lo que refiere a cumplimiento de normativa como a la gestión ambiental de aspectos críticos de su actividad. Con el desarrollo de este sistema se puede evaluar el desempeño ambiental de la empresa. (2014-2015)

Compostera 2.0, sistema de composteras con dispositivos electrónicos para medir indicadores y transmitirlos a un servidor para consolidar información y presentarlo en un tablero de control. Diseño y construcción del prototipo de la compostera y construcción del software necesario. – Director del Proyecto – Proyecto en curso.

DAEVA S.A. - IT Advisor (ene '14 – actualidad)

Asesoría en formulación y gestión proyectos de I+D.

Consultor independiente en Sistemas de Información (ago '11 – actualidad)

Consultoría especializada en implementación de Metodología en Sistemas, Gestión de Proyectos, Sistemas de Gestión de Calidad. Herramientas de competitividad y regímenes de promoción industrial.

- Clientes:

ABAKOS S.A., Diseño e implementación y certificación de su Sistema de Gestión de Calidad basado en ISO 9001:2008.

DiChena y Asociados SRL (MasterSoft), Consultoría especializada respecto al Régimen de Promoción de la Industria del Software.

Digital Express SRL, formulación de proyecto de innovación para convocatoria FONTAR 2014 – MinCyT.

E-buyPlace SA, Consultoría especializada respecto al Régimen de Promoción de la Industria del Software.

FCP Consulting SRL, Consultoría especializada respecto al Régimen de Promoción de la Industria del Software.

Global Services S.A., Consultoría especializada respecto al Régimen de Promoción de la Industria del Software.

Infosis SA, Consultoría especializada respecto al Régimen de Promoción de la Industria del Software.

Plus Sistemas SRL, Consultoría especializada respecto al Régimen de Promoción de la Industria del Software.

Procoop SRI, formulación de proyecto de innovación para convocatoria FONSOFT 2014 – MinCyT.

Renxo S.A., Consultoría especializada respecto al Régimen de Promoción de la Industria del Software. Diseño e implementación y certificación de su Sistema de Gestión de Calidad basado en ISO 9001:2015.

Staffing IT, formulación de proyectos de innovación para convocatorias FONSOFT y FONTAR 2012 – MinCyT. Consultoría especializada respecto al Régimen de Promoción de la Industria del Software.

Grupo Dafing SRL, Mantenimiento Sistema de Gestión de Calidad certificado bajo normas ISO 9001:2008.

Staffing IT S.A., Consultoría especializada respecto al Régimen de Promoción de la Industria del Software.

SoftING S.R.L.,

Consejería tecnológica para la implementación de SCRUM y reingeniería del proceso de desarrollo de software (FONTAR ANR CT-I NA 003/11).

Implementación de políticas de calidad de software.

Implementación Herramientas de Calidad, REDMINE – Instalación y Adecuación.

Consultoría para la definición del Tablero de Comando de la empresa.

Evaluador de Proyectos del Concurso Apoyo Competitividad PyME 2012 – GCBA, SubSecretaría de Desarrollo Económico – UBA FCE. (2012)

Auditor Técnico para el Régimen de Promoción de la Industria del Software - (Ago '11 – Jun '12)

Secretaría de Industria, Ministerio de Industria, Presidencia de la Nación – UBA FCE.

Las tareas realizadas en las auditorías son

Verificación técnica y de actividades realizadas por los recursos humanos de la empresa.

Verificación de actividades de I+D, evaluación del grado de innovación, de aportes realizados, y recursos utilizados.

Revisión de proyectos de desarrollo. Análisis técnico, de recursos utilizados y de facturación asociada.

Verificación de actividades de exportación.

Aeroterra S.A. (Mar '10 – Jul '11)

Gerente de Proyectos y Calidad Corporativa. (Feb '11 – Jul '11)

Responsable de la administración de los recursos profesionales de sistemas y su asignación en proyectos.

Responsable de la gestión de los proyectos de la empresa.

Responsable del Sistema de Gestión de Calidad certificado bajo ISO 9001:2008.

Responsable de formulación y gestión de proyectos de competitividad.

Líder de Proyectos y Calidad Corporativa. (Mar '10 – Ene '11)

Responsable de la implementación de metodología SCRUM para el proceso de construcción de software.

Proyecto: Yacimiento Virtual

Responsable del Sistema de Gestión de Calidad certificado bajo ISO 9001:2008.

SoftING SRL (Abr '07 – Feb '10)

Dirección de Proyectos, Líder de Proyectos. // Responsable del Sistema de Gestión de Calidad.

Cliente: SoftING SRL (proyectos corporativos)

Gestión de herramientas de fomento y de promoción a la competitividad (Fonsoft / PACC).

Proyecto: COREX (MinCyT - FONSOFT ANR 2007 - Proyecto N° NA 112/07) – Dirección del proyecto. "Desarrollo de COREX set de herramientas para la administración del proceso de desarrollo de software con tecnología Genexus". (08/2007 – 01/2009).

Proyecto: Certificación ISO 9001:2008 IEC 90003 (MinCyT - FONSOFT ANR 2008 NA 167/08) – Dirección del proyecto. (08/2007 – 11/2009).

Cliente: Nación AFJP

Proyecto NSB CC Sistema de Beneficio - Gestión del proyecto (04/2008 – 12/2008)

Proyecto Migración de aplicaciones GX 8.0 a GX 9.0 - Gestión del proyecto. (12/2007 – 04/2008)

Cliente: Nación Servicios

Proyecto: HomeBanking Ticket Nación – Gestión del proyecto de desarrollo. (11/2007 – 04/2008)

Cliente: Banco Santander Río

Gestión de proyectos de desarrollo y documentación de software. (06/2007 – 10/2007)

Carriers Interconnect (Mar '06 – Abr '07)

Software Lab, Gestor de Negocio.

Cliente: ETECSA (Cuba)

Proyecto: Implantación ITXi. PSM - Implantación del sistema de Interconexión Internacional en la empresa cubana de telecomunicaciones.

Cliente: Colombia Telecomunicaciones (Colombia)

Proyecto: Colombia Telecomunicaciones (Colombia). PM - Implantación y puesta en producción del sistema de Interconexión Internacional.

Cliente: CTC - Telefónica de Chile (Chile)

Proyecto: Implementación del sistema de Interconexión Nacional.

Pragma Consultores. Consultor (Oct '04- Feb '06)

Cliente: Repsol YPF. (Sep '05 – Feb '06)

Soporte metodológico para la definición del plan de Continuidad de Servicio (ITIL). Gerencia de Procesamientos de Datos, Dirección de Sistemas de Información.

Cliente: Pioneer Natural Resource Argentina. (Sep '05 – Feb '06)

Soporte metodológico para la administración de proyectos.

Proyecto: Wells Files, Record Management.

Proyecto: Normalización de información geográfica para Secretaría de Energía de la Nación.

Cliente: Tecpetrol S.A. (Oct '04 – Ago '05)

Proyecto Integridad de ductos, Soporte de administración de proyectos (PSM), en la implementación del sistema GIS para el control de ductos.

Proyecto: Georrelevamiento, adquisición de información satelital y modelización de yacimiento. Soporte de administración de proyectos (PSM)

Fundación Proydesa – Instructor (Abr '04 – Dic '04)

- Carrera de Administrador de Base de Datos Oracle 9i – Curso: Introducción a Oracle 9i: SQL.

Gobierno de la Ciudad de Buenos Aires, Secretaría de Hacienda y Finanzas. (Nov '03 – Dic '04)

Unidad de Informática, Consultor Analista Senior de Sistemas Análisis y diseño, especificación de requerimientos, elaboración de prototipos Gráficos y casos de prueba.

Proyecto SIGAF (Sistema Integral de Gestión Administrativa y Financiera)

Qualisys S.R.L. – Informática Estratégica (Ene '02 – Oct '03)

Cliente: PanAmerican Energy, Analista Funcional de Aplicaciones Técnicas y de Geo-ciencias, Relevamiento y análisis de requerimientos de información técnica.

Sistema *AT*, sistema de gestión del archivo técnico (Record Management). Realizando el análisis, diseño, desarrollo e implementación.

Sistema *Bdiep*, Base de datos integrada de ingeniería y producción. Administrador del sistema y a cargo de relevamiento y análisis de nuevas necesidades.

Proyecto *UNP*, Única nomenclatura de Pozos, Coordinador del proyecto.

GlobalStar – TE.SA.M. Argentina (Nov '00 – Dic '01)

Gerencia de Ingeniería y Sistemas, ABC Project. Analista Senior de Facturación y Tasación. Desempeñándome como responsable del proceso de Tasación y Mediation Device (FCC). Relevamiento y análisis de requerimientos de Tasación, Roaming, Consumos.

Telefónica de Argentina S.A. – Unidad Larga Distancia (May '99 – Oct '00)

Gerencia de Sistemas Informáticos, Desarrollo y Mantenimiento. de Sistemas. Líder de Proyecto.

Proyecto SINTI, sistema de interconexión de operadores Internacionales.

Proyecto LICOR, sistema de Liquidación de Corresponsales.

Telintar S.A., Unidad Operativa Sur (Sep '97 – May '99)

Gerencia de Informática, Pasante de Asistencia a Usuario Final.

Analista Funcional Jr. Relevamiento y análisis de requerimientos.

ANEXO L

C.V. Martín Riera

Contact



www.linkedin.com/in/mart%C3%ADn-pablo-riera-b637019
(LinkedIn)

Top Skills

Software Project Management
Project Management
PMI

Languages

Inglés (Professional Working)
Portugués (Limited Working)
Español (Native or Bilingual)

Publications

Tecnología Informática

Martín Pablo Riera

Gerente Comercial Émerix

Summary

- Encargado de Planificar, Diseñar, ejecutar y controlar los planes comerciales, alineados a los objetivos de corto, mediano y largo plazo establecidos en la Organización.
- Experiencia en roles de gestión previos: Gerente de Centro de Excelencia - 4 años, Gerente de Operaciones- 2 años, Gerente del Área de Proyectos -1 año, PM/Líder de Proyectos - 3 años.
- Dirección de Equipo Gerencial a cargo del producto, metodología, consultoría, desarrollo, calidad y preventa.
- Proyectos Gerenciados/Liderados logrando implementaciones exitosas en Argentina, Brasil, México, Chile, Costa Rica, Panamá y Honduras.
- Ingeniero en Sistemas de Información – UTN Facultad Regional Buenos Aires.
- Otras Habilidades: Negociación y Administración de conflictos, Experto en soluciones para el Negocio de Cobranzas y Recupero, Soporte experto a la preventa y procesos comerciales.

Experience

Émerix

Gerente Comercial
January 2018 - Present

Ciudad autónoma de Buenos Aires, Argentina

Soft Office

13 years 3 months

Gerente del Centro de Excelencia
November 2013 - January 2018 (4 years 3 months)

Dirigir el equipo de managers, expertos y consultores que conforman el Centro de Excelencia Emerix. Esta unidad de trabajo es cross a la compañía y aborda 5 ejes claves para el desarrollo y posicionamiento estratégico de la empresa:

- Gerencia de Producto: redefinición y desarrollo de la familia de productos Emerix y soluciones de cobranzas para diversos mercados en Latinoamérica.

- Émerix Network: posicionamiento del expertise en el Mercado gestionando el conocimiento como líderes en cada temática específica, con presencia activa en distintos canales de exposición (congresos, webinars, newsletters, portal web).
- Preventa: soporte experto y consultivo al proceso de venta, integrando estrategia comercial con necesidades de Mercado.
- Metodología de Implementación: adopción de nuevas metodologías de trabajo que optimicen la calidad y esfuerzo, minimizando riesgos para el cliente, y que faciliten la tercerización de las implementaciones.
- Unidad de Formación Émerix: distribución y paquetización del conocimiento para toda la compañía, y formación de nuevos especialistas propios y de terceros.

Responsable de alinear la innovación y actividades desarrolladas sobre estos ejes claves de acción, con el plan estratégico de la compañía. Dar soporte experto de gestión, negocio y producto a las gerencias de Operaciones y Comercial. Responsable de Calidad ISO 9001/90003.

Gerente de Operaciones

April 2011 - November 2013 (2 years 8 months)

Responsable de las gerencias de que conforman la Operación de la compañía: Desarrollo e Infraestructura, Proyectos de Implementación y/o desarrollo, Servicios al cliente. Asegurar la ejecución y facturación de las implementaciones y los servicios Post producción. Integrar los procesos y las metodologías de las tres Gerencias que conforman operaciones, Liderar e integrar a los gerentes de cada una de estas áreas. Manejo de tableros gerenciales e indicadores, seguimiento y control de Proyectos, Servicios, y roadmap de producto. Establecimiento de objetivos anuales de los colaboradores, alineados a los objetivos establecidos para el área. Motivación, coaching, soporte experto en instancias de negociación y manejo de conflictos. Gestión de personal a cargo, jefe directo de 3 gerentes de quienes depende una estructura de 30 colaboradores. Soporte a los procesos comerciales de Hunting y Farming. Asegurar el cumplimiento de las metodologías establecidas y de los procesos de calidad ISO 9001/90003, concernientes a las gerencias de operaciones.

Gerente del área de Proyectos

January 2010 - March 2011 (1 year 3 months)

Responsable de todos los proyectos de implementación y/o desarrollo de la compañía. Asegurar la facturación. Manejo de tableros gerenciales, seguimiento y control de Proyectos, asignación de recursos, tablero de

Page 2 of 4

rentabilidad. Establecimiento de objetivos anuales de los colaboradores, alineados a los objetivos establecidos para el área. Motivación, coaching, soporte experto a los PMs en la negociación y manejo de conflictos. Detección de talentos y desarrollo de colaboradores. Gestión de personal a cargo, 12 colaboradores en el área. Relacionamiento con el cliente, generación de nuevos negocios sobre la base instalada. Asegurar el cumplimiento de las metodologías de implementación y políticas de calidad.

Gerente/Líder de Proyectos

June 2008 - January 2010 (1 year 8 months)

Planificación, control y seguimiento de proyectos. Evaluación y gestión de riesgos. Gestión y resolución de conflictos internos y con el cliente. Detección de desvíos y gestión de acciones correctivas. Control de cumplimiento de estándares y normas de implementación de proyectos. Gestión y negociación del alcance, tiempo y costo de los proyectos. Control y seguimiento de Rentabilidad de los proyectos.

Experiencia en Clientes: Proyectos de Implementación Sistema de Gestión de Cobranza para Mora Avanzada (Banco Primera línea Internacional), Proyectos de Administración operativo/contable de carteras en mora avanzada (Banco Público), gestión de Mora Judicial y Legales (Banco Privado local), Proyectos Mora Temprana (Financiera) , y Proyecto Regional de implementación de gestión de cobranzas (empresa multinacional, 4 Países sud y centro América)

Líder de Proyectos

October 2005 - June 2008 (2 years 9 months)

Planificación, control y seguimiento de proyectos. Evaluación de riesgos. Gestión de conflictos internos y con el cliente. Detección de desvíos y realización de acciones correctivas. Control de cumplimiento de estándares de implementación (metodología, construcción y documentación). Especificación detallada y control del alcance.

Experiencia en Clientes: 3 Bancos, Agencia Nacional de Promoción Científica y Tecnológica , Procesadora de tarjetas internacional, Caja de Previsión, Courier Internacional implementaciones exitosas en 6 países, y un retail.

Analista Funcional

November 2004 - October 2005 (1 year)

Analista funcional en proyectos de implementación Emerix. Responsable del modelado de estrategias de negocio, análisis de procesos de migración de datos, análisis de integración con aplicativos centralizados del cliente, análisis

de integración con sistemas de terceros (sistemas de correo, dispositivos discadores de potencia/predictivos)

Experiencia en clientes:

- Proyecto de implementación de sistema de Gestión de Cobranzas en Banco, Mora Temprana.
- Soporte Post-Producción Sistema de Gestión de Morosidad en Empresa de Distribución de Gas

Deloitte Argentina

Analista Profesional de ERS

March 2004 - October 2004 (8 months)

Evaluación de procesos de negocio desde su perspectiva funcional y tecnológica. Auditoría en áreas de sistemas, soporte a la auditoría financiera.

Education

Universidad Tecnológica Nacional

Master of Business Administration (MBA) (incompleto), Administración y gestión de empresas, general · (2011 - 2013)

Dale Carnegie Training

Presentaciones de Alto Impacto, Comunicación empresarial · (2011 - 2011)

Universidad de Belgrano

Management para Mandos Medios, Administración y gestión de empresas, general · (2009 - 2009)

Universidad Tecnológica Nacional

Especialista en Project Management, Gestión de proyectos · (2008 - 2008)

PMValue

Metodología de Administración de Proyectos, Gestión de proyectos · (2005 - 2005)

ANEXO M

C.V. Marcelo Masci



Curriculum Vitae

Esp. Ing. Marcelo Adrián Masci

3 de Febrero de 1979 | Argentino | Casado | DNI 27.181.201

CABA
p-3919✉ marcelomasci@gmail.com🌐 <http://ar.linkedin.com/in/marcelomasci>

Continuar mi desarrollo como manager de equipos multidisciplinarios, para llevar adelante proyectos de valor agregado para la organización, apoyados en la gestión por procesos, la gestión de calidad y la gestión del conocimiento.

EXPERIENCIA LABORAL	Sep/13 – Actual	Banco Comafi – www.comafi.com.ar Industria Bancaria - Servicios financieros
	Jul/17 - Actual	Jefe de Equipo de Gestión de la Demanda de la GERENCIA DE GESTIÓN Y SEGUIMIENTO, GERENCIA DE SERVICIOS CORPORATIVOS. Responsable de la detección y relevamiento de las necesidades de la Organización, el análisis de impacto en procesos, sistemas y operaciones, la estimación de costos, y el análisis del costo/beneficio.
	Sep/13 – Jun/17	Jefe de Equipo de Mejora Continua de la GERENCIA DE CONTROL DE CALIDAD, GERENCIA DE SISTEMAS. Responsable de los equipos de Mejora de Procesos, Desarrollo de Herramientas TICs y Control de Presupuesto. Responsable de asegurar mediante la implementación de una metodología de sistemas, el adecuado desarrollo, adquisición, implementación y mantenimiento de sistemas, considerando la estrategia del área. Responsable del control presupuestario de toda la Gerencia de Sistemas. Logros ▶ Implementación de un nuevo sistema para la gestión de la demanda de TI y mejora de todo el proceso para 1500+ usuarios sobre Microsoft XRM Dynamics utilizando metodología SCRUM. Conformación de un equipo de trabajo de 6 personas con diversos roles para lograr los objetivos del área.
	May/05 – Ago/13	Soft Office – www.softoffice.com.ar Servicios Informáticos– Banca / Financieras / Consumo / Servicios públicos / Call Centers / Manufactura y Servicios
	Jul/09 – Ago/13	Jefe de Proyectos de Mejora Continua de la SUB-GERENCIA GENERAL. Responsable el equipo de proyecto para el cumplimiento de los <i>procesos</i> y las <i>políticas</i> de acuerdo los modelos CMMi e ISO con foco en los proyectos de implementación de la compañía. Responsable equipo de proyectos de gestión del conocimiento, incluyendo la gestión de la <i>Base de Conocimientos</i> , coordinación de reuniones de pares y planes de inducción. Responsable del equipo proyectos de innovación tecnológica para obtener <i>financiamiento</i> del Estado (Subsidios, Créditos, etc.). Logros ▶ Gerenciamiento exitoso del proyecto de certificación de la norma ISO 9001/90003 con menor presupuesto económico del estimado. Colaboración en el crecimiento del área de Financiamiento, permitiendo obtener más capital para financiar el crecimiento de la empresa.
	Ago/06 – Jul/09	Líder de Proyectos de la GERENCIA DE PRODUCTOS ÉMERIX (Sistemas de Gestión de Carteras de Clientes, Otorgamiento de Créditos y CRM). <i>Responsable</i> por el Desarrollo y Mantenimiento de Productos, coordinando un equipo de analistas funcionales y testers, brindando servicios a las Gerencias de Implementaciones, Servicios al Cliente y Comercial. <i>Responsable</i> de Capacitación a Clientes y Colaboradores de Soft Office. Logros ▶ Formación del área de Testing, mejorando radicalmente la calidad de los productos de la empresa. Colaboración en el crecimiento del área de Desarrollo de Producto, permitiendo la profesionalización de los colaboradores. Desarrollo del Proceso de Inducción de nuevos empleados, permitiendo reducir el tiempo en que un nuevo colaborador pasa a ser productivo.
	May/05 – Ago/06	Líder Funcional del PROYECTO DE IMPLEMENTACIÓN DE ÉMERIX(Sistema de Gestión de Cobranzas y CRM) para el BANCO CREDICOOP . <i>Análisis y Diseño</i> de Estrategias de Negocio. <i>Configuración</i> del Producto. <i>Capacitación. Análisis y Diseño</i> de Customizaciones. Logros ▶ Construcción de un kit de herramientas para realizar el seguimiento de los requerimientos desde su captura hasta su realización.
	Dic/00 – May/05	Grupo Sidus – www.sidus.com.ar Grupo de Empresas Farmacéuticas
	Dic/00 – May/05	Analista Funcional
	May/00 – Dic/00	Teleperformance Argentina – www.teleperformance.com Administración de Contact Centers
May/00 – Dic/00	Administrador de redes	

EXPERIENCIA DOCENTE	Feb/12–Actual	Instituto de Formación Técnica Superior Nro.19 <i>Av. Rivadavia 3846 – Ciudad Autónoma de Buenos Aires</i>
	Feb/12 - Actual	Docente titular de los seminarios "Profundización en Sistemas Aplicados" y "Profundización y Actualización" de la carrera de Técnico Superior en Análisis de Sistemas.
EXPERIENCIA DOCENTE	Dic/03– Mar/11	Universidad Tecnológica Nacional – Facultad Regional Buenos Aires <i>Departamento de Especialidad de Sistemas de Información Medrano 951 – Ciudad Autónoma de Buenos Aires</i>
	Dic/03 –Mar/11	Docente auxiliar en varias asignaturas de la carrera Ingeniería en Sistemas de Información, de la Universidad Tecnológica Nacional – Facultad Regional Buenos Aires.
FORMACIÓN	Título de Posgrado Mar/16 – Actual	Magister en Gestión Estratégica de Sistemas y TI Universidad de Buenos Aires – Escuela de Posgrado de la Facultad de Ciencias Económicas <i>Cursada Finalizada. Trabajo Final de Maestría en curso.</i>
	Título de Posgrado Mar/11 – Dic/13	Especialista en Ingeniería Gerencial Universidad Tecnológica Nacional – Escuela de Posgrado. <i>Promedio final 9,13</i>
	Título de Grado Mar/98 – Dic/03	Ingeniero en Sistemas de Información Universidad Tecnológica Nacional – Facultad Regional Buenos Aires. <i>Promedio final 7,50</i>
	Título Medio Mar/92 – Dic/97	Técnico Electrónico Colegio Nuestra Señora de La Paz. <i>Promedio final 8,60.</i>
		<i>Certificaciones / Cursos / Seminarios</i>
	Sep/10 – Sep/10	Introducción a la serie de Normas IRAM-ISO 9000. Guía de Aplicación ISO 90003 para sistemas de software – IRAM
	Ago/09 – Sep/09	Curso de Posgrado en Liderazgo y Dirección de Personas – UADE Executive Education
	Ago/08 – Sep/08	Administración de Proyectos con MS Project 2007 – Grupos de Usuarios Microsoft
	Jul/07 – Nov/07	PMI Especialista en Administración de Proyectos – IT Quality
	Abr/07 – May/07	Taller de Negociación – Universidad Tecnológica Nacional
Feb/06 – Mar/06	PMI Metodología de Administración de Proyectos – Pmvalue	
May/04	I Congreso Nacional de Estudiantes de Sistemas y Tecnologías de la Información –UTN FRBA	
May/04	Seminario de Metodologías Ágiles – UTN FRBA	
Dic/03	Certificación de Calificación Profesional IBM – IBM / UTN <i>Orador invitado</i>	
Ene/15 – Dic/15	Universidad Tecnológica Nacional – Maestría en Ingeniería en Sistemas de Información Orador invitado por el Mg. Luciano Straccia - Modelos de Organizaciones y Sistemas de Información.	
Sep/08 – Sep/10	Universidad de Buenos Aires– Maestría en Administración de Negocios Orador invitado por el Mg. Daniel Piorun para el seminario de Gestión del Conocimiento.	
Jul/10– Jul/10	Universidad Nacional Experimental Simón Rodríguez (Venezuela) Orador invitado por el Profesor Luis Alvarez para el seminario de Gestión de Proyectos. <i>Investigación y Desarrollo</i>	
Ene/10 – Sep/13	Director de Proyectos de Innovación Tecnológica financiados por la Agencia Nacional de Promoción Científica y Tecnológica , bajos los fondos FONTAR y FONSOFT. <i>Idiomas</i>	
Mar/92 – Dic/97	Inglés – Estudios particulares y cursos corporativos Conocimientos avanzados en comprensión y expresión, oral y escrito. Prácticas avanzadas de conversación.	
Actual	Aseguramiento de la Calidad. Gestión del Conocimiento. Gestión de Proyectos de Software. Gestión de Procesos. Gestión de Calidad (ISO 9001, CMMi). Metodologías Ágiles. SCRUM. Auditoría de Calidad. Transferencia de Conocimiento. Mejora de Procesos. Gestión de Procesos de Negocio (BPM). Gestión de Desarrollo de Productos Software. Gestión de Financiamiento. Gestión de Servicios. Liderazgo de Equipos de Trabajo. Manejo del Relacionamiento con los Clientes. Innovación. Negociación. Organización. Optimización. Planificación. Capacitación. Coordinación.	

ANEXO N

C.V. Pablo Cigliuti

Currículum Vitae

Pablo Cigliuti, CBIP, CDMP

Resumen

Pablo Cigliuti es ingeniero en sistemas de información con experiencia laboral de más de 12 años en el área de Business Intelligence. Es socio fundador de Dthink, consultora especializada en la temática y está certificado internacionalmente como profesional de Data Management (CDMP) y de Business Intelligence (CBIP).

Además es titular de la cátedra de Business Intelligence en la Universidad Tecnológica

Nacional y coordinador de los cursos de actualización de posgrados de Business Intelligence y de Data Mining de la misma universidad. También dirige el Instituto de Análisis de Información perteneciente a la fundación FUNDESCO.

1. Estudios:

Posgrado:	<ul style="list-style-type: none"> ● Magister en Tecnología Informática aplicada a la Educación. – Facultad de Informática – U.N.L.P. – Tesis defendida satisfactoriamente en marzo 2016 ● Curso de Project Managment, metodología del P.M.I. – Realizado en la Universidad de Belgrano – año 2004 – 80 hs.
Universitarios:	<ul style="list-style-type: none"> ● Ingeniero en Sistemas de Información – Universidad Tecnológica Nacional – Facultad Regional Buenos Aires – 2003
Secundarios:	<ul style="list-style-type: none"> ● Título de Bachiller Común del Colegio E.M.E.M. N° 6 Manuel Belgrano.
Inglés:	<ul style="list-style-type: none"> ● Hasta 6° año, con exámenes rendidos en el Inst. Lenguas Vivas.

2. Experiencia docente:

Coordinador Académico / Docente	<ul style="list-style-type: none"> ● Universidad Tecnológica Nacional – Facultad Regional Buenos Aires Posgrado de actualización de Business Intelligence e Introducción al Data Mining. – 2011 a la actualidad Co coordinador del curso de Data Mining – 2015 a la actualidad
Jefe de Cátedra / Profesor Adjunto	<ul style="list-style-type: none"> ● Universidad Tecnológica Nacional – Facultad Regional Buenos Aires Materia: Inteligencia de Negocios (electiva)– Abr. 09 a la actualidad.
Profesor Adjunto	<ul style="list-style-type: none"> ● Universidad Tecnológica Nacional – Facultad Regional Buenos Aires Materia: Análisis de Sistemas – Abr. 07 a la actualidad.

Jefe de Laboratorios	<ul style="list-style-type: none">● Universidad Tecnológica Nacional – Facultad Regional Buenos Aires Coordinador de laboratorios dependiente del Departamento de Sistemas de Información – Abr. 10 a la actualidad.
	<ul style="list-style-type: none">● Instituto Tecnológico de Buenos Aires Materia: Gestión de Datos - Jul. 09 a Mar 10.
A.T.P.	<ul style="list-style-type: none">● Universidad Tecnológica Nacional – Facultad Regional Buenos Aires Materia: Análisis de Sistemas – Abr.99 a Mar. 07

3. Antecedentes Laborales:

Dthink (Abr. 12 a la actualidad) **Cargo Actual:**
Socio-Fundador

Brindamos servicios especializados en Business Intelligence para organizaciones de diferente tipo, asesorando, liderando y/o diseñando soluciones de este tipo.

Nobleza Piccardo (Jun. 06 a Mar. 12)

Cargo Actual: Application Support Manager - Ene. 2011 a Mar 2012.

Tareas: Como gerente de soporte de aplicaciones soy responsable, junto a mi equipo, de brindar soporte a todas las aplicaciones que utiliza la empresa. Tengo a mi cargo un total de 9 personas, siendo 3 de ellas empleados de Nobleza Piccardo y el resto tercerizados. Además, me encargo de la contratación y negociación con proveedores. El área es nueva, con lo cual tengo como principal desafío establecer una metodología de trabajo aplicando las mejores prácticas propuestas por ITIL.

Cargo: Business Intelligence Manager - Jun. 06 a Dic 10

Tareas: Ingresé como BI Executive y al año fui promovido a BI Manager. Como gerente de BI tuve a cargo la definición de la estrategia de consumo de información para la compañía y la evolución de los distintos *data warehouses*. Desde Argentina también daba servicio de BI a Paraguay, Uruguay, Venezuela y Colombia. Además oficié varias veces de experto, asesorando a otros países de la región en lo correspondiente a sus soluciones de Business Intelligence.

Axxon Consulting S.A. (Feb. 2005 May. 2006)

Sector: Desarrollo. **Tareas:** Líder de proyecto del área de Business Intelligence. Encargado de dar consultoría y generar nuevas oportunidades de negocios en los diferentes clientes.

Argecard S.A. (Jun. 2003 Ene. 2005)

Sector: IT. **Tareas:** Analista de Business Intelligence. Análisis y diseño de modelos para las diferentes áreas del negocio. Responsable de los procesos ETL.

Intersoft S.A. (Dic. 1999 May. 2003)

Sector: Dpto. de Servicios. **Tareas:** Líder de proyecto. A cargo de proyectos de migraciones, testing, configuraciones e implementaciones en grandes clientes. Análisis, diseño e implementación de herramientas OLAP.

Sector: Producto y Tecnología. **Tareas:** Análisis, programación y mantenimiento del sistema de liquidación. Manejo de base de datos Essentia e Ideafix, Oracle. Compilación y Pruebas de ports.

AFIP-DGI (Sep. 1999 a Nov 1999)

Sector: Grupo Auditores Fiscales de Sistemas, dependiente de la División de Asuntos Fiscales Operativa.
Tareas: Auditoría fiscal de sistemas.

Advance Telecomunicaciones – Telefónica de Argentina (Sep. 1997 a Nov. 1998)

Sector: Gestión y Operación de Redes. **Tareas:** Administración de servidores y redes en ambiente Unix.

5. Artículos en revista con referato

- + Cigliuti, P., Martins, S., García-Martínez, R. 2014. *Proceso de Identificación de Comportamiento de Estudiantes Recursantes Utilizando EVEAs*. Revista Iberoamericana de Educación en Tecnología y Tecnología en Educación, 13: 61-71. ISSN 1850-9959.
- + **Comunicaciones a congresos con referato:**
- + Garbarini, R., Cigliuti, P., Saavedra-Martínez, P., Pollo-Cattaneo, M. F., Rivero-Peña Y. 2018. “Proyecto Plataforma E-Learning para el aprendizaje colaborativo de la Inteligencia de Negocios”. XVII Congreso Internacional de Informática en la Educación “INFOREDU” 2018. Habana, Cuba. Memorias de la XVII Convención y Feria Internacional Informática 2018, Editorial Joven Club, ISBN: 978-959-7255-00-0.
- + Garbarini, R., Cigliuti, P., Saavedra-Martínez, P. 2017. Propuesta de estrategia de gestión de datos maestros en el contexto de la migración a un sistema ERP. Simposio “Informática, Matemática y Ciencias de la Información”, Conferencia Científica Internacional, 8va Edición – Universidad de Holguín, Cuba. Memorias del evento, ISBN 978-959-16-3272-2. <http://eventos.uho.edu.cu/index.php/ccm/cci2017/paper/view/564>
- + Garbarini, R., Cigliuti, P., Osuna, G. 2017. Desarrollo de un sistema de puntos verdes inteligentes como instrumentos de educación ambiental. Simposio “Informática, Matemática y Ciencias de la Información”, Conferencia Científica Internacional, 8va Edición – Universidad de Holguín, Cuba. Memorias del evento, ISBN 978-959-16-3272-

2. <http://eventos.uho.edu.cu/index.php/ccm/cci2017/paper/view/586>

- + Garbarini, R., Gaspes, E., Osuna, G., Cigliuti, P., Saavedra-Martínez, P., Pollo-Cattaneo, M. F., Recchini L. & Rivero-Peña Y. 2016. Construcción y validación de un sistema web/móvil como herramienta de intervención tecnológica para la evaluación dinámica en asignatura de grado en ingeniería en sistemas de información. 4to. Congreso Nacional de Ingeniería Informática / Sistemas de Información. Área Temática: Educación en Ingeniería (EI). ISSN 2347-0372.
- + Garbarini, R., Gaspes, E., Osuna, G. 2016. Sistema de atención temprana de contingencias ambientales para áreas urbanas. IX CONGRESO INTERNACIONAL GEOMÁTICA 2016 – La Habana, Cuba. Memorias de la XVI Convención y Feria Internacional Informática 2016, Editorial Joven Club (959-289), ISBN: 978-959-289-122-7.
- + Garbarini, R., Gaspes, E., Osuna, G., Cigliuti, P., Saavedra-Martínez, P. & Pollo-Cattaneo, M. F. 2015. Construcción de un Sistema de Composteras Inteligentes como instrumentos de educación ambiental. 3er. Congreso Nacional de Ingeniería Informática / Sistemas de Información. Área Temática: Aplicaciones Informáticas y de Sistemas de Información. ISBN 978-987-1896-47-9.
- + Garbarini, R., Gaspes, E., Macri, M., Osuna, G., Cigliuti, P., Saavedra-Martínez, P. & Pollo-Cattaneo, M. F. 2015. Propuesta de Sistema de Composteras Inteligentes como instrumentos de educación ambiental. Proceedings XVII Workshop de Investigadores en Ciencias de la Computación (Tecnología Informática Aplicada en Educación). Artículo 6934. ISBN 978-987-633-134-0.
- + Garbarini, R., Cigliuti, P., Saavedra-Martínez, P., Pollo-Cattaneo, M. F. & Burstyn, A. 2014. Elementos para una Estrategia de Laboratorio Virtual orientado a un Laboratorio Universitario de Ingeniería en Sistemas de Información. Proceedings XVI Workshop de Investigadores en Ciencias de la Computación. Pág. 994-998. ISBN 978-950-34-1084-4.
- + Cigliuti, P., Martins, S., Garcia-Martinez, R. 2014. Identificación del Comportamiento de Estudiantes Basado en Explotación de Información. Memorias IV Jornadas de Enseñanza de la Ingeniería. ISSN 2313-9056.
- + Cigliuti, P., Martins, S., Rodríguez, D., García-Martínez, R. 2014. Identificación de Contenidos

Débilmente Apropriados por Estudiantes Basada en Explotación de Información. Caso de Estudio. XII Workshop Tecnología Informática Aplicada en Educación. Proceedings XX Congreso Argentino de Ciencias de la Computación. Universidad Nacional de la Matanza. ISBN 978-987-3806-05-6.

- ✦ Garbarini, R; Haim, A; Cigliuti, P; Sánchez C; Saavedra Martínez, P. 2013. Evaluación de áreas geográficas adecuadas para la implementación de plantas solares termoeléctricas de torre mediante la utilización de sistemas de información geográfica. 1er Congreso Nacional de Ingeniería Informática / Sistemas de Información. Área Temática: Aplicaciones Informáticas y de Sistemas de Información.
- ✦ Garbarini, R., Cigliuti, P., Pollo Cattaneo, Ma.F, Bursztyn, A. 2013. Implementación de un Sistema de Gestión de Calidad y Servicios en Laboratorio Universitario de Ingeniería en Sistemas de Información. VIII Congreso de Tecnología en Educación y Educación en Tecnología. ISBN 978-987-1676-04-0 Artículo: #5490.
- ✦ Pollo-Cattaneo. M., Pytel, P., García-Martínez, R., Vegega, C., Amatriain, H., Ramón, H., Mansilla, D., Deroche, A., Cigliuti, P., Saavedra-Martínez, P., Garbarini, R., Rodríguez, D., Britos, P., Tomasello, M. (2013). Prácticas y Aplicaciones de Ingeniería de Requisitos en Proyectos de Explotación de Información. Proceedings del XV Workshop de Investigadores en Ciencias de la Computación, Pág. 171-175. ISBN 978-9-872-81796-1
- ✦ Cigliuti, P., Pollo-Cattaneo, F., García-Martínez, R. 2012. Procesos de Identificación de Comportamiento de Comunidades Educativas Centradas en EVEAs. Proceedings del XIV Workshop de Investigadores en Ciencias de la Computación. Pág. 954-956. ISBN 978-950-766-082-5.

- **Disertaciones**

1. II Jornadas Nacionales TIC – Universidad Tecnológica Nacional – Ago. 14 – Tema: Big Data
2. III Congreso Nacional de Sistemas de Gestión y Mejora Continua – UTN-FRBA – Sep. 11 – Tema: Business Intelligence como pieza fundamental de la mejora continua
3. Evento: Microstrategy Users Group – Julio 09. Tema: Caso de éxito de tableros de comando Nobleza Piccardo.

4. IDC Business Intelligence Conference – May. 09 Tema: Estrategia de Información.

• **Seminarios y Cursos:**

- Certified Business Intelligence Professional – Otorgado por TDWI (The Data Warehouse Institute) – Agosto 2014.
- Certified Data Management Professional – Otorgado por DAMA (Data Management Association) – Agosto 2014
- Certificación ITIL OSA – agosto 2011
- Certificación ITIL Foundation v3 – diciembre 2008
- Microstrategy 8
- Advanced Narrowcasting – Dic 06 ● Freeform SQL Essentials – Sep 06 ● Narrowcasting Essentials – Sep 06
- Implementing the Powerful EIS – Sep 06
- Reporting Essentials – Jun 06
- Advanced Reporting – Jun 06
- Project Design – Jun 06

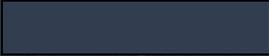
- Curso de introducción a SQLServer 2005. IT College. Noviembre 2005.
- Certificación en SQLServer 2000 – Installing, Configuring and Administering SQLServer 2000 – Junio de 2005
- Capacitación de Business Objects 6.01 básico y para administradores, en todas sus herramientas. Noviembre de 2004, dictado por grupo ASSA. 55 hs



ANEXO O

C.V. Rafael Natera

Contactar



www.linkedin.com/in/rafael-natera-7aa07496 (LinkedIn)

Aptitudes principales

- Technology
- Computer Science
- Computer Hardware

Languages

- Español (Native or Bilingual)
- Inglés (Professional Working)

Certifications

- EF English Test - B2 LvL Score

Rafael Natera

Consultor de desarrollo Oracle Netsuite en 3KSYS SRL

Extracto

IT Senior y Programador de tecnologías WEB. Aficionado de la Electrónica Analógica y Digital

Experiencia

3KSYS SRL
Desarrollador de software - Oracle Netsuite
marzo de 2018 - Present (2 años 2 meses)
Gran Buenos Aires, Argentina
Desarrollador de interfaces de Integración y microservicios (REST WebApi).

Wyndham Concorde Resorte & Spa Isla de Margarita
Sub-Gerente de Sistemas
noviembre de 2013 - febrero de 2018 (4 años 4 meses)
Av. Raul Leoni, Porlamar, Estado Nueva Esparta, Venezuela

Centro Medico Quirurgico La Fe
Proyecto - Coordinador de Migración de Sistemas
noviembre de 2015 - 2016 (1 año)
Jorge Coll Pampatar, Estado Nueva Esparta, Venezuela

Wis de Venezuela
Desarrollador de Aplicaciones Web
enero de 2013 - octubre de 2013 (10 meses)
Av. Aldonza Manrique, Estado Nueva Esparta Venezuela

Educación

Universidad de Oriente (VE)
Licenciado en Informática, Ciencias de la computación · (2007 - 2012)

Cisco Networking Academy
CCNA V4, Networking