

DIVISIÓN DE ESTUDIOS DE POSGRADO E
INVESTIGACIÓN

APLICACIÓN MÓVIL DE APOYO EN LA
IDENTIFICACIÓN DE COLABORADORES
MEDIANTE PERFILES DE CONOCIMIENTO EN
GRUPOS DE DESARROLLO GLOBAL DE
SOFTWARE

TESIS

PRESENTADA POR:
LUIS GUILLERMO CÓRDOVA MORAS

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

DIRECTOR:
DR. ÓSCAR MARIO RODRÍGUEZ ELÍAS

HERMOSILLO, SONORA

AGOSTO, 2017





"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"

SECCIÓN: DIV. EST. POS. E INV.
No. OFICIO: DEPI/227/17.
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN
DE TESIS.

09 de Agosto de 2017

C. LUIS GUILLERMO CÓRDOVA MORAS,
PRESENTE.

Por este conducto, y en virtud de haber concluido la revisión del trabajo de tesis que lleva por nombre "Aplicación móvil de apoyo en la identificación de colaboradores mediante el perfil de conocimiento en grupos de desarrollo global de software" que presenta para el examen de grado de la MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN, y habiéndola encontrado satisfactoria, nos permitimos comunicarle que se autoriza la impresión del mismo a efecto de que proceda el trámite de obtención de grado.

Deseándole éxito en su vida profesional, quedo de usted.

ATENTAMENTE

DR. OSCAR MARIO RODRÍGUEZ ELÍAS
DIRECTOR

M.S.I. MARÍA DE JESÚS VELÁZQUEZ MENDOZA
SECRETARIA

DRA. MARÍA TRINIDAD SERNA ENCINAS
VOCAL

M.C.O. ROSA IRENE SANCHEZ FERMÍN
JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



S.E.P.

RISF/momv*

INSTITUTO TECNOLÓGICO
DE HERMOSILLO
DIVISIÓN DE ESTUDIOS
DE POSGRADO



Dedicatoria

A ustedes, que me han encaminado, en la salud y en la enfermedad, a subir peldaño a peldaño la escalera de mi propia realización personal, dedico este trabajo.

A ustedes, que me han adoctrinado en el auténtico valor de la verdad, que es una sola y que sólo puede erigirse sobre las virtudes de la honestidad, la rectitud y la integridad, dedico este trabajo.

A ustedes, que me han enseñado a hacer aprecio de la humildad y de la solidaridad, y con tal acto me han llevado a comprender la dignidad que yace misteriosa tras la sonrisa que se dibuja sobre un rostro abatido, dedico este trabajo.

A ustedes, papá y mamá, por haber sido el ejemplo al cual me así en incontables ocasiones cuando hube de hacer frente a lo desconocido y a lo incierto, dedico este trabajo y les extiendo eternamente mi gratitud y devoción. ¡Que Dios los bendiga hoy y siempre!

Agradecimientos

Antes de comenzar, me veo en la necesidad de admitir que es bastante probable que me olvide de mencionar expresamente a todas las personas a las que en mi corazón doy gracias por hacer posible este proyecto.

Esta sección es tan sólo un manifiesto en el que reconozco a aquellos que con su consejo, trabajo o compañía me han apoyado en mi trayectoria hacia la obtención del grado de Maestro en Ciencias, y sin embargo hay tantas personas a las que debo agradecer y tantos motivos para estar agradecido, que seguramente no logre colocarlo todo en las siguientes líneas.

Ante todo, doy gracias al Consejo Nacional de Ciencia y Tecnología por la asignación de la beca No. 684383/583429 que me permitió cursar un plan de estudios integrado a su programa nacional de posgrados de calidad, y elogio, de paso, los esfuerzos de todos aquellos miembros de la comunidad científica que habilitan la formación de nuevos investigadores alrededor del mundo. Igualmente, gracias a mi comité directivo, y en especial al director de este trabajo de tesis, el Dr. Óscar Mario Rodríguez Elías, quien junto con el Dr. Ramón René Palacio Cinco sentó las bases para el desarrollo del proyecto.

Qué decir de mis compañeros de estudios, a quienes les tengo un profundo respeto, y de quienes atesoro muchísimos momentos en los que me hicieron contemplar el lado amable de las cosas. Es difícil encontrar un adjetivo que haga justicia a la gran experiencia que ha sido para mí coincidir con ellos.

Por último, agradezco a todos y cada uno de los miembros de mi familia el ser una entrada de apoyo constante en mi vida, en especial a aquellos que no se encuentran más entre nosotros. Doy fe de que su recuerdo es y será un aliciente para la consecución de esta y de cada meta que pueda esperar por mí en el futuro. Gracias por todo. Los amo enormemente.

Resumen

Fortaleciéndose por la apertura e integración socioeconómicas que caracterizan a la era contemporánea, los avances de la globalización hacia una redefinición de la entrega de bienes y servicios se han ido demostrando en una escala internacional: la evidente homogeneización en las filosofías de trabajo globales ha desencadenado el origen de nuevos paradigmas de operación que apuntan, principalmente, a aquellas organizaciones cuya labor se extiende a lo largo de distintos países y culturas.

Dentro del dominio de las tecnologías de información, y en medio de estos nuevos paradigmas, destaca el desarrollo global de software, llamado con frecuencia por su acrónimo anglosajón, *GSD*. Este modelo de trabajo puede consistir en el reparto de la labor de desarrollo de software de una empresa entre distintos equipos de profesionales directamente contratados que radican en diferentes localizaciones a nivel mundial, o bien, en la encomienda del desarrollo parcial o total de sus productos de software a empresas de terceros sub-contratadas alrededor del mundo, derivando, en ambos casos, en un nivel de prestaciones y una rentabilidad objetivamente mayores, en comparación con los valores obtenidos del reclutamiento de recursos humanos a nivel local.

No obstante, pese a sus irrefutables ventajas, es importante que se tengan en consideración los desafíos estratégicos que viene a plantear el *GSD*, puesto que las distancias entre el personal que participa en los proyectos de esta índole obstaculizan su proceso de colaboración, provocando a menudo incidencias relacionadas con cuestiones de liderazgo, sincronización de labores, comunicación y cultura, e imponiendo, en el acto, barreras que dificultan a los trabajadores el acceso a los recursos de conocimiento y a la experiencia explícita de sus compañeros.

Es a causa de los atenuantes impuestos por dichas barreras que surge la necesidad de emplear, tanto una técnica efectiva de gestión de conocimiento organizacional, como una herramienta tecnológica que pueda ser puesta fácilmente al alcance de los trabajadores, procurando que su uso en conjunto asegure el acceso global a los recursos de conocimiento como una realidad tangible para cada programador dentro de las comunidades de desarrollo global de software.

Así pues, las siguientes páginas narran en orden cronológico las etapas comprendidas entre la concepción inicial y la implementación de un proyecto de desarrollo de software que logra asentar un medio para mitigar algunas de las adversidades a las que los programadores se ven obligados a hacer frente en ambientes geográficamente dispersos, gracias a la creación de una aplicación móvil que brinda apoyo en el proceso de identificación de colaboradores en proyectos de desarrollo global de software, cimentada sobre modelos propuestos con anterioridad para la solución de problemas y la toma de decisiones en gestión de conocimiento, mismos que especifican procedimientos claros para la construcción y valoración de perfiles de conocimiento de personal.

Abstract

Strengthened by the socioeconomic opening and integration that characterize the contemporary era, the advances of globalization towards a redefinition of the delivery of goods and services have been demonstrated on an international scale: the evident homogenization in global working philosophies has brought the origin of new operational paradigms that focus, for the most part, on those organizations whose work extends across different countries and cultures.

Within the domain of Information Technologies, and among these paradigms, global software development, often called by its initials, *GSD*, stands out. This model of workforce may either consist of the distribution of a given company's software development tasks between different groups of professionals who are hired directly by the company and live in different locations worldwide, or of the entrustment of partial or total software development of their products through subcontracted third-party companies around the world, ultimately resulting for both cases in an objectively higher level of performance and profitability, in contrast with the values achieved by the recruitment of human resources at the local level.

However, in spite of its irrefutable advantages, it is also important to take into account the strategic challenges that *GSD* is forced to cope with, since the distance between personnel that participates in projects of this nature hinders their collaboration process, often causing incidences and issues related to factors like leadership, synchronization of work schedules, communication and culture, imposing, in the process, barriers that make it difficult for workers to access to knowledge resources and explicit experience of their colleagues.

It is because of the attenuating circumstances imposed by these barriers that the need arises for companies to employ both an effective organizational

knowledge management technique and an easily accessible technological tool for the workers, endeavouring for its use as a whole to ensure global access to knowledge resources as a reality for each programmer within global software development communities.

Thus, the following pages run chronologically through the stages between the initial conception and the implementation of a software development project that manages to establish a means to diminish some of the adversities that programmers have to deal with in geographically dispersed environments, thanks to the creation of a mobile application that supports the process of identifying collaborators in global software development projects, based upon models that were previously proposed for problem solving and decision making in knowledge management, which provide clear procedures for the construction and appraisal of personnel knowledge profiles.

Índice general

Dedicatoria	i
Agradecimientos	ii
Resumen	iii
Abstract	v
1 Introducción	1
1.1 Antecedentes	2
1.2 Definición del problema	5
1.2.1 Planteamiento del problema	5
1.2.2 Preguntas de investigación	5
1.3 Objetivos	6
1.3.1 Objetivo general	6
1.3.2 Objetivos específicos	6
1.4 Justificación	6
1.5 Alcances y delimitaciones	7
1.6 Metodología	8
1.7 Organización de la tesis	9
2 Estado del arte	10
2.1 Desarrollo global de software	11
2.1.1 Globalización	12
2.1.2 Entornos distribuidos	13
2.1.2.1 Offshoring	14
2.1.2.2 Outsourcing	14
2.1.3 Tendencias de desarrollo global de software	14

2.1.3.1	Enfoque de áreas de proceso en proyectos de desarrollo global de software	15
2.1.3.2	Herramientas empleadas en proyectos de desarrollo global de software	19
2.1.4	Áreas de problema y retos en el desarrollo global de software	21
2.2	Gestión de conocimiento	22
2.2.1	Conocimiento tácito	23
2.2.2	Conocimiento explícito y transición del conocimiento tácito	23
2.2.3	Aprendizaje organizacional	24
2.2.4	Estrategias de gestión de conocimiento en entornos de desarrollo global de software	25
2.3	Trabajos relacionados	31
2.3.1	Publicaciones relacionadas	31
2.3.2	Tecnologías relacionadas	33
2.3.2.1	Sistemas colaborativos o groupware	33
2.3.2.2	Sistemas de gestión de conocimiento o KMS	34
2.3.2.3	Sistemas de localización de expertos	34
2.3.3	Herramientas relacionadas	36
2.4	Uso de plataformas móviles	37
2.4.1	Principales plataformas móviles según su cuota de mercado	38
2.4.1.1	Android	38
2.4.1.2	iOS	39
2.4.1.3	Windows Phone	39
2.4.2	Desarrollo de aplicaciones móviles	40
2.4.2.1	Soluciones nativas	40
2.4.2.2	Soluciones de web móvil	41
2.4.2.3	Soluciones híbridas	42
2.4.2.4	Android como plataforma de desarrollo móvil	42
2.4.2.5	iOS como plataforma de desarrollo móvil	42
2.4.2.6	Windows Phone como plataforma de desarrollo móvil	43
2.4.3	Tendencias del desarrollo móvil con impacto en entornos distribuidos	43
2.4.3.1	Mobile 2.0	44

2.4.3.2	Soluciones web progresivas	45
2.5	Conclusiones del estado del arte	45
3	Análisis y diseño del sistema	48
3.1	Contexto de negocio	49
3.1.1	Antecedentes	49
3.1.2	Necesidades del cliente o mercado	50
3.1.3	Supuestos e implicaciones previas a la implementación	50
3.2	Visión de la solución	51
3.2.1	Enunciado de visión de la solución	51
3.2.2	Características primarias del sistema	51
3.2.3	Características secundarias del sistema	52
3.3	Alcance de la solución	52
3.3.1	Alcance por entrega	52
3.3.2	Limitaciones y exclusiones	52
3.4	Contexto del sistema	53
3.4.1	Ambiente de operación	53
3.4.2	Modelado de procesos de negocio	53
3.4.3	Diagrama de contexto	55
3.5	Requerimientos de arquitectura	55
3.5.1	Drivers funcionales y casos de uso	55
3.5.2	Drivers de restricciones	57
3.5.3	Drivers de atributos de calidad en base a escenarios de calidad	57
3.6	Diseño de la arquitectura	59
3.6.1	Estilo arquitectónico seleccionado	59
3.6.2	Patrón arquitectónico seleccionado	60
3.6.3	Componentes identificados y su funcionalidad	62
3.6.3.1	Funcionalidad comprendida por el SVPC	62
3.6.3.2	Funcionalidad comprendida por el SCPC	64
3.6.3.3	Funcionalidad comprendida por la aplicación móvil	65
4	Implementación del sistema	67
4.1	Técnicas base para la implementación de la propuesta	68
4.1.1	Definición de una ontología de conocimiento en desarro- llo y prueba del software	68

4.1.2	Extracción y registro de perfiles de conocimiento	69
4.2	Tecnologías base para la implementación de la propuesta	72
4.2.1	Base de datos de perfiles de conocimiento	72
4.2.2	API REST de gestión de perfiles de conocimiento	74
4.2.3	Servicio de valoración de perfiles de conocimiento o SVPC	75
4.2.3.1	Registro de una nueva valoración	75
4.2.4	Servicio de consulta de perfiles de conocimiento o SCPC	76
4.2.4.1	Empate de perfiles de conocimiento existentes con perfil de necesidad extraído desde la consulta del usuario	77
4.3	Aplicación móvil de identificación de perfiles de conocimiento	78
4.3.1	Selección de conocimientos a integrar en el perfil de necesidad	79
4.3.2	Sumarización de conceptos y uso de filtro geográfico para perfiles de expertos	80
4.3.3	Dictamen y detalle de candidato seleccionado	82
5	Análisis de resultados	84
5.1	Integración de los elementos del sistema	85
5.2	Planteamiento de un escenario de uso factible	87
6	Conclusiones	92
6.1	Apreciaciones complementarias	93
6.2	Contribuciones y aportaciones	94
6.2.1	Discernimiento final	95
6.3	Propuestas de trabajo futuro	96
	Bibliografía	99

Índice de cuadros

Cuadros del estado del arte

I. Tendencias de desarrollo global de software	15
II. Áreas de proceso en proyectos de desarrollo global de software - Gestión de procesos	16
III. Áreas de proceso en proyectos de desarrollo global de software - Gestión de proyecto	16
IV. Áreas de proceso en proyectos de desarrollo global de software - Gestión de ingeniería	17
V. Áreas de proceso en proyectos de desarrollo global de software - Soporte	18
VI. Estrategias de gestión de conocimiento en base a factores de personal, empleando el método de revisión sistemática de literatura	26
VII. Estrategias de gestión de conocimiento en base a factores de personal, empleando el método de entrevista	27
VIII. Estrategias de gestión de conocimiento en base a factores de proyecto, empleando el método de revisión sistemática de literatura	27
VIII. Estrategias de gestión de conocimiento en base a factores de proyecto, empleando el método de revisión sistemática de literatura (2)	28
IX. Estrategias de gestión de conocimiento en base a factores de proyecto, empleando el método de entrevista	28
IX. Estrategias de gestión de conocimiento en base a factores de proyecto, empleando el método de entrevista (2)	29
IX. Estrategias de gestión de conocimiento en base a factores de proyecto, empleando el método de entrevista (3)	30
X. Estrategias de gestión de conocimiento en base a factores tecnológicos, empleando el método de revisión sistemática de literatura	30
XI. Estrategias de gestión de conocimiento en base a factores tecnológicos, empleando el método de entrevista	30

Cuadros de análisis y diseño del sistema

XII. Necesidades del cliente o mercado	50
XIII. Supuestos e implicaciones	50
XIV. Características primarias del sistema	51
XV. Características deseables del sistema	52
XVI. Alcance según versiones de entrega	52
XVII. Detalle de casos de uso	55
XVII. Detalle de casos de uso (2)	56
XVIII. Restricciones de desarrollo	57
XIX. Escenarios de calidad propuestos acorde a la norma ISO/IEC 9126-1	57

Índice de figuras

Figuras de análisis y diseño del sistema

Fig. 1: Modelado de procesos de negocio	54
Fig. 2: Diagrama de contexto	55
Fig. 3: Diagrama de casos de uso	56
Fig. 4: Arquitectura conceptual	61
Fig. 5: Despliegue de componentes del SVPC de (Rosas-Daniel, 2014)	62
Fig. 6: Diagrama de clases del SVPC y FIS de (Rosas-Daniel, 2014)	63
Fig. 7: Diagrama de clases del SCPC	64
Fig. 8: Diagrama de secuencia del sistema	65

Figuras de implementación del sistema

Fig. 9: Categorías y términos distinguidos por la ontología	69
Fig. 10: Modelo de construcción de perfiles de conocimiento	70
Fig. 11: Creación de un perfil de conocimiento	71
Fig. 12: Esquema relacional de la base de datos de perfiles de conocimiento	73
Fig. 13: Función del SVPC para asignar una valoración al perfil de un candidato	76
Fig. 14: Función del SCPC para dar inicio a una búsqueda de conocimiento	77
Fig. 15: Función del SCPC para encontrar perfiles candidatos según parámetros establecidos	78
Fig. 16: Vista de categorías y conceptos	79
Fig. 17: Vista de resumen de selección	80
Fig. 18: Cuadro de diálogo de aplicación de filtro de localización	81
Fig. 19: Vista de candidatos disponibles	82

Figuras de análisis de resultados

Fig. 20: Esquema de integración de elementos	86
Fig. 21: Escenario A	88
Fig. 22: Escenario B	89
Fig. 23: Escenario C	90

Capítulo 1

Introducción

La ingeniería de software es una disciplina basada íntegramente en el conocimiento, debido a que, durante el curso de cada una de las etapas que conforman sus procesos, la información que en ellos se gestiona atraviesa una serie de transformaciones consecutivas, hasta que finalmente se produce el resultado deseado y el conocimiento llega, entonces, a su consolidación.

No obstante, la construcción del conocimiento por sí misma no garantiza su difusión, y no es en absoluto raro llegar a observar cómo, en ambientes organizacionales, una mala gestión del conocimiento, particularmente del registro y clasificación del mismo, puede llevar a una empresa a invertir, innecesariamente, recursos de tiempo, dinero y esfuerzos para así lograr efectuar el mismo descubrimiento en más de una ocasión. (Dalkir, 2005)

Esto suele darse, con más frecuencia, en ambientes de trabajo distribuidos, donde la propia constitución del entorno, en el que coadyuvan personas desde distintas zonas geográficas, crea una atmósfera que propicia que las organizaciones experimenten, con mayor fuerza, los efectos de las malas prácticas concernientes a la gestión del conocimiento. (Nidhra et al., 2012)

Es por tanto que el desarrollo de esta tesis gira en torno a la creación de una herramienta que permite distinguir, dentro de entornos de desarrollo global de software, a aquellos individuos capaces de aportar conocimiento a los usuarios que así lo precisen, valiéndose para ello del modelo de valoración de perfiles de conocimiento creado por (Rosas-Daniel et al., 2015), el cual, a

través de técnicas de lógica difusa, logra especificar en qué medida un individuo cumple con los requerimientos asociados al perfil de conocimiento definido para un determinado puesto dentro de una organización.

El presente capítulo describe los antecedentes de la gestión del conocimiento, del desarrollo global de software y de las herramientas y técnicas que competen a ambos. Una vez descritos dichos apartados, el problema es definido y contextualizado, y las preguntas y objetivos de investigación, tanto generales como específicos, son establecidos.

Posteriormente, se presenta una justificación de la investigación realizada, junto con su alcance y delimitaciones, indicando cuál fue la metodología empleada durante la misma.

Por último, se distingue la organización de la tesis: un apartado en el que se redacta la estructura de cada uno de los capítulos que componen este documento, junto con una breve explicación de sus contenidos.

1.1. Antecedentes

El conocimiento es el conjunto de experiencias, ideas y percepciones que, al ser analizadas y organizadas, pueden ser aplicadas en la resolución de problemas. Es el resultado de la asimilación de información por parte de un individuo que, según las necesidades de su entorno, emplea lo aprendido en las actividades que realiza habitualmente. (Alavi y Leidner, 2001)

En el ámbito organizacional, el conocimiento constituye uno de los activos más importantes, pues aquella empresa que logra construirlo y gestionarlo con éxito crea riqueza y/o valores añadidos para sí misma, lo cual suele llevarla a alcanzar posiciones ventajosas en el mercado. (Nieves-Lahaba y León-Santos, 2001)

Se define, pues, como perfil de conocimiento: «Aquel conjunto de rasgos estructurados que describen el conocimiento requerido, asociado a los recursos y capacidades, que permitan habilitar la generación dinámica de competencias laborales, competencias de los procesos clave y competencias distintivas que generen valor a la organización.» (Velázquez-Mendoza, 2013)

En el desarrollo de software, contar con acceso al conocimiento apropiado en el momento apropiado, es un factor crucial que puede, incluso, resultar decisivo en el fracaso o éxito de un proyecto. Esto se hace evidente al considerar que el principal problema en torno al conocimiento en los procesos de desarrollo de software radica en que sólo se suele disponer explícitamente de una pequeña fracción de él, siendo éste mayoritariamente tácito y compuesto por los conocimientos repartidos entre los actores principales de dichos procesos, quienes a menudo no cuentan con ninguna noción respecto a la experiencia profesional del resto de individuos que conforman su equipo de trabajo. (Jelínek, 2012)

Lo anterior representa un reto especialmente grande para los equipos que laboran bajo el paradigma del desarrollo global de software, el cual responde a un proceso de desarrollo de productos de software llevado a cabo por distintos equipos de personal, situados en distintas ubicaciones geográficas, pudiendo pertenecer éstos a empresas de terceros o a la misma organización responsable de la implementación del producto. (Aranda, 2009)

Si bien, dicha distancia entre equipos es establecida con el fin de abaratar costos de mano de obra y proveer acceso a una serie de ventajas de peso considerable, es importante señalar que igualmente introduce diversos desafíos relacionados con la falta de comunicación presencial, diferencias en el idioma y las costumbres de quienes participan en estos proyectos, entre otras cuestiones. (Aranda, 2009)

Se distinguen en los entornos de desarrollo global de software dos principales corrientes: el Offshoring y el Outsourcing. La primera toma lugar cuando una organización establece una división de sí misma en una localización remota, distribuyendo sus procesos de desarrollo entre dicha división y su sede, mientras que la segunda se da cuando una organización subcontrata y encomienda su labor de desarrollo al personal de una organización de terceros, ya sea parcial o totalmente. (Richardson et al., 2012)

Estas tendencias aparecieron hace más de tres décadas, cuando algunas de las empresas líderes en tecnologías de información americanas comenzaron a subcontratar personal en países como India, Irlanda e Israel en los años 70. Se le atribuye el liderazgo ante este suceso a India, al lograr producir un 35 % más de empleos, año con año, durante la década comprendida de 1998 a 2008, convirtiéndose en el país con mayor cantidad de personal subcontratado a nivel

mundial actualmente, tanto en empleos en general como en lo que a empleos relacionados con la gestión de tecnologías información se refiere. (Mullan et al., 2008)

Así pues, como fruto de la globalización, cada vez es más común observar cómo los proyectos, en ingeniería de software, son desarrollados en conjunto por equipos ubicados en distintas localidades. A pesar de que, bajo el panorama más optimista, el desarrollo global de un proyecto de software “simple”, es tan complicado como el de los proyectos más complejos llevados a cabo en forma local, aún se echan en falta estándares, metodologías y herramientas que respalden este modelo. (Galvina y Smite, 2011)

El uso de una tecnología en pleno auge, como lo es el desarrollo de software para plataformas móviles, podría marcar una inflexión a través de la cual sería posible encontrar una solución en común a algunos de los problemas vinculados a los entornos de trabajo distribuidos. Este hecho adquiere mayor sentido si tomamos en cuenta que el uso de los smartphones, con todas sus implicaciones, ha logrado tal repercusión en nuestra vida diaria que, recientemente, se ha comparado su impacto con el de la llegada de la energía eléctrica a los hogares durante el siglo pasado. (Salesforce Marketing Cloud, 2014)

Podemos aseverar, pues, que en un futuro más bien próximo, las plataformas móviles y, especialmente, el desarrollo de software para plataformas móviles, proporcionará un punto de entrada, o bien, jugará un papel clave en la resolución de problemas en los que la gestión de conocimiento se vea comprometida, dentro de todo tipo de ambientes laborales.

Asimismo, la creación y mantenimiento de una base de perfiles de conocimiento se postula, en este trabajo, como una práctica que promete respaldar a los grupos de programadores que operan en conjunto desde distintas partes del mundo, al perfilar las áreas de competencia de cada participante individualmente, elaborando con ellas una lista de conocimientos asociados a cada programador en un intento por asentar un medio para facilitar la búsqueda de colaboradores potenciales a los cuales dirigirse, según qué conocimiento sea requerido.

Cabe destacar que, previo a la elaboración de esta tesis, se desarrolló un modelo y una herramienta de software que, mediante el uso de lógica difusa, busca valorar perfiles de conocimiento, pudiendo identificar al más capacitado

entre ellos acorde al criterio establecido. Subsecuentemente se implementó un prototipo para validar el funcionamiento del modelo construido, utilizando para ello perfiles de muestra.

La aplicación propuesta emplea, pues, dicho modelo con el objetivo de ayudar a aminorar este impacto, ya que la clasificación de perfiles de conocimiento dentro de equipos de trabajo preestablecidos, sean éstos geográficamente distribuidos o co-localizados, puede constituir una técnica de gran utilidad en la búsqueda de personal capacitado para la solución de problemas que competan al desarrollo y prueba del software.

1.2. Definición del problema

Tal como se mencionó anteriormente, uno de los mayores retos al que se enfrenta el personal que labora en entornos de desarrollo global de software, consiste en distinguir qué personas se encuentran cualificadas para brindarles soporte en determinadas tareas de desarrollo y de qué manera pueden hacerlo, dada la carencia de un medio para identificar y establecer contacto con el personal experto.

En consecuencia, se hace presente la necesidad de emplear herramientas, de corte tecnológico, que ayuden a los integrantes a mitigar los problemas asociados a sus entornos de trabajo.

1.2.1. Planteamiento del problema

¿Qué características debería tener una aplicación móvil de apoyo en la identificación de colaboradores, mediante el perfil de conocimiento, para ayudar a solucionar algunos de los problemas de comunicación que, por lo general, se presentan en los entornos de desarrollo global de software?

1.2.2. Preguntas de investigación

- ¿Cómo facilitar el acceso a personas que puedan ayudarnos a resolver problemas en el marco del desarrollo global de software, mediante la

identificación de las personas adecuadas en función de sus perfiles de conocimiento?

- ¿Qué características debe reunir el desarrollo de una herramienta móvil para poder usar estos mecanismos, y así apoyar en la colaboración entre equipos de desarrollo global de software?
- ¿Qué beneficios pudiera proveer dicha herramienta?

1.3. Objetivos

1.3.1. Objetivo general

Diseñar e implementar una aplicación para dispositivos móviles, que apoye en la identificación de alternativas de ayuda en la solución de problemas, dentro de equipos de desarrollo global de software, con base en el perfil de conocimiento de los miembros del equipo.

1.3.2. Objetivos específicos

- Revisar y analizar la literatura en desarrollo global de software y sus retos.
- Diseñar e implementar la aplicación.
- Evaluar mediante un caso práctico hipotético o real.
- Identificar qué tipos de problemáticas pueden ser atendidas a través del uso de la aplicación.

1.4. Justificación

Compartir información entre miembros de un mismo equipo supone mayor dificultad cuando éstos carecen de un medio de comunicación simultánea, o

bien, no llegan a coincidir los unos con los otros en ningún momento durante sus horas de trabajo. (Prikladnicki et al., 2003)

Si nos transportamos a los entornos de desarrollo distribuidos donde, de entrada, ambos casos se cumplen, es incluso fácil comprender por qué las tareas, por regla general, toman mucho más tiempo para ser completadas en ellos, en comparación con aquellas tareas similares realizadas en entornos co-localizados. (Faßbinder y Henz, 2009)

Es por tal motivo que, la experiencia obtenida de la participación en estos últimos, no representa un respaldo suficiente para poder estimar correctamente la duración de las actividades en un proyecto distribuido.

De lo anterior concluimos que, dentro de un grupo de desarrollo global de software, una mala gestión del conocimiento resultará, irremediablemente, en un impacto en la gestión de los tiempos, no sólo de un proyecto, sino de todo el portafolio del grupo, hecho que implica, en consecuencia, periodos de desarrollo más largos para cada proyecto y, por ende, una productividad que seguramente no corresponda con el verdadero potencial del equipo. (Nidhra et al., 2012)

1.5. Alcances y delimitaciones

- Se buscará que el prototipo a desarrollar funcione en más del 50 % de los Smartphones comerciales, por lo que la plataforma destino será Google Android.
- Para el desarrollo del prototipo se hará uso del modelo de lógica difusa para validación de perfiles de conocimiento de personal desarrollado por (Rosas-Daniel, 2014).
- La validación del prototipo se realizará a través de una prueba de laboratorio. Para ello se planteará un caso hipotético basado en un escenario real como precedente.
- Por otra parte, la experimentación del prototipo se llevará a cabo en el laboratorio de Ingeniería de Software de la Maestría en Ciencias de la Computación, perteneciente a la División de Estudios de Posgrado e

Investigación del Instituto Tecnológico de Hermosillo, y para tal fin se hará uso del equipo de cómputo ubicado en dicho laboratorio.

- Es importante aclarar que existen otros factores de importancia dentro de los entornos de desarrollo global de software, tales como las barreras culturales, el miedo a expresar desconocimiento, la falta de habilidades interpersonales, entre otros temas relacionados con la gestión de proyectos que no serán considerados en esta etapa, dentro de la solución propuesta.

1.6. Metodología

La metodología del presente trabajo de tesis abarca 6 etapas: la introducción a la concepción del proyecto, tal como se reseña en el actual capítulo, el estado del arte, el análisis y diseño del sistema, la implementación del prototipo, el análisis de los resultados obtenidos y la entrega de conclusiones.

La elaboración del estado del arte ha comprendido el estudio y análisis de temas alusivos a los procesos de desarrollo global de software, entre los cuales destacan la gestión de perfiles de conocimiento, el estudio de las áreas de proceso en entornos distribuidos y sus objetivos, así como también, las herramientas empleadas en dichos entornos, tomando como fundamento algunos casos de estudio reales.

Durante la etapa de análisis del sistema se identificaron los componentes principales de la aplicación, se establecieron sus requerimientos funcionales y se definieron sus objetivos particulares.

Más adelante, entrada la fase de diseño, la información recolectada en la etapa de análisis fue empleada para modelar los componentes de la aplicación, siendo éstos entidades que emiten o reciben información, o procesos que ejecutan acciones y son llamados por alguna de estas entidades.

Pasado este punto, se implementó un prototipo para dispositivos móviles, con el objetivo de lograr identificar a aquellas personas que, dentro de entornos de desarrollo global de software, pudiesen colaborar en la ejecución de una tarea en concreto, para lo cual el modelo de lógica difusa para valoración de

perfiles de conocimiento de personal se integró a una API web con el fin de ser consumido como un servicio externo.

Finalmente se realizaron pruebas de funcionalidad en un intento por validar la aplicación apropiada de dicho modelo, partiendo del análisis de los resultados obtenidos mediante el uso del prototipo, y se procedió a emitir las conclusiones correspondientes.

1.7. Organización de la tesis

La presente tesis cuenta con un total de seis capítulos, donde el capítulo I y actual, funge como una introducción al trabajo de investigación y a su posterior propuesta de aplicación.

El capítulo II introduce el estado del arte de los temas principales sobre los cuáles se fundamenta el proyecto de tesis. Entre ellos pueden observarse la gestión del conocimiento, el uso de plataformas móviles y el desarrollo global de software junto con sus áreas de investigación y sus mecanismos de apoyo más comunes. Adicionalmente, este capítulo enlista y describe algunas publicaciones, tecnologías y herramientas que guardan similitudes con la solución propuesta.

El capítulo III corresponde al análisis y diseño del sistema, en el que se delimitaron los alcances del prototipo a crear, junto con sus componentes principales, y se describieron los criterios a cumplir a manera de requisitos.

El capítulo IV abarca la implementación del sistema, durante la cual se creó el prototipo de la aplicación, con el fin de probar el modelo de valoración de perfiles de conocimiento en un dispositivo móvil.

El capítulo V, titulado “Análisis de resultados”, incluye un estudio más profundo de los resultados del capítulo anterior, es decir, de la implementación del sistema descrito en este documento.

Para dar por terminado el trabajo de tesis, el capítulo VI refleja las conclusiones y el trabajo futuro, dando a conocer las reflexiones finales resultantes de la investigación, al igual que los posibles enfoques de trabajo que puedan surgir como continuación de la misma.

Capítulo 2

Estado del arte

Este capítulo define, a modo de guía, algunos de los principales términos de investigación vinculados a las áreas de conocimiento que conciernen al actual proyecto de tesis, tomando como referencia la literatura expuesta en trabajos relacionados.

Por ende, se hace patente que la intención del presente estado del arte es dar respuesta a aquellas interrogantes suscitadas durante la concepción de la propuesta, en forma previa al trabajo de investigación, tales como:

- ¿Qué aspectos deben ser tomados en cuenta para poder transmitir conocimiento en los entornos de desarrollo global de software?
- ¿Qué influencia tienen los procesos y herramientas actuales de gestión de conocimiento en los entornos distribuidos?
- ¿Qué implicaciones tendría el uso de plataformas móviles en la construcción de una herramienta para la identificación y valoración de perfiles de conocimiento?
- ¿Cómo puede emplearse el modelo actual de valoración de perfiles de conocimiento para la consecución de una solución eficiente que, a través de dicho modelo, permita localizar personal experto en entornos de desarrollo global de software?

2.1. Desarrollo global de software

El desarrollo global de software, como parte de la globalización, es una filosofía que alude a un mismo proceso de desarrollo realizado en conjunto desde distintos sitios del mundo, involucrando como factores clave la comunicación, la coordinación de actividades y el control de grupos. (Mohagheghi, 2004)

Se trata de un fenómeno cuya importancia se encuentra en ascenso, pues la constante necesidad de las grandes empresas por mantener su modelo de negocio rentable y competitivo dentro del panorama mundial, obteniendo el mayor costo-beneficio posible, hace que estas contemplen emplear mano de obra desde el extranjero, a un costo menor que el local. Tan sólo a principios de la pasada década, alrededor de 50 países ya se encontraban participando en algún proyecto de esta índole. (Ågerfalk et al., 2008)

El desarrollo global de software involucra tres tipos de distancias de mayor distinción (Ågerfalk et al., 2005):

- **Distancia geográfica:** Incide en la comunicación entre los miembros de un equipo dentro de un entorno distribuido, pues impide que éstos tengan una conversación presencial, salvo que se desplacen hacia el lugar donde residen los demás integrantes.
- **Distancia temporal:** Se refiere a la diferencia entre las zonas horarias correspondientes a la zona geográfica en la que se sitúa cada parte del equipo: a mayor distancia temporal, menor será la cantidad de horas de trabajo simultáneo compartido, de manera que la mayor parte de la comunicación entre ambas partes será, forzosamente, asíncrona. Esto afecta la coordinación de procesos entre los integrantes del equipo.
- **Distancia sociocultural:** Abarca tanto las potenciales diferencias de nacionalidad, como también las diferencias culturales que pueda haber en cada organización de manera individual, mismas que afectan los procesos de control dentro de un proyecto.

En este contexto, un solo equipo puede estar separado por estas 3 distancias, convirtiéndose en lo que a menudo se denomina “equipo virtual”. En algunos casos, el equipo puede contar con todos sus recursos de personal co-ubicados en una misma localidad, pero a la vez depender mayoritariamente

del soporte brindado por otros equipos conformados por personas con las que trabajan remotamente.

2.1.1. Globalización

El término “globalización” es empleado para describir un proceso de integración continua por el que atraviesan gran parte de los países del mundo a día de hoy, y el cual involucra un crecimiento en las transacciones y la interdependencia dada entre dichos países. (Scholte, 2008)

Respaldado tanto por la aceleración del progreso tecnológico, producto de la liberalización del comercio, como por la creciente importancia de las leyes supranacionales, este controversial fenómeno ha puesto en marcha una competencia mucho más rigurosa que nunca entre las distintas economías del mundo. (Mrak, 2000)

Una de las características clave del ámbito de las tecnologías de información, a principios del siglo XXI, fue precisamente su tendencia hacia la globalización, la cual también ha venido incrementando, desde entonces, la conectividad y la integración política, cultural, social, económica y tecnológica, tanto entre naciones, como entre corporaciones e individuos. Sin embargo, el impacto de la globalización en el desarrollo de los sistemas informáticos ha demostrado ser particularmente contundente. (Cho, 2007)

Esto se debe en gran medida a la forma en que el acceso a internet ha logrado penetrar en los lugares más recónditos del planeta, transformando el estilo de vida de más personas, en más sitios, de forma menos costosa, y manteniendo un ritmo de cambio más acelerado que el que hubiese sido pronosticado, aún como resultado de la predicción más optimista, hace dos décadas. (Naím, 2009)

La tendencia más reciente de la globalización en entornos organizacionales consiste en la subcontratación de servicios técnicos, administrativos y de tecnologías de la información, en el extranjero, desde países desarrollados hacia países en vías de desarrollo. (Aspray et al., 2006)

La cantidad de servicios que hoy día pueden estar sujetos a subcontratación es considerable, y esto puede resultar especialmente conveniente para las

economías emergentes, pues representa, de cara a las empresas subcontratadas, una oportunidad para incrementar su productividad al instruirse en el uso de las técnicas utilizadas en empresas con mayor desarrollo económico. (Mullan et al., 2008)

2.1.2. Entornos distribuidos

Las organizaciones dedicadas al desarrollo de software encaran, en la actualidad, un cambio ligado a un paradigma emergente: la distribución de los procesos y equipos de desarrollo. (Palacio et al., 2009)

Según diversas fuentes, este cambio de enfoque se debe a numerosas razones (Palacio et al., 2011):

- Las compañías que contemplan al software y/o a su desarrollo como su activo central, precisan de un flujo constante de personal experto, y buscan suplir esta necesidad al emplear programadores de distintas ciudades y países.
- Con el fin de mantenerse más cerca del mercado del consumidor y lograr tener un tiempo de respuesta más corto al interactuar con él, muchas compañías establecen grupos de desarrollo cercanos a la localización de sus clientes.
- Los grupos de desarrollo virtuales pueden formarse, potencialmente, más rápido que los grupos co-ubicados, al acceder a una base más amplia de trabajadores potenciales.
- Al cubrir distintas zonas horarias simultáneamente, las compañías adquieren la capacidad de trabajar en sus proyectos críticos durante más horas de forma continua, lo que las lleva a obtener menores tiempos de entrega.
- Los costos de reclutamiento de recursos humanos se reducen cuando se incluye a personal que reside en países donde la mano de obra es más asequible.

Dentro de los entornos de desarrollo global de software existen dos principales vertientes: el Offshoring y el Outsourcing.

2.1.2.1. Offshoring

Se habla de offshoring cuando una organización establece una división de sí misma en una localización remota, distribuyendo su función entre dicha división y su sede. (Richardson, 2007)

Este método, cuyo nombre en ocasiones es traducido como “deslocalización”, implica el cambio de localización de un servicio o proceso desde un sitio hacia otro. Muchas empresas logran reducir los costos de sus servicios de atención al cliente al llevarlos desde su país hacia otro en el cual la mano de obra resulta más asequible. (Mudambi y Venzin, 2010)

2.1.2.2. Outsourcing

Se habla de outsourcing cuando una organización hace uso del personal de otra, cediendo el cargo de la ejecución de algunos de sus procesos técnicos, administrativos y/o de aquellos que están relacionados con el manejo o gestión de tecnologías de información. (Richardson, 2007)

En entornos distribuidos, este método, cuyo nombre en ocasiones es traducido como “externalización”, implica la subcontratación de personal de otros países como apoyo en la realización y entrega de servicios, ya sea que dichos elementos de personal se encarguen parcial o totalmente de las tareas comprendidas dentro de uno o varios de los procedimientos que forman parte de la lógica de negocios de la empresa. (Mudambi y Venzin, 2010)

2.1.3. Tendencias de desarrollo global de software

Las compañías continúan expandiéndose globalmente, deslocalizando, externalizando, y adquiriendo equipos de trabajo por todo el mundo. A medida que la globalización se hace prevalente, muchas compañías han ido actualizando sus enfoques y prácticas, demostrando que el modelo de desarrollo distribuido comienza a dar signos de madurez. (Fryer y Gothe, 2008)

La tabla I incluye algunas de los últimos cambios que se han podido atestiguar en las tendencias del desarrollo global de software:

I. Tendencias de desarrollo global de software (Fryer y Gothe, 2008)	
Tendencia anterior	Nueva tendencia
Los equipos subcontratados radican principalmente en India	Los equipos subcontratados radican en distintas partes del mundo
La mano de obra subcontratada es más accesible económicamente	La subcontratación supone el acceso a una mayor base de empleados y competencias
Se tienen subcontratados numerosos proveedores	Proveedores subcontratados estratégicamente mediante outsourcing
Las tareas para las que se subcontrata personal son predominantemente tareas de mantenimiento	Nuevo ciclo de vida de desarrollo: diseño y pruebas
Las tareas para las que se subcontrata personal son predominantemente tareas de programación	Se subcontrata personal para consultoría, procesos de negocio, asistencia en el diseño de la arquitectura del software e infraestructura
Entrega global de servicios como enfoque especial	Entrega global de servicios como enfoque estándar

2.1.3.1. Enfoque de áreas de proceso en proyectos de desarrollo global de software

Una forma de catalogar las soluciones identificadas en los procesos de ingeniería de software global es separar dichos procesos en áreas. De esta forma se cuenta con una categorización de procesos que resulta de gran utilidad cuando se busca la contribución de parte de los miembros hacia la comunidad, pues éstos logran identificar con mayor facilidad las áreas de proceso en las que se consideran aptos para contribuir con el trabajo de otros miembros.

Para ello es favorable delinear una estructura de árbol denominada “modelo de áreas de proceso”, también conocida como PAM (*Process Area Model*) por su acrónimo en inglés. Estas otorgan dos perspectivas de importancia en torno al conocimiento de la empresa: la primera permitiría conocer qué áreas de proceso se encuentran bien investigadas y fundamentadas, y cuáles requieren un estudio más extensivo; la segunda permitiría a las empresas utilizar el

PAM como un marco de referencia para evaluar su ambiente de desarrollo al encontrar soluciones de modo estructurado. (Schneider et al., 2013)

Un PAM contiene varias áreas de proceso, cada una de las cuáles cuenta con una o varias sub-áreas, las cuáles a su vez pueden o no contener otras sub-áreas anidadas. Así, un ejemplo de una estructura de PAM para un proyecto de desarrollo global genérico podría incluir, sin limitarse, a las siguientes áreas de proceso:

II. Áreas de proceso en proyectos de desarrollo global de software (Schneider et al., 2013)	
Área de proceso 1 - Gestión de procesos	
Área	Sub-área
AP1.1 – Entrenamiento organizacional	AP1.1.1 – Establecimiento de la capacidad de entrenamiento organizacional
	AP1.1.2 – Impartición del entrenamiento necesario
AP1.2 – Definición de procesos organizacionales	AP1.2.1 – Establecimiento de activos de procesos organizacionales
	AP1.2.2 – Gestión de desarrollo integrado de productos y procesos
AP1.3 – Innovación organizacional y despliegue	AP1.3.1 – Selección de mejoras
	AP1.3.2 – Implementación de mejoras

III. Áreas de proceso en proyectos de desarrollo global de software (Schneider et al., 2013)	
Área de proceso 2 - Gestión de proyecto	
Área	Sub-área
AP2.1 – Planeación de proyecto	AP2.1.1 – Establecimiento de estimados
	AP2.1.2 – Desarrollo de un plan de proyecto
	AP2.1.3 - Asegurar compromiso hacia el plan
AP2.2 – Monitoreo y control del proyecto	AP2.2.1 – Monitoreo de proyecto contra plan
	AP2.2.2 – Gestionar acciones correctivas para cierre

III. Áreas de proceso en proyectos de desarrollo global de software (Schneider et al., 2013)	
Área de proceso 2 - Gestión de proyecto	
Área	Sub-área
AP2.3 – Proceso de administración	AP2.3.1 – Utilizar el proceso definido del proyecto
	AP2.3.2 – Coordinar y colaborar con los stakeholders más relevantes
	AP2.3.3 – Aplicar principios de desarrollo integrado de productos y procesos
	AP2.3.4 – Gestión cuantitativa de proyecto
	AP2.3.5 – Gestión estadística de desempeño de subprocesos
	AP2.3.6 – Iniciación de proyecto
	AP2.3.7 – Evaluación de portafolio
	AP2.3.8 – Cierre de proyecto
	AP2.3.9 – Establecer gestor de seguridad
AP2.4 – Proceso de documentación	AP2.4.1 – Diseño y desarrollo
	AP2.4.2 – Producción
	AP2.4.3 - Mantenimiento
AP2.5 – El equipo distribuido y la gestión del proyecto	AP2.5.1 – Gestión de formación de equipos
	AP2.5.2 – Coordinación / Comunicación
	AP2.5.3 – Motivación
	AP2.5.4 – Conciencia
	AP2.5.5 - Gestión del proyecto distribuido

IV. Áreas de proceso en proyectos de desarrollo global de software (Schneider et al., 2013)	
Área de proceso 3 - Ingeniería	
Área	Sub-área
AP3.1 – Gestión y desarrollo de requerimientos	AP3.1.1 – Desarrollar requerimientos del cliente
	AP3.1.2 – Desarrollar requerimientos de producto
	AP3.1.3 - Analizar y validar requerimientos
	AP3.1.4 - Gestionar requerimientos

IV. Áreas de proceso en proyectos de desarrollo global de software (Schneider et al., 2013)	
Área de proceso 3 - Ingeniería	
Área	Sub-área
AP3.2 – Soluciones técnicas	AP3.2.1 – Seleccionar soluciones de componentes de producto
	AP3.2.2 – Desarrollar el diseño
	AP3.2.3 - Implementar el diseño del producto
AP3.3 – Integración de producto	AP3.3.1 – Preparar integración del producto
	AP3.3.2 – Asegurar compatibilidad de interfaces
	AP3.3.3 – Ensamblar componentes de producto y entregar el producto
AP3.4 – Validación	AP3.4.1 – Preparación de proyecto para validación
	AP3.4.2 – Validación de producto o componentes de producto

V. Áreas de proceso en proyectos de desarrollo global de software (Schneider et al., 2013)	
Área de proceso 4 - Soporte	
Área	Sub-área
AP4.1 – Análisis causal y resolución	AP4.1.1 – Determinación de causas de defectos
	AP4.1.2 – Abordar causas de defectos
AP4.2 – Análisis de decisiones y resolución	AP4.2.1 – Evaluar alternativas
AP4.3 – Afianzamiento de calidad en el proceso y el producto	AP4.3.1 – Evaluación objetiva de procesos y productos de trabajo
	AP4.3.2 – Proveer visión objetiva
AP4.4 – Proceso de adquisición	AP4.4.1 – Preparación de adquisiciones
	AP4.4.2 – Selección de proveedores
	AP4.4.3 – Monitoreo de proveedores
	AP4.4.4 – Aceptación del cliente
	AP4.4.5 - Especificación de seguridad

V. Áreas de proceso en proyectos de desarrollo global de software (Schneider et al., 2013)	
Área de proceso 4 - Soporte	
Área	Sub-área
AP4.5 – Proceso de mantenimiento	AP4.5.1 – Análisis de problemas y modificaciones
	AP4.5.2 – Implementación de modificaciones
	AP4.5.3 – Revisión/aceptación de mantenimiento
	AP4.5.4 - Migración
AP4.6 – Proceso de revisión en conjunto	AP4.6.1 – Revisiones de gestión de proyecto
	AP4.6.2 – Revisiones técnicas

2.1.3.2. Herramientas empleadas en proyectos de desarrollo global de software

Algunas de las herramientas de colaboración empleadas con frecuencia en los entornos de desarrollo global de software distribuidos son (Lanubile et al., 2010):

- **Sistemas de control de versiones:** En los entornos de desarrollo global es preciso contar con una configuración sistemática que permita a los desarrolladores leer, modificar y compartir artefactos de software de modo controlado. SVN, Git y Mercurial, son ejemplos de herramientas que permiten gestionar los cambios que se realizan sobre el software en forma segura.
- **Rastreadores de errores:** Son herramientas de supervisión empleadas para manejar las incidencias que puedan presentarse en la codificación e implementación del software, sean errores, cambios o peticiones de soporte. Su funcionalidad se centra en una base de datos a la que todo el equipo tiene acceso a través de internet, y donde todas las incidencias encontradas se ven representadas.

- **Herramientas de compilación:** Mientras más distribuido se encuentre el proyecto, mayor se vuelve la necesidad de contar con un medio para compilar el código remotamente en forma segura. Maven y CruiseControl son ejemplos de herramientas de compilación que permiten crear y programar flujos de trabajo sobre un determinado repositorio remoto, facilitando la ejecución de scripts, la compilación de binarios, el uso de frameworks para la aplicación de pruebas, el despliegue de versiones del sistema y el envío de notificaciones automáticas vía email a los colaboradores.
- **Herramientas de modelado:** La colaboración basada en modelos es lo que distingue a la ingeniería de software colaborativo de las actividades de colaboración más generales, las cuales consisten en compartir archivos, mas no en compartir contenido. Las herramientas de modelado colaborativo ayudan a los desarrolladores a crear artefactos formales de software, empleando UML u otros lenguajes de modelado para la representación de procesos de software.
- **Centros de conocimiento:** Estos son sistemas de gestión de contenido que permiten a los miembros del equipo de desarrollo compartir conocimiento explícito en la web. Un centro de conocimiento suele contener documentos internos, referencias técnicas, estándares, repositorios de preguntas frecuentes y guías para la aplicación de buenas prácticas. Algunos ejemplos son KnowledgeTree y el sistema de ayuda incluido en el popular IDE Eclipse.
- **Herramientas de comunicación:** Los ingenieros de software han adoptado una amplia gama de tecnologías de comunicación para dar soporte a proyectos donde la interacción directa, por causas circunstanciales, no se considera una opción viable. Entre las herramientas de comunicación asíncronas se encuentran el correo electrónico, los portales de noticias y los foros web, mientras que las herramientas de comunicación sincronizada son más tradicionales. Entre ellas se encuentran las llamadas telefónicas estándar, los sistemas de mensajería instantánea, y más recientemente, las llamadas de voz sobre IP y las reuniones virtuales a través de videoconferencias.

- **Aplicaciones de Web 2.0:** También llamada “Web Social”, la Web 2.0 aplica los principios del software colaborativo basándose en la contribución directa por parte del usuario. Entre las aplicaciones clave de Web 2.0 se distinguen los blogs, los micro-blogs, las wikis, las redes sociales y los sistemas de etiquetado colaborativo.

2.1.4. Áreas de problema y retos en el desarrollo global de software

Aunque los beneficios del desarrollo global de software son evidentes, las distancias que toman lugar en él traen consigo desafíos que, acorde a la propuesta de (Portillo-Rodríguez et al., 2012), conviene canalizar como parte de alguna de las siguientes áreas de problema:

- **PA1 - Dispersión geográfica:** Las ventajas de la interacción frente a frente, en su mayoría, se pierden al optar por una interacción remota.
- **PA2 - Desglose de coordinación y control:** La distribución geográfica hace que la dificultad de gestionar el control y la coordinación de actividades del proyecto se vea incrementada.
- **PA3 – Pérdida de comunicación:** Si se tiene en mente que la comunicación presencial constituye el canal de interacción más prolífico, está claro que el proceso de comunicación en los entornos de desarrollo global se ve comprometido por su propia naturaleza, algo que implica problemas relacionados con la colaboración y confianza entre los miembros del equipo.
- **PA4 – Pérdida del espíritu de equipo:** Los equipos crean un ambiente de trabajo apropiado cuando existe confianza entre sus miembros, y estos se conocen los unos a los otros. Las limitaciones en la comunicación, de mano de las posibles diferencias culturales, obstaculizan la creación de un espíritu de equipo saludable.
- **PA5 – Diferencias culturales:** Cuando individuos que proceden de distintas culturas trabajan en conjunto, cabe contemplar la posibilidad de que entre ellos se presenten malentendidos que afecten el desempeño del equipo negativamente.

2.2. Gestión de conocimiento

La gestión del conocimiento es una herramienta básica para la gestión empresarial. Es el proceso constante de buscar, identificar, clasificar, proyectar, presentar y usar de un modo más eficiente la experiencia y el conocimiento acumulados en una organización, de forma que mejore el alcance del empleado para conseguir ventajas competitivas. Su propósito principal es traducir el conocimiento en acción y traducir la acción en resultados. (Nieves-Lahaba y León-Santos, 2001)

Existen 4 motivos clave detrás del interés que despierta la gestión del conocimiento (Dalkir, 2001):

- **La globalización del negocio:** Actualmente las organizaciones procuran contar con más presencia global, buscando presentar competencia en múltiples localizaciones, en múltiples idiomas y orientando sus servicios a múltiples culturas.
- **La aparición de empresas más magras:** El ritmo de trabajo de las empresas ha ido acelerándose con los años, adquiriendo estas la capacidad de producir más en menor cantidad de tiempo, en relación a décadas anteriores.
- **Amnesia corporativa:** El conocimiento no es un elemento fijo, sino que por el contrario, se encuentra en flujo constante. Esto puede ocasionar problemas en su continuidad, especialmente cuando no se gestiona correctamente.
- **Avances tecnológicos:** La tecnología ha logrado que la conectividad se vuelva no sólo ubicua, sino que ha conseguido cambiar radicalmente las expectativas que se tenían hasta hace pocos años con respecto a lo que se concebía, entonces, sería su aplicación en un futuro inmediato. Es un hecho que los tiempos de respuesta se miden, hoy, en minutos, motivo por el que las empresas continúan buscando nuevas herramientas que les permitan estrechar, aún más, la conexión existente entre ellas.

A raíz de estos factores, se ha llegado a la conclusión de que las claves para el éxito dentro de un programa de gestión de conocimiento son las siguientes (Nieves-Lahaba y León-Santos, 2001):

- Las acciones para desarrollar la gestión del conocimiento deben corresponderse con la estrategia de la compañía.
- Realizar un estudio de las actividades de la empresa para concentrar los esfuerzos en las que generan ventajas competitivas.
- Transformar el conocimiento implícito en explícito.
- Asegurarse de que la recolección, almacenamiento y recuperación del conocimiento, producen un aumento de valor en la empresa.
- Asegurarse de que la herramienta escogida para el almacenamiento y recuperación es la más cercana a la ideal.
- Transmitir el conocimiento recolectado a toda la organización.
- La gerencia debe estar en disposición de apoyar las iniciativas que se desarrollen como resultado del proceso.
- Favorecer en los integrantes de la empresa la capacidad de aprendizaje del entorno, para lograr una mejor adaptación a los cambios.

2.2.1. Conocimiento tácito

El conocimiento tácito es la suma de conocimientos que forman parte de la experiencia propia de una persona, en la cual influyen rasgos de su identidad: creencias, valores, puntos de vista, intuición, etc. Al involucrar factores inherentes a la personalidad del individuo, es difícil lograr que el conocimiento tácito sea estructurado o externalizado objetivamente. (Uriarte, 2010)

Pese a ello, el conocimiento tácito es altamente valorado organizacionalmente, ya que genera ventajas competitivas al ser virtualmente imposible de imitar para los competidores. (Nonaka, 1995)

2.2.2. Conocimiento explícito y transición del conocimiento tácito

Contrario al conocimiento tácito, el conocimiento explícito sí puede ser clasificado, almacenado y distribuido. Puede encontrarse en diversas formas: fórmu-

las matemáticas, normas y estándares, especificaciones, procedimientos científicos, manuales y tutoriales, bases de datos y un largo etcétera. Por su naturaleza puede transmitirse con facilidad de un individuo a otro, e incluso ser procesado por un equipo informático, distribuido electrónicamente y almacenado en una base de datos.

Es gracias a la transición del conocimiento tácito hacia el conocimiento explícito que el aprendizaje organizacional es posible. Este proceso se da en cuatro etapas, identificadas como (Nonaka, 1995):

- **Etapa de socialización (conocimiento tácito a tácito):** Se inicia, por regla general, al crear un campo de interacción que permite a los individuos compartir sus experiencias y modelos mentales, transfiriendo de esta forma el conocimiento tácito de una persona a otra.
- **Etapa de exteriorización (conocimiento tácito a explícito):** Comienza a partir de un diálogo o reflexión colectiva que, apoyada en una metáfora o analogía apropiada, permite a los miembros enunciar el conocimiento tácito oculto.
- **Etapa de combinación/asociación (conocimiento explícito a explícito):** Comienza con la distribución, por medio de redes, del conocimiento recién creado y del conocimiento existente por parte de otras secciones o áreas de la organización, distinguiendo y esclareciendo, así, un nuevo conocimiento, producto o servicio.
- **Etapa de interiorización (conocimiento explícito a tácito):** En esta etapa se retroalimenta el proceso completo, con la intención de poner en práctica el nuevo conocimiento.

2.2.3. Aprendizaje organizacional

El aprendizaje organizacional se refiere a la capacidad de la organización de mejorar constantemente, de adaptarse a las transformaciones de la sociedad y a los factores vinculados a ella. (Senge et al., 2002)

Una organización capaz de aprender debe antes desarrollar cinco competencias para considerarse tal (Senge, 1995):

- **Desarrollar el dominio personal:** El individuo debe aprender a reconocer sus verdaderas capacidades y las de las personas que le rodean, pues solo tras identificar en un plano introspectivo quién es, qué desea y qué es capaz de hacer, podrá identificarse con la visión de la organización, proponer soluciones creativas y aceptar el compromiso de crecer y aprender junto con la organización.
- **Gestionar los modelos mentales:** La forma de pensar de un grupo de personas clave restringe, en ocasiones, la visión de quiénes están bajo su cargo. Los líderes deben conocer y manejar sus modelos mentales para promover comunicaciones claras y efectivas dentro de sus equipos y convertirse, así, en un medio de apoyo para su crecimiento.
- **Impulsar la visión compartida:** La clave para lograr una visión que se convierta en fuente de inspiración y productividad para la organización es que todos los miembros sean conscientes de sus visiones personales y que las concilien con la visión organizacional. Todas las visiones personales pueden alimentar la gran visión de la organización, y cada elemento del personal puede sentir en ella una conexión íntima que lo impulsa a convertirla en realidad.
- **Fomentar el aprendizaje en equipo:** Fortalecer equipos de trabajo es una premisa centrada en el diálogo, en el pensamiento colectivo como la fórmula para obtener mejores ideas, buscando un espíritu de cooperación sin anular al individuo.
- **Desarrollar el pensamiento en sistemas:** Se trata de pensar en las organizaciones y sus contextos como sistemas que coexisten y se afectan unos a otros, pues la realidad funciona en base a sistemas globales. Para ello es necesario comprender cómo funcionan los elementos del mundo exterior, prestando atención al conjunto y no a las piezas aisladas.

2.2.4. Estrategias de gestión de conocimiento en entornos de desarrollo global de software

El establecimiento de una comunidad de práctica es una estrategia en entornos de desarrollo distribuido que suele surgir naturalmente, cuando se reúne a dis-

tintos miembros con un interés en común dentro de un dominio en particular, o bien, puede formarse específicamente con la idea de lograr el intercambio de conocimiento. Es deseable que esta práctica se apoye en un portal web al cual sus integrantes puedan acceder para compartir información o discutir puntos de vista. (Dossani y Kenney, 2008)

Algunas estrategias de gestión de conocimiento más específicas se ven reflejadas en las tablas mostradas a continuación, las cuales describen procedimientos ideados con el fin de mitigar al máximo los posibles riesgos que pudieran surgir en el proceso de transferencia del conocimiento en entornos de desarrollo global de software, clasificando dichas estrategias en base a la naturaleza de los factores a los que corresponden y especificando exactamente el medio en el que incurren para lograr su objetivo, pudiendo tratarse de una revisión sistemática de literatura en materia de casos documentados de solución de incidencias en ambientes similares, o de técnicas de comunicación y entrevista con el personal pertinente, según convenga (Nidhra et al., 2012):

VI. Estrategias de gestión de conocimiento en base a factores de personal (Nidhra et al., 2012)	
Método: Revisión sistemática de literatura	
Área o proceso	Estrategia
Desarrollo de puentes culturales	Acatar culturas de alto colectivismo
	Realizar visitas in situ y sesiones de repetición
	Conducir talleres culturales
Dotación de personal	Estimular la motivación individual
	Implementar programas de <i>mentoring</i> y <i>shadowing</i>
	Designar remitentes de conocimiento creíbles
Fomento a la confianza	Reforzar comunicación con el cliente
	Promover relaciones de alta calidad con el cliente
	Comprender lenguaje y cultura de negocios del cliente
	Viajar al lugar dónde radica el cliente con el fin de fortalecer la relación con el mismo
Atributos personales	Capacitar al personal de TI para incrementar su capacidad
	Aprender por experiencia
	Mejorar las habilidades interpersonales y de gestión de tecnología
	Fomentar el aprendizaje proactivo y los canales de ayuda punto a punto

VII. Estrategias de gestión de conocimiento en base a factores de personal (Nidhra et al., 2012)	
Método: Entrevista	
Área o proceso	Estrategia
Puentes de lenguaje	Emplear la comunicación verbal con tanta frecuencia como sea posible
Dotación de personal	Transferir conocimiento según la experiencia del destinatario
	Asegurar la transferencia del conocimiento en todo momento, aun cuando se presentan retos
	Grabar las sesiones de transferencia de conocimiento en las que se involucren miembros novatos
	Supervisar los procesos de shadowing a través del uso de escritorios compartidos y llamadas telefónicas
Fomento a la confianza	Ubicación de empleados subcontratados in situ
	Comunicación continua
	Interacción frente a frente
Atributos personales	Repetir proceso de transferencia de conocimiento hasta que el receptor asimila el conocimiento al 100 %
	Destinar esfuerzos a la comprensión de requerimientos

VIII. Estrategias de gestión de conocimiento en base a factores de proyecto (Nidhra et al., 2012)	
Método: Revisión sistemática de literatura	
Área o proceso	Estrategia
Comunidad de práctica	Obtener conocimiento a través de discusiones grupales
Directrices de proyecto	Tomar en cuenta a los analistas de negocio subcontratados en la ingeniería de requerimientos
	Asignar términos de negocio a entidades lógicas
	Mantener legibilidad en el proceso de documentación
	Delegar la especificación de los test de aceptación al equipo de desarrollo subcontratado
	Agendar juntas semanales adicionales para cerrar posibles brechas de conocimiento entre los integrantes
	Destinar recursos a la transferencia de conocimiento
Verificación de proceso de transferencia del conocimiento	Aplicar exámenes orales y escritos
	Realizar presentaciones invertidas para la validación de requerimientos
	Apoyar la incentivación
	Realizar sesiones de repetición

VIII. Estrategias de gestión de conocimiento en base a factores de proyecto (Nidhra et al., 2012)	
Método: Revisión sistemática de literatura	
Área o proceso	Estrategia
Facilitadores clave	Comprender los subprocesos del aprendizaje organizacional
	Aprovechar la base de conocimiento y experiencia de todos los puntos (miembros del equipo)
	Establecer ayudas de navegación dinámicas para la búsqueda de información
	Modularización
	Utilizar fuentes de <i>expertise</i> externo
	Colaboración conjunta
	Identificación de personalidades dentro del trabajo
Puentes de comunicación	Impartir capacitación en salón de clases a novatos
	Estar pendiente de los hitos, las notificaciones de status y las reuniones de trabajo del proyecto
	Formalizar estructuras de comunicación
	Llevar a cabo juntas informales y presenciales entre cliente y vendedor
	Promover la comunicación a gran escala

IX. Estrategias de gestión de conocimiento en base a factores de proyecto (Nidhra et al., 2012)	
Método: Entrevista	
Área o proceso	Estrategia
Distancia temporal	Usar herramientas de comunicación multimedia
	Adherirse a la agenda de transferencia de conocimiento
Comunidad de práctica	Organizar sesiones grupales para solucionar un problema
	Organizar sesiones de grupo al iniciar la jornada de trabajo
	Designar a un miembro experimentado de otro equipo como conductor de las sesiones de grupo
Directrices de proyecto	Documentar procesos
	Leer documentación antes de iniciar la transferencia del conocimiento
	Proporcionar una visión conjunta del documento antes de iniciar la transferencia del conocimiento

IX. Estrategias de gestión de conocimiento en base a factores de proyecto (Nidhra et al., 2012)	
Método: Entrevista	
Área o proceso	Estrategia
Gestión de tareas	Asignar una cantidad de tiempo suficiente a la comprensión de requerimientos
	Construir conocimiento del equipo a fin de reducir los puntos de fallo individual
	Asignar tiempo al empleado del siguiente turno
Deadlines de proyecto	Planificar correctamente la transferencia del conocimiento
	Efectuar stand-ups y reuniones de fin de jornada
	Asignar horas extras a la fase de diseño
	Involucrar en la fase de diseño a tantos miembros como sea posible
	Inmiscuir al analista de negocios externos en la localidad del cliente
Verificación	Impartir cursos de e-learning diseñados para evaluar el conocimiento actual
	Aplicar exámenes orales semanalmente
	Programar actualizaciones regulares a través de correo electrónico y reuniones a través de pantalla compartida
	Llevar a cabo un proceso de marcado para asegurar la entrega de la transferencia de conocimiento
Reducción de costos extra	Transferir conocimiento a través de medios de comunicación, en vez de a través de visitas presenciales
	Asegurar la entrega de conocimiento del anterior proveedor al nuevo
	Obtener acceso rápido y sencillo al repositorio de conocimiento del cliente
	Reclutar personal con base en la experiencia pasada dentro de dominios similares
Innovación	Enviar empleados a sesiones de capacitación en instituciones autorizadas
	Programar clases impartidas por expertos del dominio
	Organizar tormentas de ideas
	Emplear guías de usuario y documentación para la solución de nuevos problemas

IX. Estrategias de gestión de conocimiento en base a factores de proyecto (Nidhra et al., 2012)	
Método: Entrevista	
Área o proceso	Estrategia
Rotación de empleados	Mantener al menos a un 30 % de los empleados en cada capa del conocimiento
	Buscar que el empleado actual transfiera conocimiento al empleado que va a reemplazarlo
	Preparar el plan de transferencia de conocimiento antes de que el empleado abandone el cargo
Puentes de comunicación	Establecer canales para la comunicación directa del equipo subcontratado con el equipo local

X. Estrategias de gestión de conocimiento en base a factores tecnológicos (Nidhra et al., 2012)	
Método: Revisión sistemática de literatura	
Área o proceso	Estrategia
Uso y acceso del groupware	Utilizar un gestor de documentos y configuraciones
	Utilizar técnicas de escritorio compartido para los aspectos menos complejos del negocio
	Emplear correo, wikis y mensajería instantánea para transmitir información técnica en forma detallada
	Dar preferencia al correo electrónico como herramienta de comunicación asíncrona
	Utilizar servicios de mensajería instantánea como sustituto de la comunicación verbal e informal
	Gestionar cursos, programas de tutoría, minería y perfilado de conocimiento a través de la web
	Obtener acceso a un repositorio de conocimiento

XI. Estrategias de gestión de conocimiento en base a factores tecnológicos (Nidhra et al., 2012)	
Método: Entrevista	
Área o proceso	Estrategia
Soporte a herramientas de software	Llevar a cabo video conferencias y juntas en vivo
	Usar correo electrónico para comunicación asíncrona
Mantenimiento de un TMS	Mantener un repositorio junto con un sistema de gestión de cambios

2.3. Trabajos relacionados

Esta sección da a conocer, de entrada, algunas de las publicaciones y trabajos de tesis relacionados con el sistema actual.

De manera adicional, se incluyen las descripciones generales de los tipos de herramientas dentro de los que la aplicación desarrollada podría clasificarse, pudiendo ser éstos los sistemas colaborativos, los sistemas de gestión de conocimiento o los sistemas de localización de expertos.

Por último, se hace mención a algunos de los sistemas existentes cuya funcionalidad y/u objetivos se asemejan a los de la propuesta.

2.3.1. Publicaciones relacionadas

- *Predicting groupware usage* (Yoo, 1998): Publicación realizada hace cerca de dos décadas. Proyecta las implicaciones del uso del groupware hacia un futuro, tanteando su impacto en los procesos sociales dentro de las empresas, dando al groupware la denominación de tecnología social.
- *Knowledge management and knowledge management systems: Conceptual foundations and research issues* (Alavi y Leidner, 2001): Artículo que da a conocer los fundamentos de los KMS, haciendo mención a los cuatro procesos básicos del conocimiento (creación, almacenamiento, transferencia y aplicación) y brindando ejemplos específicos de herramientas de TI que, como sistemas de gestión de conocimiento, dan soporte a estos procesos.
- *Collaborative software development: A discussion of problem solving models and groupware technologies* (DeFranco-Tommarello y Deek, 2002): Trabajo que sienta las bases para el desarrollo de un modelo de solución de problemas mediante software colaborativo, considerando las dinámicas comunes del trabajo en equipo.
- *A belief network model for expert search* (Macdonald y Ounis, 2006): Este documento propone un modelo de creencias, basado en redes bayesianas, para mejorar el orden de los resultados obtenidos de los sistemas de

búsqueda de expertos, otorgándoles una prioridad acorde a la consulta realizada.

- *A two-stage model for expert search* (Cao et al., 2006): Propuesta de un modelo de dos etapas para la localización de personal experto, consistiendo la primera en una búsqueda de perfiles evidenciados mediante los documentos asociados a los expertos, y la segunda en una revisión de coincidencias (citas) entre expertos y los términos que estos emplean.
- *Can technology build organizational social capital? The case of a global IT consulting firm* (Sherif et al., 2006): Estudio que evidencia el rol de los sistemas de gestión de conocimiento en la acumulación del capital social dentro de las empresas que hacen uso de ellos, y su efecto en la creación y transferencia de conocimiento.
- *Conceptualization of the antecedents and impacts of KMS utilization: A preliminary framework* (Oyefolahan y Dominic, 2010): En este artículo se exploran los antecedentes de los KMS, explicando su contribución para con las prácticas de transferencia de conocimiento, y estableciendo un marco de trabajo conceptual como fundamento teórico.
- *Groupware y software social: propuesta de marco de evaluación analítica para herramientas de software libre* (Tramullas et al., 2011): Documento que define un modelo de análisis de funcionalidad de uso del groupware, para después utilizarlo para determinar la utilidad y verificar la integración de nuevas prestaciones sociales dentro de las aplicaciones de trabajo en grupo más populares.
- *Construcción de un modelo para el diseño de perfiles de conocimiento* (Velázquez-Mendoza, 2013): Tesis que aborda intensivamente el concepto del perfil de conocimiento, contribuyendo a la creación de una ontología de conocimiento en desarrollo de software, y por tanto, a la generación de un modelo para la generación de perfiles de conocimiento orientados a la gestión de recursos humanos dentro de las empresas.
- *Studying social interactions in groupware systems* (Montané Jiménez et al., 2015): Reporte de estudio aplicado a 30 usuarios de sistemas colaborativos, en la búsqueda de posibles mejoras ajustables a los mecanismos empleados en dichos sistemas.

2.3.2. Tecnologías relacionadas

2.3.2.1. Sistemas colaborativos o groupware

Los sistemas colaborativos, también llamados groupware, son aplicaciones o conjuntos de herramientas diseñadas para cubrir las necesidades de dos o más personas, de forma que puedan cooperar y coordinar sus actividades dentro de un entorno en común con objetivos compartidos. (Montané Jiménez et al., 2015)

Un groupware es un tipo especializado de software social que constituye un espacio de trabajo en colaboración donde se presta soporte a tres tipos de actividades principales (Tramullas et al., 2011):

- **Comunicación e intercambio de información:** Entre los usuarios de un groupware puede haber publicación e intercambio de contenido, generalmente a modo de mensajes o documentos.
- **Mensajería y conferencia:** Sistemas de interacción directa, sincrónica o asincrónica, entre usuarios. Por ejemplo: foros de discusión, conferencias, sistemas de mensajería instantánea, etc.
- **Coordinación:** Actividades cuya finalidad es coordinar las acciones de los miembros del grupo sobre documentos, revisiones, reuniones, etc. En esta categoría se distinguen los calendarios y agendas compartidas.

Cuando los miembros de un equipo de desarrolladores se encuentran geográficamente separados, el éxito de su comunicación puede depender de su uso efectivo de los sistemas colaborativos, los cuales comúnmente integran utilidades como clientes de correo electrónico, calendarios compartidos, agendas de grupo, herramientas auxiliares en gestión de información e incluso acceso a la web en una misma herramienta. (DeFranco-Tommarello y Deek, 2002)

En la actualidad el uso del groupware, en su calidad de software diseñado específicamente para facilitar el trabajo en equipo y el compartimiento de información, está siendo ampliamente propuesto por expertos en sistemas informáticos como la herramienta de TI ideal para las nuevas estructuras organizacionales, pues involucra interacciones interpersonales y procesos sociales. (Yoo, 1998)

2.3.2.2. Sistemas de gestión de conocimiento o KMS

En consonancia con el interés que han proyectado las empresas hacia el aprendizaje organizacional y la gestión del conocimiento, numerosos investigadores y especialistas en sistemas informáticos se han encargado de dar promoción a una clase de herramientas a las que ellos llaman “sistemas de gestión de conocimiento” (KMS por su acrónimo en inglés), cuyo objetivo es apoyar la creación, transferencia y aplicación del conocimiento dentro de la organización. (Alavi y Leidner, 2001)

Los tres usos comunes de los sistemas de gestión de conocimiento en el ámbito de las tecnologías de información son (Sherif et al., 2006):

- El almacenamiento de las lecciones aprendidas a nivel individual y organizacional.
- El acceso al conocimiento y a las experiencias del personal de la organización, a través de la interacción entre usuarios.
- La creación de redes de conocimiento.

Un claro ejemplo de KMS son las Intranets, redes privadas empleadas en miles de empresas para administrar el conocimiento interno, cuyo uso ha contribuido a mantener una comunicación activa y aumentar los esfuerzos colaborativos entre los miembros del personal de las empresas que se sirven de ellas, fortaleciendo la confianza y la capacidad de mutuo acuerdo entre dichos miembros.

Por tanto, de lo anteriormente expuesto, se entiende que el uso efectivo de los sistemas de gestión de conocimiento puede contribuir drásticamente al desarrollo de las técnicas de intercambio de conocimiento de la organización, así como a su desarrollo social y de capital humano, mejorando, consecuentemente, su desempeño general. (Oyefolahan y Dominic, 2010)

2.3.2.3. Sistemas de localización de expertos

Localizar las competencias y conocimientos de los miembros de una comunidad es un problema vital para mejorar la gestión del conocimiento, llegando a

considerarse un factor crítico a causa del reciente aumento de grupos de trabajo distribuidos.

Algunos sistemas de localización de expertos, también llamados sistemas de búsqueda de expertos, emplean como concepto clave las redes de conocimiento, las cuáles son grafos que relacionan a las personas que comparten conocimientos dentro de un grupo. En dicho contexto, las personas constituyen los nodos del grafo, y sus conocimientos se representan por medio de aristas. (Pujol y Sangüesa, 2001)

Otra perspectiva empleada por ciertos sistemas de localización de expertos para descubrir talento implica construir y mantener, en forma manual, una base de datos donde el *expertise* de los usuarios es almacenado. (Cao et al., 2006)

En cambio, los sistemas de búsqueda de expertos modernos para entornos empresariales emplean mecanismos para extraer información mediante el análisis de los documentos que conforman el perfil de cada candidato. De estos documentos se compone la evidencia del conocimiento de los expertos, la cual puede tomar múltiples formas: curriculums en distintos formatos, listas de sitios web visitados por el experto, e incluso la información que éste ha brindado en dichos sitios podría ser empleada para perfilar su conocimiento. (Macdonald y Ounis, 2006)

Indistintamente de su enfoque, existen tres parámetros determinantes en la valoración de estas herramientas (Macdonald y Ounis, 2008):

- Las técnicas o modelos aplicados como criterio de evaluación del conocimiento de los expertos, las cuales impactan en la selección de candidatos.
- La forma en la que el conocimiento es relacionado con cada experto y viceversa, la cual impacta en la precisión del procedimiento.
- La calidad del conocimiento evidenciado por cada candidato, la cual impacta en la fiabilidad proyectada hacia el usuario.

Aunque los sistemas de localización de expertos son aún herramientas demasiado recientes y su uso no se extiende a un gran número de organizaciones a gran escala, trabajos como el de (Palacio et al., 2011) han logrado esclarecer un ciclo de vida para el flujo de interacción que se da entre las personas que,

estando en entornos distribuidos, precisan localizar colaboradores con la finalidad de solicitar su apoyo en la solución de problemas, coincidiendo así con el flujo de interacción presente en estos sistemas:

- Durante la primera etapa se identifica la información requerida por el usuario en torno a la actividad en la cual se encuentra involucrado al momento de la consulta.
- Durante la segunda etapa se busca un momento apropiado para solicitar el apoyo de los colaboradores.
- Durante la tercera y última etapa, se da inicio a la interacción necesaria para establecer una solución para el problema que se presenta. Esto requiere que el usuario pueda monitorear cierta información del colaborador para determinar si se encuentra o no disponible en el momento que se le precisa.

2.3.3. Herramientas relacionadas

La solución delimitada comparte uno o varios rasgos de funcionalidad con los siguientes proyectos:

- *NETEXPERT: Sistema basado en inteligencia artificial para la localización de expertos* (Pujol y Sangüesa, 2001): Descripción de una herramienta que permite buscar por diferentes vías, dentro de una comunidad, qué miembro posee un conocimiento dado.
- *Expert search evaluation by supporting documents* (Macdonald y Ounis, 2008): Este trabajo propone una evaluación de los documentos de soporte (currículum, publicaciones, etc.), empleándolos como un medio para la valoración del *expertise* del candidato evaluado y su relevancia para con la consulta realizada.
- *Providing support for starting collaboration in distributed software development: A multi-agent approach* (Palacio et al., 2009): Planteamiento de un modelo de apoyo a trabajadores que laboran en entornos de desarrollo distribuido, sugiriendo un escenario de implementación de un prototipo basado en un enfoque multi-agentes.

- *What information would you like to know about your co-worker? A case study* (Aranda et al., 2010): Trabajo que extrae los requerimientos para el desarrollo de una herramienta que facilite la comunicación y coordinación en entornos de desarrollo global, al analizar las necesidades de los programadores que se desenvuelven en dichos ambientes, en relación con cuáles de los datos personales y profesionales de sus compañeros de trabajo, les parece más importante conocer.
- *Tool to facilitate appropriate interaction in global software development* (Palacio et al., 2011): Publicación que acuña el término de esferas de trabajo colaborativo en referencia a un grupo de personas que organizan y definen su trabajo individualmente, identificando de paso las áreas dónde su trabajo incide con el de sus compañeros, logrando así un procedimiento de trabajo en conjunto. Este artículo detalla también la elaboración de una herramienta de mensajería instantánea para mejorar la productividad en dichos ambientes.
- *Construcción de un modelo de lógica difusa para validación de perfiles de conocimiento de personal* (Rosas-Daniel, 2014): Trabajo de tesis que describe el diseño y construcción de un modelo de lógica difusa que permite empatar a los individuos, en base a sus capacidades, con los roles que éstos pueden desempeñar satisfactoriamente dentro de una empresa. Constituye el principal cimiento para el desarrollo del sistema descrito en el presente documento, pues el prototipo que demuestra la aplicación del modelo se integra a este proyecto como un servicio de valoración de perfiles de conocimiento.

2.4. Uso de plataformas móviles

Son plataformas móviles aquellos sistemas operativos que tienen por finalidad ser desplegados en dispositivos portátiles, tales como tablets o teléfonos inteligentes, permitiendo a sus usuarios la instalación y uso de aplicaciones móviles en dichos dispositivos. Como ejemplos notables de estas plataformas se distinguen: iOS, Android, BlackBerry OS, Symbian, webOS o Windows Phone. (Rouse y Steele, 2011)

2.4.1. Principales plataformas móviles según su cuota de mercado

Una de las primeras cuestiones que deben abordar los desarrolladores de aplicaciones móviles es la selección de la plataforma final en la que llevarán a cabo el despliegue de sus aplicaciones.

En términos de mercado, acorde a la base total de usuarios de dispositivos móviles en el mundo, las principales plataformas móviles actualmente son:

2.4.1.1. Android

Este sistema operativo es administrado por Google. Está basado en el kernel de Linux y funciona principalmente en dispositivos móviles con pantalla táctil, como tablets y smartphones, aunque adicionalmente Google ha creado versiones de Android para otros dispositivos, como el caso de Android TV para televisiones, Android Auto para automóviles y Android Wear para relojes inteligentes, cada uno con una interfaz dedicada. Existen también algunas otras variantes para ordenadores portátiles, consolas de videojuegos, cámaras digitales, entre otros dispositivos electrónicos.

Acorde a las cuotas del mercado móvil, Android es la plataforma con mayor aceptación comercial a día de hoy por un margen bastante amplio (Richter, 2017). Se prevé que incluso, en un futuro próximo, podría llegar a desplazar a Microsoft Windows como sistema operativo más utilizado en general. (Popa, 2017).

Inicialmente creado por Android, Inc., empresa absorbida por Google en el año 2005, la plataforma Android fue revelada al público en el año 2007, al mismo tiempo que se estableció el Open Handset Alliance, un consorcio de compañías de hardware, software y telecomunicaciones, dedicado al desarrollo de estándares abiertos para dispositivos móviles. (Open Handset Alliance, 2007)

Hacia Julio de 2013, el servicio de descarga de aplicaciones de la plataforma, de nombre Google Play, contaba con más de un millón de aplicaciones disponibles y más de 50 mil millones de descargas.

Una encuesta aplicada durante el mismo año a desarrolladores de aplicaciones móviles concluyó que el 71 % desarrollaba aplicaciones para Android, y más tarde, en 2015, los resultados arrojados por una encuesta adicional indicaban que el 40 % de los desarrolladores profesionales en general consideraban a Android su plataforma de desarrollo prioritaria. (Constantinou, 2015)

2.4.1.2. iOS

Llamado originalmente iPhone OS tras su lanzamiento en septiembre del año 2007, y renombrado en junio del año 2010, iOS es un sistema operativo creado y desarrollado por la empresa Apple Inc. en forma exclusiva para sus dispositivos móviles iPhone, iPod Touch, iPad, Apple Watch y AppleTV.

Hacia el año 2011, iOS acaparaba el 60 % del mercado de los dispositivos móviles. Sin embargo, a fines de 2014, sólo el 14.8 % de los usuarios de smartphones y el 27.6 % de los usuarios de tablets a nivel mundial eran usuarios de la plataforma. (Llamas et al., 2015)

Por tanto, iOS es actualmente, por detrás de Android, la segunda plataforma móvil más popular en el mercado, habiendo alcanzado la cifra de 1.4 millones de aplicaciones desarrolladas para su despliegue en distintos gadgets diseñados por Apple. (Ranger, 2015)

2.4.1.3. Windows Phone

El desarrollo de esta plataforma comenzó en febrero del año 2010, como una intención por parte de Microsoft Corporation de renovar completamente su anterior sistema operativo para dispositivos móviles, Windows Mobile, mismo que a diferencia de Windows Phone, se enfocaba mayoritariamente en el mercado empresarial, y no en el de los consumidores casuales.

Windows Phone, de la misma forma que Android e iOS, cuenta con acceso a un servicio de descarga a través de la nube con más de 300,000 aplicaciones disponibles. (Mullally et al., 2011)

Diversos estudios recientes reportan que el uso de Windows Phone como plataforma, tras haber alcanzado su cénit en 2013, logrando colocar en el mer-

cado un parque de dispositivos correspondiente al 3.4 % del total de móviles a nivel mundial, ha disminuido en tal forma que dicha base de usuarios se redujo hasta 2.5 % durante el año 2014, y posteriormente hasta 1.7 % al finalizar el año 2015. (Chokkattu, 2015)

2.4.2. Desarrollo de aplicaciones móviles

Numerosos son los fundamentos que convergen en el atractivo comercial del mercado de las aplicaciones móviles, cuyas bazas principales son (Tracy, 2012):

- Acceso a redes móviles de datos que permiten a los usuarios descargar contenido o actualizar sus aplicaciones desde casi cualquier localización geográfica, dando lugar a una nueva generación de aplicaciones cuya funcionalidad se basa en la conectividad constante a Internet a través del consumo de datos móviles.
- Dispositivos móviles de alto rendimiento a precios relativamente económicos, siendo incluso asequibles para casi cualquier consumidor objetivo.
- Tiendas de aplicaciones integradas en las plataformas, de uso sencillo, y en las que pueden encontrarse cientos de miles de aplicaciones.
- Soporte a aplicaciones de terceros y no solamente a aplicaciones desarrolladas por el fabricante del dispositivo, como sucedía antaño durante la primera generación de teléfonos inteligentes.
- Necesidad constante de mantener la simplicidad en el diseño como estándar, siguiendo el principio KISS (*Keep It Simple, Stupid*).

2.4.2.1. Soluciones nativas

El código nativo es aquel que se programa en forma exclusiva para una plataforma o procesador en concreto, empleando un set de instrucciones de código diseñadas para el mismo, de manera tal que para lograr ejecutarlo en otras plataformas, el código ha de ser interpretado por algún emulador que brinde al dispositivo las instrucciones necesarias para replicar las características del procesador para el cual el código ha sido programado.

Las soluciones nativas usualmente emplean código escrito en algún lenguaje compilado, implicando que la aplicación ha de ser compilada en tiempo de ejecución. Los lenguajes compilados suelen proveer soluciones más rápidas que su contraparte, los lenguajes interpretados, debido a la generación de archivos ejecutables. (Charland y Leroux, 2011)

Los propietarios de los sistemas operativos se muestran interesados en lograr que los desarrolladores programen aplicaciones nativamente para sus plataformas, de forma que puedan sacar ventaja de sus prestaciones particulares, lo cual los lleva a desarrollar aplicaciones usando el lenguaje correspondiente a la plataforma del proveedor, haciendo que versiones de la misma aplicación para otras plataformas constituyan proyectos independientes hechos con código diferente al original. (Serrano et al., 2013)

2.4.2.2. Soluciones de web móvil

El término “web móvil” se emplea para describir dos tipos de aplicaciones web, ya sea algún sitio web tradicional diseñado para navegadores móviles, o una aplicación diseñada con tecnologías web, la cual incluye elementos renderizados a modo de aplicación móvil nativa, y suele recibir el nombre de “aplicación móvil híbrida”. (Serrano et al., 2013)

Desde la perspectiva del desarrollador sus ventajas son claras, pues al ser desarrolladas por medio de tecnologías web, son más accesibles para programadores inexpertos, manejan menor cantidad de código, son más fáciles de probar y depurar, además de que una misma versión puede visualizarse desde distintas plataformas con tan sólo escasos ajustes.

No obstante, tanto de parte del desarrollador, como de parte del usuario final, las aplicaciones de web móvil aún cuentan con grandes áreas de oportunidad, pues su usabilidad es ajustada debido a las limitaciones presentes en el diseño de interfaces gráficas para móviles, dado que no se emplean los elementos nativos de las plataformas para las cuáles son desarrolladas.

Por otra parte, la capa del navegador agrega un nivel de procesamiento extra que supone un desempeño más lento que el de sus contrapartes nativas, limitando, también, el uso de los sensores y prestaciones exclusivas de los

dispositivos móviles actuales, como el acelerómetro, el giroscopio, el sensor de proximidad, entre otros. (Sin et al., 2012)

2.4.2.3. Soluciones híbridas

Cómo se mencionó anteriormente, las soluciones híbridas son aquellas soluciones que conjugan elementos de soluciones de web móvil con elementos de soluciones nativas, resultando más robustas frente a las primeras, y más sencillas y asequibles, para el desarrollador, frente a las segundas.

El enfoque híbrido surge como respuesta al problema de acceso al hardware que presentan las aplicaciones web móviles, mezclando la portabilidad de éstas con el acceso a las funciones particulares del dispositivo visto en aplicaciones nativas, dado que las aplicaciones híbridas son instaladas dentro del dispositivo, pero su codificación se lleva a cabo utilizando tecnologías web. (Pretel y Lago, 2014)

2.4.2.4. Android como plataforma de desarrollo móvil

Es el más libre de los sistemas operativos comerciales para dispositivos móviles. A pesar de que su desarrollo nativo se lleva a cabo mediante Java, las aplicaciones resultantes no emplean la máquina virtual de Java como intérprete, sino un par de soluciones propias de Google denominadas Dalvik (para versiones anteriores a la versión 5.0) y ART (versión 5.0 en adelante).

Esta plataforma constituye la opción más accesible de cara a los desarrolladores para la creación de aplicaciones móviles, pues requiere un solo pago de \$25 USD por una licencia de desarrollador completa, sin ningún tipo de restricción, permitiendo publicar un número ilimitado de aplicaciones desde una misma cuenta. (Masner et al., 2015)

2.4.2.5. iOS como plataforma de desarrollo móvil

Xcode es el entorno de desarrollo impuesto por Apple para su plataforma móvil, cuyas aplicaciones pueden desarrollarse utilizando Objective C o Swift

cómo lenguajes nativos. Estas aplicaciones deben ser diseñadas estrictamente en dispositivos propietarios de Apple (Macbooks, iMacs, etc.) puesto que de otro modo no será posible asegurar que su funcionamiento final tras su despliegue sea el esperado.

Aunque las herramientas de desarrollo para iOS son gratuitas, la licencia de desarrollo que permite publicar aplicaciones para esta plataforma tiene un costo de \$99 USD anuales. (Masner et al., 2015)

2.4.2.6. Windows Phone como plataforma de desarrollo móvil

El despliegue de aplicaciones para esta plataforma se lleva a cabo bajo el framework de .NET empleando Microsoft Visual Studio como entorno de desarrollo, motivo por el que sus aplicaciones pueden ser escritas nativamente tanto en C# como en Visual Basic.

Desde el punto de vista de costos, es necesario que los grupos de desarrollo paguen una única suma de \$19 USD para poder registrarse con una cuenta individual y publicar sus aplicaciones, o de \$99 USD para registrarse con una cuenta empresarial, siendo esta última una licencia más completa que la licencia básica, la cual restringe el uso de ciertas funcionalidades, y permite publicar un número limitado de aplicaciones. (Masner et al., 2015)

2.4.3. Tendencias del desarrollo móvil con impacto en entornos distribuidos

El trabajo colaborativo requiere de herramientas de comunicación como medio de apoyo a las actividades de coordinación y colaboración llevadas a cabo por los trabajadores que se desenvuelven en ambientes distribuidos.

Acorde a la forma en la que los trabajadores interactúan, el trabajo colaborativo puede clasificarse en tres niveles de interacción (Covili et al., 2009):

- **Síncrona:** Es únicamente posible cuando la colaboración se da en forma presencial y continua.

- **Asíncrona:** Se da entre trabajadores que interactúan a distintos tiempos, utilizando un componente centralizado como un punto de sincronización.
- **Multisíncrona:** En este contexto las tareas son divididas en fases autónomas, y una sincronización subsecuente de la información de todos los trabajadores es requerida para obtener una vista consistente del trabajo realizado.

Las tendencias sugeridas en el presente apartado podrían ser tomadas en cuenta cuando se requiere realizar trabajo en forma asíncrona o multisíncrona, tanto en entornos distribuidos como en entornos contiguos.

2.4.3.1. Mobile 2.0

Mobile 2.0 es el nombre que se le da a una bifurcación del enfoque Web 2.0, la cuál puede ser vista como su sucesor, pues extiende las capacidades de Web 2.0 hacia los dispositivos móviles, dónde sus limitaciones se afrontan con capacidades nunca antes vistas, como por ejemplo, el cómputo de datos de usuarios basado en su localización geográfica. (Griswold, 2007)

Algunas de las características de este enfoque son (Pardo-Kuklinski et al., 2010):

- Su ubicuidad, la cual permite captar el punto de inspiración del usuario, puesto que los dispositivos móviles han ido pasando de ser herramientas de consumo primario de información a ser una herramienta de producción de contenidos.
- La sinergia entre aplicaciones a través de mashups, al estilo de Google Maps y su uso extendido por varias aplicaciones.
- La conducción de servicios basados en la ubicación y movilidad del usuario, dado que ambas son cualidades esenciales y distintivas de los dispositivos móviles con la que los dispositivos de escritorio no pueden competir.
- El planteamiento de una búsqueda móvil en forma distinta a la que se utiliza desde los ordenadores, con énfasis en el momento, evento y lugar.

2.4.3.2. Soluciones web progresivas

Las aplicaciones web progresivas son aplicaciones web móviles cuya experiencia se presenta como la de una aplicación programada nativamente, prescindiendo de cualquier tipo de instalación en el dispositivo, debido a que el acceso a ellas se efectúa desde un navegador web. Algunas de sus cualidades más destacables son (Google Developers, 2016):

- Tal como su nombre lo indica, son aplicaciones progresivas debido a que están diseñadas de forma que a mayor flujo de datos de los usuarios hacia ellas, es decir, a mayor grado de uso, su desempeño y experiencia mejoran.
- Son responsivas: se adaptan al tamaño del dispositivo final en el que se implementan.
- Son independientes de conectividad. Esto quiere decir que aún si no se cuenta con ninguna conexión a Internet, basta con haber accedido al menos una vez previamente para poder continuar utilizándolas localmente.
- Son fáciles de compartir vía URL.
- Pueden ser instaladas en el dispositivo, si se decide no hacer uso de navegadores web para visualizarlas

2.5. Conclusiones del estado del arte

Durante el presente capítulo se expusieron los conceptos clave que describen la investigación realizada como parte del trabajo de tesis que respalda la propuesta, conceptos entre los que figuran el desarrollo global de software, la gestión del conocimiento y el uso de plataformas móviles.

De lo anterior podemos concluir que el desarrollo global de software es aquel paradigma de desarrollo que implica la realización de un trabajo de desarrollo entre distintos equipos procedentes de distintas partes del mundo, por lo que, por naturaleza, se da en entornos distribuidos.

De entre las dos principales vertientes que fueron distinguidas en lo que a tipos de entornos distribuidos respecta, la aplicación propuesta encaja fi-

losóficamente con los entornos de offshoring, en los que una empresa divide su función entre varios equipos sin recurrir a la mano de obra de terceros.

Posteriormente se analizó el término de gestión del conocimiento, un proceso que busca identificar, clasificar, proyectar, presentar y usar de un modo más eficiente la experiencia y el conocimiento acumulados dentro de una esfera de colaboración. Dentro de este apartado se analizaron algunas estrategias de gestión de conocimiento puestas en práctica en entornos de desarrollo global, con objeto de conocer algunos de los protocolos actuales empleados en dichos entornos para la solución de problemas relacionados con la colaboración y comunicación de sus participantes.

Más adelante se ahondó en el aspecto de las plataformas móviles disponibles, puesto que el despliegue de la aplicación, tal cual se encuentra estipulado en el proyecto, debe poder llevarse a cabo en más del 50 % de los smartphones comerciales, por lo que se determinó que la plataforma destino será, efectivamente, tal cual estaba contemplado en un inicio, Google Android, al contar con una base de usuarios que representa alrededor del 71 % de la cuota total del mercado de dispositivos móviles en la actualidad. (GlobalStats, 2016)

Otro tópico importante abordado en la sección de uso de plataformas móviles, concerniente al desarrollo de aplicaciones para las mismas, fue la comparación entre las soluciones nativas, las soluciones de web móvil y las soluciones híbridas, de entre las cuales, para el cumplimiento de los objetivos planteados en el anterior capítulo, destaca la primera por ofrecer un mejor rendimiento al brindar acceso a todas las prestaciones tecnológicas de los dispositivos móviles.

Una vez concluida la búsqueda de conceptos de investigación del estado del arte, se hizo una revisión de trabajos relacionados en la que se enlistaron sistemas, modelos y trabajos científicos cuyo eje central radica, ya sea en el proceso de interacción entre individuos en entornos distribuidos, o bien, en la búsqueda de expertos, dentro y fuera de dichos entornos.

Finalmente, se describieron los tipos de software actuales que mantienen objetivos similares a los de la propuesta, distinguiendo entre ellos los sistemas colaborativos o groupware, los sistemas de gestión de conocimiento y los sistemas de localización de expertos. Cabe recalcar que, aunque se profundizó poco más en el funcionamiento de éstos últimos por ser sistemas novedosos

que mantienen una mayor semejanza con el sistema propuesto respecto a los otros dos tipos de herramientas investigados, el enfoque del sistema descrito en el presente trabajo de tesis se relaciona con todos los tres arquetipos mencionados, mas no encaja íntegramente en ninguno de ellos.

Es por tanto que la propuesta puede, en sí, considerarse un nuevo tipo de herramienta de software: un sistema colaborativo basado en la búsqueda de perfiles de conocimiento de expertos.

Capítulo 3

Análisis y diseño del sistema

Este capítulo subraya las características de los elementos de la aplicación móvil de apoyo en la identificación de colaboradores en entornos de desarrollo global de software, y de sus componentes externos, en lo concerniente a su contexto de negocio, visión, alcance, requerimientos, contexto de sistema y diseño de su arquitectura. Por ende, con el fin de explicar a fondo dichos aspectos, los procedimientos del sistema se proyectan desde diversas perspectivas:

- **Modelado de procesos de negocio:** establece el flujo existente entre los procesos operativos fundamentales.
- **Diagrama de contexto:** define los límites entre el sistema y las entidades que forman parte de su ambiente.
- **Casos de uso:** describen los escenarios en los que se desenvuelven los actores del sistema.
- **Arquitectura conceptual:** representa el diseño estructural del proyecto visto desde un punto de vista genérico.
- **Diagrama de clases:** muestra la disposición de clases de la herramienta.
- **Diagrama de despliegue y componentes:** incluye los dispositivos físicos, sus interconexiones y sus artefactos de software.
- **Diagrama de secuencia:** recorre las acciones ejecutadas por las entidades del sistema durante su operación.

3.1. Contexto de negocio

3.1.1. Antecedentes

El surgimiento de problemas a solucionar en las tareas de desarrollo de software es una constante inherente a ellas: nuevas decisiones han de ser tomadas en todo momento, y éstas marcan la dirección que seguirá personalmente cada programador con el fin de llegar al resultado esperado.

Esta libertad responde a la naturaleza abierta de la creación del software, donde un mismo objetivo puede ser abordado desde varias perspectivas, todas ellas con diferentes procedimientos, pero con una salida en común.

Sin embargo, aún en esta disciplina en la que, virtualmente, nunca se presentan escenarios donde a un problema corresponde una única solución, es común que cada elección implique costos diferentes en el rendimiento final del sistema, y a medida que esta brecha se abre queda claro que es menester del programador dar con aquel desenlace que, respetando el objeto fundamental, resulte menos costoso en términos generales.

Asimismo, cada elección podría implicar el uso de conocimientos diferentes, y la selección de la vía más eficiente podría requerir de una formación con la que los programadores posiblemente no cuenten, manifestándose, entonces, la necesidad de consultar al personal experto que se encuentre disponible.

No obstante, solicitar orientación puede resultar una labor compleja si se desconoce qué se desea lograr, cuáles son las tecnologías o los métodos de apoyo en el contexto de la solución que se requiere encontrar o a qué miembro del equipo se ha de consultar para ello.

Es por esto que la aplicación descrita a lo largo de este capítulo pretende llevar a cabo el análisis de perfiles de conocimiento mediante el modelo propuesto por (Rosas-Daniel, 2014), basándose en la valoración de dichos perfiles para así lograr identificar, dentro de un mismo equipo, quiénes son los contribuidores potenciales para la solución de problemas particulares en materia de desarrollo de software, centrándose en escenarios de entornos distribuidos geográficamente, dónde este tipo de situaciones se ven enfatizadas, por defecto, a causa de las distancias que segmentan al personal.

3.1.2. Necesidades del cliente o mercado

XII. Necesidades identificadas	
ID	Descripción
N01	Los equipos de trabajo necesitan un procedimiento de control del cual puedan valerse para llevar el registro de los perfiles de conocimiento de sus integrantes en forma segura.
N02	Los equipos de trabajo necesitan una herramienta que integre mecanismos para obtener y filtrar los perfiles de conocimiento de sus integrantes en base a consultas específicas.
N03	Los programadores requieren un método establecido para describir el conocimiento del que carecen para solucionar un problema, preferentemente a través de una lista de conceptos predefinidos. Esta lista debe ser exactamente idéntica en todas las instancias del sistema que hagan uso de ella.
N04	Tras haber establecido cuál es exactamente el problema que se desea resolver, los programadores precisan de un recurso a través del cual les sea posible valorar qué persona dentro de su equipo de trabajo se encuentra mejor capacitada para brindarles soporte en la búsqueda de una solución en concreto para el problema que se presenta.
N05	Deben suministrarse opciones a los programadores para filtrar a los expertos encontrados según su localización geográfica. De igual modo, otros parámetros para refinar la búsqueda de colaboradores también pueden ser considerados a solicitud del usuario.
N06	Es necesario establecer un medio de comunicación a través del cual el usuario pueda exponer su problemática en cuestión al colaborador seleccionado, ya sea como parte de la misma solución o de forma externa.

3.1.3. Supuestos e implicaciones previas a la implementación

XIII. Supuestos e implicaciones	
ID	Descripción
S01	Los equipos de trabajo se encuentran familiarizados con algún método de construcción de perfiles de conocimiento de personal.
S02	Los equipos de trabajo disponen de un repositorio o base de datos para registrar los perfiles de conocimiento de sus integrantes en forma remota.
S03	Los integrantes de los equipos consienten que se haga difusión de sus datos personales de contacto, de manera que estos puedan facilitarse a sus compañeros de trabajo.

3.2. Visión de la solución

3.2.1. Enunciado de visión de la solución

Diseñar una aplicación que permita a los programadores que se desenvuelven en entornos de desarrollo global encontrar, dentro de sus equipos, personal capaz de proveer una solución a los problemas que puedan presentarse en sus tareas relacionadas con el desarrollo y/o prueba del software.

3.2.2. Características primarias del sistema

XIV. Características requeridas			
ID	Descripción	Prioridad	Necesidad asociada
C01	Los programadores pueden especificar, mediante un <i>checklist</i> , cualquier conjunto de conocimientos que competa a la naturaleza de su problema e introducirlos a la aplicación móvil con el fin de encontrar personal experto.	Alta	N03
C02	Con el fin de efectuar el empate del perfil de conocimiento solicitado con el perfil del colaborador idóneo, se integra el modelo de valoración de perfiles a una API web, a fin de que pueda consumirse desde la aplicación móvil a modo de servicio.	Alta	N04
C03	A través del servicio de valoración de perfiles de conocimiento, se entrega al programador el dictamen que indica cuál de los disponibles es el perfil más apto para brindarle asistencia.	Alta	N02, N04
C04	Se hace uso de una ontología de conocimiento en desarrollo y prueba del software como un consenso claro alrededor de las categorías y términos que pueden distinguirse al hacer uso de la herramienta.	Alta	N03
C05	Los programadores pueden elegir a qué colaborador solicitar apoyo en aquellos casos en los que se les presente a más de un colaborador candidato.	Alta	N04
C06	La aplicación permite a los programadores acceder a la información de contacto de los expertos seleccionados.	Alta	N06

3.2.3. Características secundarias del sistema

XV. Características deseables			
ID	Descripción	Prioridad	Necesidad asociada
C07	Previo al envío de la consulta, la aplicación permite establecer un filtro geográfico, de modo que los programadores pueden, también, procurar expertos según la ubicación en la que radiquen.	Media	N05
C08	La aplicación incluye un mecanismo para describir el status de los contribuidores, acorde a su disponibilidad, al momento de la búsqueda.	Baja	N04
C09	La aplicación permite a los programadores contactar internamente a los expertos desde un módulo de intercambio de mensajes.	Baja	N06

3.3. Alcance de la solución

3.3.1. Alcance por entrega

XVI. Entregas distinguidas		
Versión de entrega	Funcionalidad integrada	Características incluidas
1.0	Módulo de listado de conocimientos, búsqueda de perfiles de colaboradores y dictamen	C01, C02, C03, C04, C05, C06
2.0	Filtro de expertos según ubicación geográfica	C07
3.0	Módulo de intercambio de mensajes e integración de factores de disponibilidad	C08, C09

3.3.2. Limitaciones y exclusiones

- Durante la fase inicial del proyecto no serán considerados requerimientos relacionados con la interacción de los programadores con los colaboradores. En su lugar, la aplicación se centrará inicialmente en el proceso de identificación de colaboradores candidatos.

- Las interfaces se desarrollarán en forma minimalista durante la etapa de elaboración del prototipo, para después modificarse con el diseño final una vez que la funcionalidad total del sistema haya sido programada e implementada con éxito.
- Las funciones de comunicación dentro de la aplicación se acotarán, en un principio, al envío de correos electrónicos entre usuarios. No se generarán notificaciones de parte del programador hacia el colaborador dentro de la aplicación, tal cual lo especifica la característica C09 del cuadro anterior. De ser necesarios otros medios, éstos se detallarán durante la fase de implementación del sistema.

3.4. Contexto del sistema

3.4.1. Ambiente de operación

- Despliegue en dispositivos Android 5.1.1 en adelante (API Level 22)
- REST como método de comunicación con servicios web (véase sección 4.2)
- JSON como formato de intercambio de datos (véase sección 4.2)

3.4.2. Modelado de procesos de negocio

Acorde al modelo detallado en la figura 1, entre los procesos llevados a cabo por la aplicación resaltan:

- Los programadores pueden formular una búsqueda especificando cualquier conjunto de conocimientos que consideren necesario.
- La aplicación muestra el perfil de la persona más apta para brindar apoyo según la consulta introducida.
- Los programadores pueden elegir a qué colaborador solicitarán apoyo, en dado caso de que la aplicación presente más de un colaborador candidato.

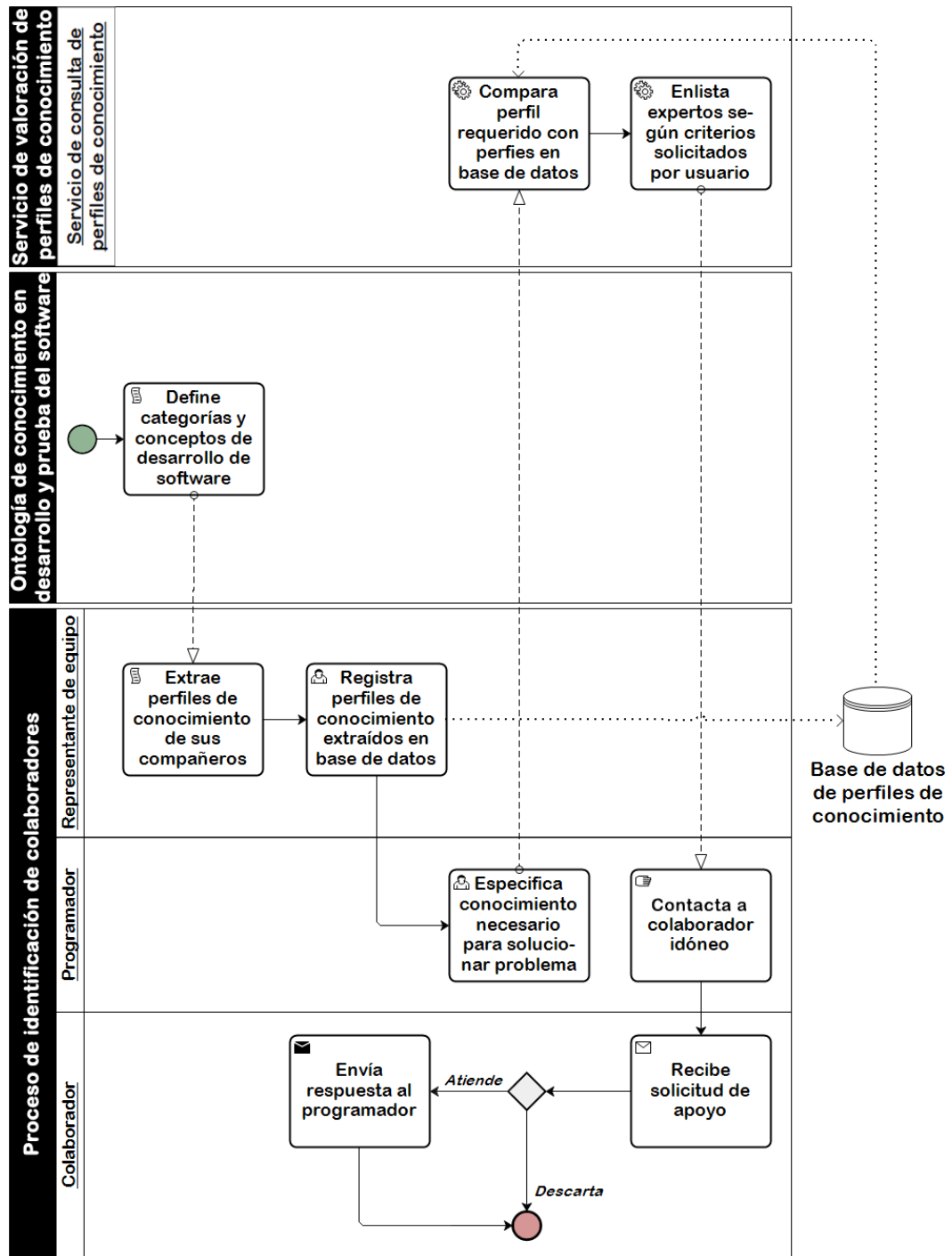


Fig. 1: Modelado de procesos de negocio

3.4.3. Diagrama de contexto

La figura 2 define los límites entre el sistema y su ambiente, ilustrando en un diagrama de contexto a los principales actores que asumen un rol en su funcionamiento, junto con la información que introducen u obtienen del mismo.

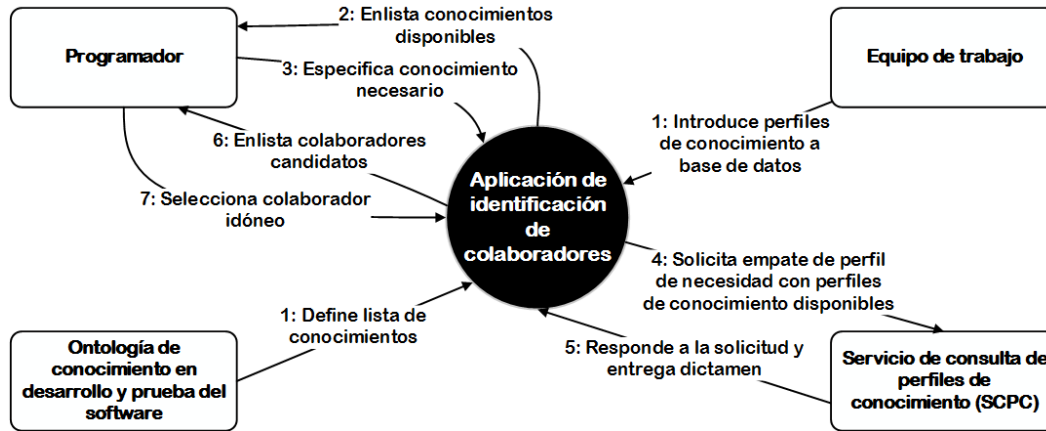


Fig. 2: Diagrama de contexto

3.5. Requerimientos de arquitectura

3.5.1. Drivers funcionales y casos de uso

XVII. Casos de uso		
ID	Descripción	Implicaciones o características asociadas
CU01	El representante de equipo da de alta los perfiles de los integrantes.	S01, S02
CU02	El SVPC evalúa y asigna una valoración a los perfiles de los integrantes.	S02, C02
CU03	El programador selecciona los conocimientos que requiere para la solución de su problema.	C01, C04
CU04	Tras estructurar su perfil de necesidades, el programador inicia una búsqueda de contribuidores.	C01, C02, C04
CU05	El SCPC busca a los candidatos cualificados.	C02, C03

XVII. Casos de uso		
ID	Descripción	Implicaciones o características asociadas
CU06	El SCPC muestra el dictamen al programador a través de la aplicación móvil.	C03
CU07	Tras recibir una respuesta, el programador puede solicitar apoyo al experto que se le ha señalado.	S03, C05, C06

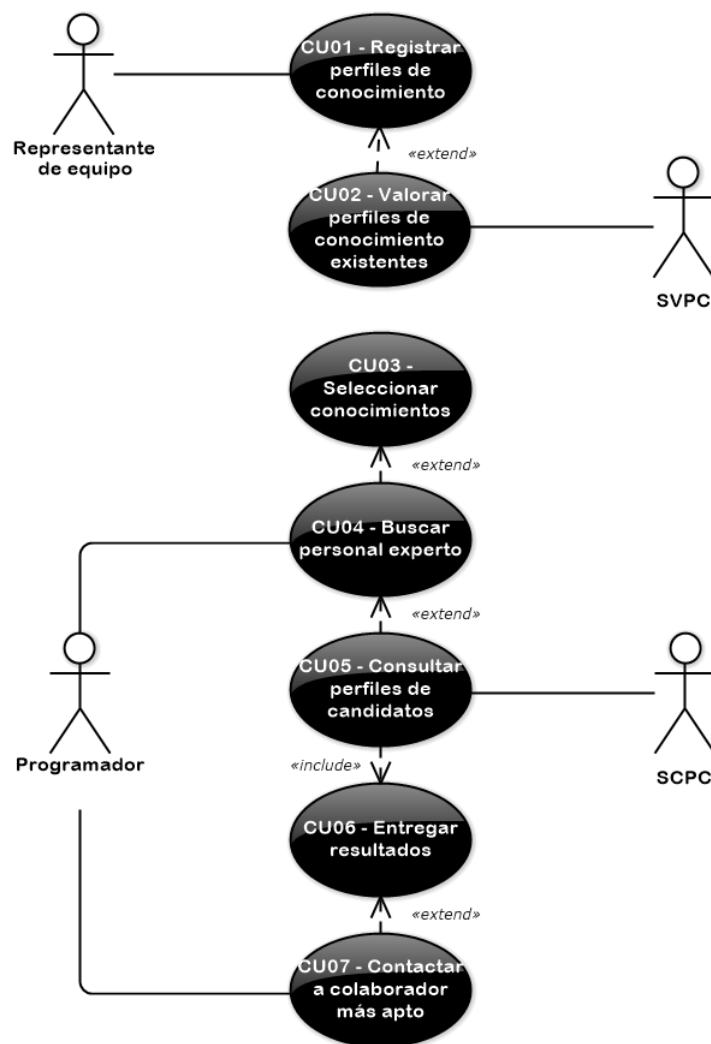


Fig. 3: Diagrama de casos de uso

3.5.2. Drivers de restricciones

XVIII. Restricciones	
Tipo de restricción	Descripción
De usuario	El usuario deberá observar sólo conocimientos relacionados con el desarrollo o prueba de software entre los conocimientos que aparecerán listados en el módulo de búsqueda de perfiles de colaboradores.
De desarrollo	Todas las librerías externas seleccionadas para la implementación de la solución deben ser de uso gratuito.
	Los puntos descritos en el apartado de ambiente de operación (3.4.1) deberán cumplirse en forma obligatoria.
	Las características secundarias del sistema sólo serán implementadas una vez que quede documentado el funcionamiento correcto de todas las características primarias.

3.5.3. Drivers de atributos de calidad en base a escenarios de calidad

Los siguientes drivers fueron definidos acorde a los atributos de calidad identificados en la norma ISO/IEC 9126-1 (International Organization for Standardization, 2000):

ID	EAC01
Atributo externo	Funcionalidad
Atributo interno	Cumplimiento de funcionalidad
Fuente	Programador
Estímulo	Se lleva a cabo una búsqueda de colaboradores
Artefacto	Módulo de búsqueda de perfiles de colaboradores
Entorno	Tiempo de ejecución
Respuesta	La aplicación muestra al programador una lista de colaboradores potenciales que podrían asistirle en la solución de su problema.
Medida	Todos los colaboradores potenciales expuestos, cumplen efectivamente con el perfil de necesidad extraído de la consulta.

ID	EAC02
Atributo externo	Funcionalidad
Atributo interno	Precisión
Fuente	Proceso de identificación de colaboradores
Estímulo	El resultado de la consulta es mostrado al usuario
Artefacto	Módulo de dictamen
Entorno	Tiempo de ejecución
Respuesta	La aplicación muestra al programador solamente perfiles que contienen los conocimientos especificados por el usuario.
Medida	Los perfiles son ordenados en forma descendente acorde al porcentaje de correspondencia para con el perfil de necesidad extraído de la consulta.

ID	EAC03
Atributo externo	Eficiencia
Atributo interno	Comportamiento en el tiempo
Fuente	Proceso de identificación de colaboradores
Estímulo	Se lleva a cabo una búsqueda de colaboradores
Artefacto	Módulo de búsqueda de perfiles de colaboradores
Entorno	Tiempo de ejecución
Respuesta	El dictamen es entregado al usuario dentro del tiempo límite.
Medida	Tiempo estimado para valoración (menor a 20 segundos).

ID	EAC04
Atributo externo	Mantenibilidad
Atributo interno	Modificabilidad
Fuente	Desarrollador
Estímulo	Se desea agregar una función de contacto dentro de la aplicación
Artefacto	Código de proyecto de aplicación móvil de identificación de colaboradores
Entorno	Tiempo de desarrollo
Respuesta	Nuevas funciones pueden ser agregadas modularmente sin afectar el comportamiento de las funciones actuales.
Medida	Cantidad de funciones que pudieran agregarse en posteriores versiones de la aplicación.

ID	EAC05
Atributo externo	Portabilidad
Atributo interno	Adaptabilidad
Fuente	Desarrollador
Estímulo	La aplicación puede llevarse a otras plataformas
Artefacto	Código de proyecto de aplicación móvil de identificación de colaboradores
Entorno	Tiempo de desarrollo
Respuesta	El código de la aplicación puede versionarse para otras plataformas tal cual como se encuentra, obviando los cambios que deban hacerse al seguir las convenciones del lenguaje nativo que corresponda a la nueva plataforma seleccionada.
Medida	Versiones para otros sistemas operativos móviles.

3.6. Diseño de la arquitectura

3.6.1. Estilo arquitectónico seleccionado

Dado que se busca que el modelo de valoración de perfiles de conocimiento pueda ser consumido de manera externa, se adopta un estilo arquitectónico de tipo Cliente/Servidor para dar respaldo a la solución.

En este modelo, cada equipo o proceso en la red es un cliente capaz de comunicarse y enviar solicitudes, o un servidor encargado de emitir la respuesta y proveer recursos o servicios a través de la web. (Nishida y Nguyen, 2013)

Su ventaja más evidente radica en librar a los clientes de las responsabilidades de procesar y almacenar la información, haciendo la estructura del sistema más escalable. (Zhu et al., 2016)

Trasladando este esquema de funcionamiento a un caso práctico, y haciendo uso del equipo del laboratorio de Ingeniería de Software de la Maestría en Ciencias de la Computación, el nuevo servicio de valoración de perfiles de conocimiento será montado en un servidor del Instituto Tecnológico de Hermosillo, y los clientes, serán entonces, los dispositivos móviles en los que se lleve a cabo el despliegue del prototipo.

3.6.2. Patrón arquitectónico seleccionado

La aplicación móvil emplea el patrón MVA (*Model-View-Adapter*), haciendo uso de un componente mediador para facilitar su implementación mediante el uso de directivas implementadas en el código.

MVA es una variante del patrón MVC (*Model-View-Controller*), mismo que a su vez es un patrón reconocido por separar la lógica del negocio, o el modelo, de la presentación de los datos, o la vista. (Hintze, 2009)

El patrón MVA parte de la separación absoluta del modelo y la vista, habilitada a través de adaptadores que actúan como mediadores sin la necesidad de que el desarrollador especifique su lógica cada vez que exista la necesidad de que el modelo y la vista interactúen entre sí. (Zamudio et al., 2012)

Sin embargo, fuera del dominio del cliente, el sistema, visto como una conjunción de la lógica de la aplicación y de los procesos ejecutados del lado del servidor, utiliza el patrón MVVW (*Model-View-ViewModel*). Este patrón extrae el comportamiento de presentación de la vista, separándolo de las cuestiones del desarrollo de la interfaz de usuario. (Sørensen y Mihailesc, 2010)

Bajo este esquema, varios de los elementos del sistema presentan enlaces bidireccionales entre sí, y el flujo resultante es el siguiente:

1. Mediante la vista, el usuario emite una solicitud en busca de colaboradores candidatos, la cual es dirigida primeramente hacia el adaptador.
2. El adaptador recibe la solicitud de la vista y la estructura dentro de un objeto JSON antes de expedirla al modelo de la vista.
3. El modelo de la vista solicita la acción del servicio de consulta de perfiles de conocimiento enviando el objeto JSON dentro de una petición HTTP al modelo del servidor empleando REST.
4. El modelo del servidor procesa la entrada de datos y el SCPC realiza una nueva búsqueda cuyos resultados se remiten al modelo de la vista.
5. El modelo de la vista dirige la réplica del SCPC hacia el adaptador, y este se encarga de reestructurarla.
6. El adaptador entrega la respuesta final a la vista: el usuario puede, ahora, visualizar el dictamen del SCPC y establecer contacto con los expertos.

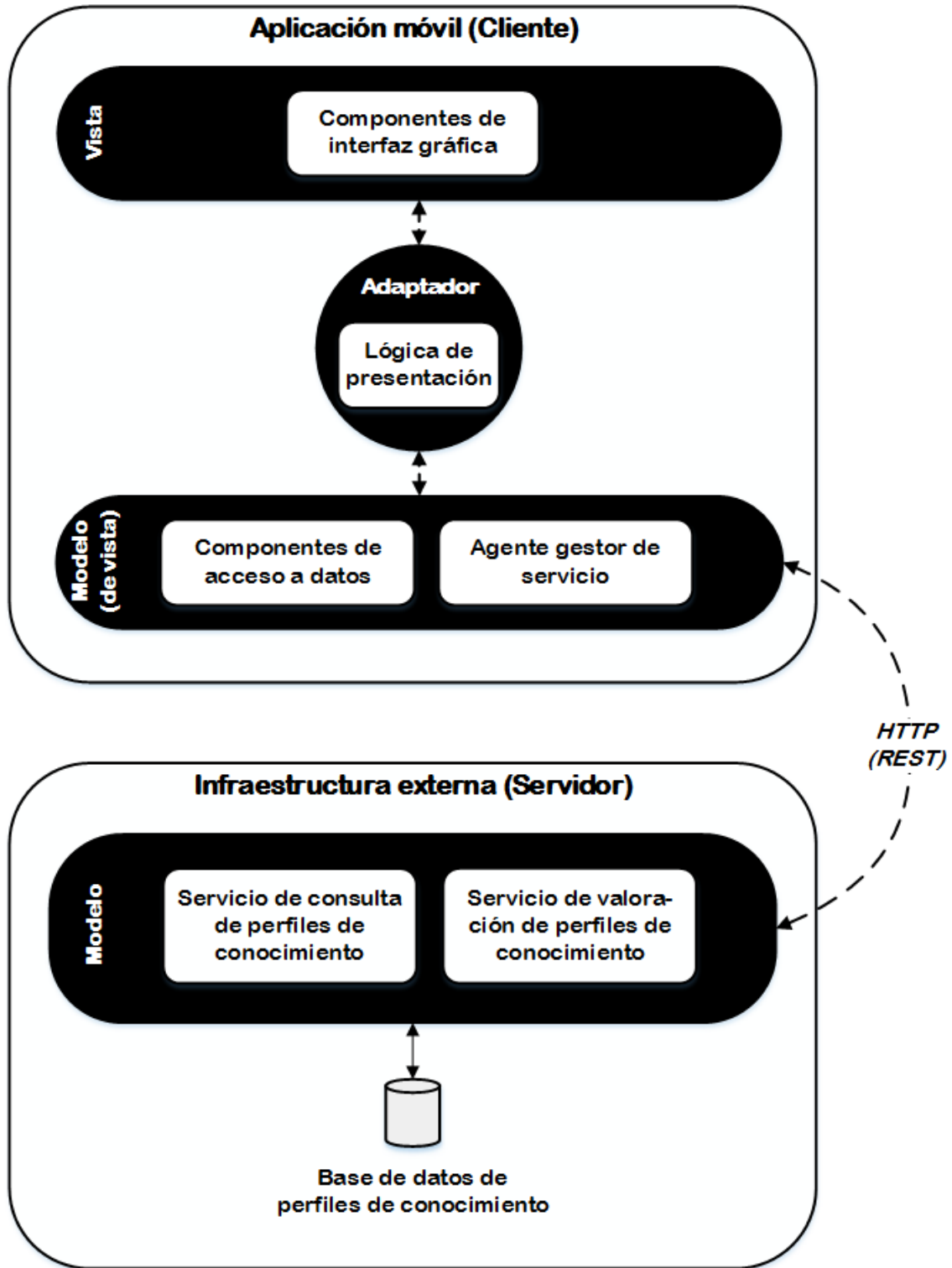


Fig. 4: Arquitectura conceptual

3.6.3. Componentes identificados y su funcionalidad

3.6.3.1. Funcionalidad comprendida por el SVPC

La figura 5 ilustra los nodos, componentes y artefactos que conforman la lógica del servidor, como producto de la adaptación de las librerías del modelo de valoración a una API web para hacer factible su consumo por parte de aplicaciones externas. Posteriormente, la figura 6 describe a detalle las clases dentro del servicio de valoración de perfiles de conocimiento, incluyendo aquellas que realizan los procesos de lógica difusa del FIS (*Fuzzy Inference System* ó Sistema de Inferencia Difuso).

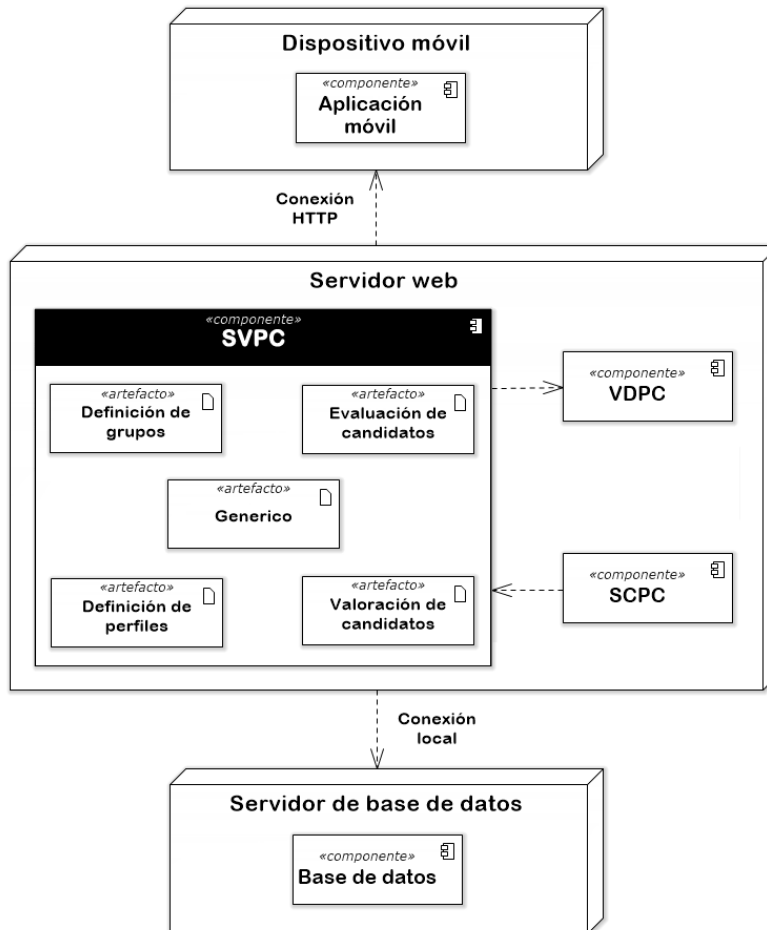


Fig. 5: Despliegue de componentes del SVPC de (Rosas-Daniel, 2014)

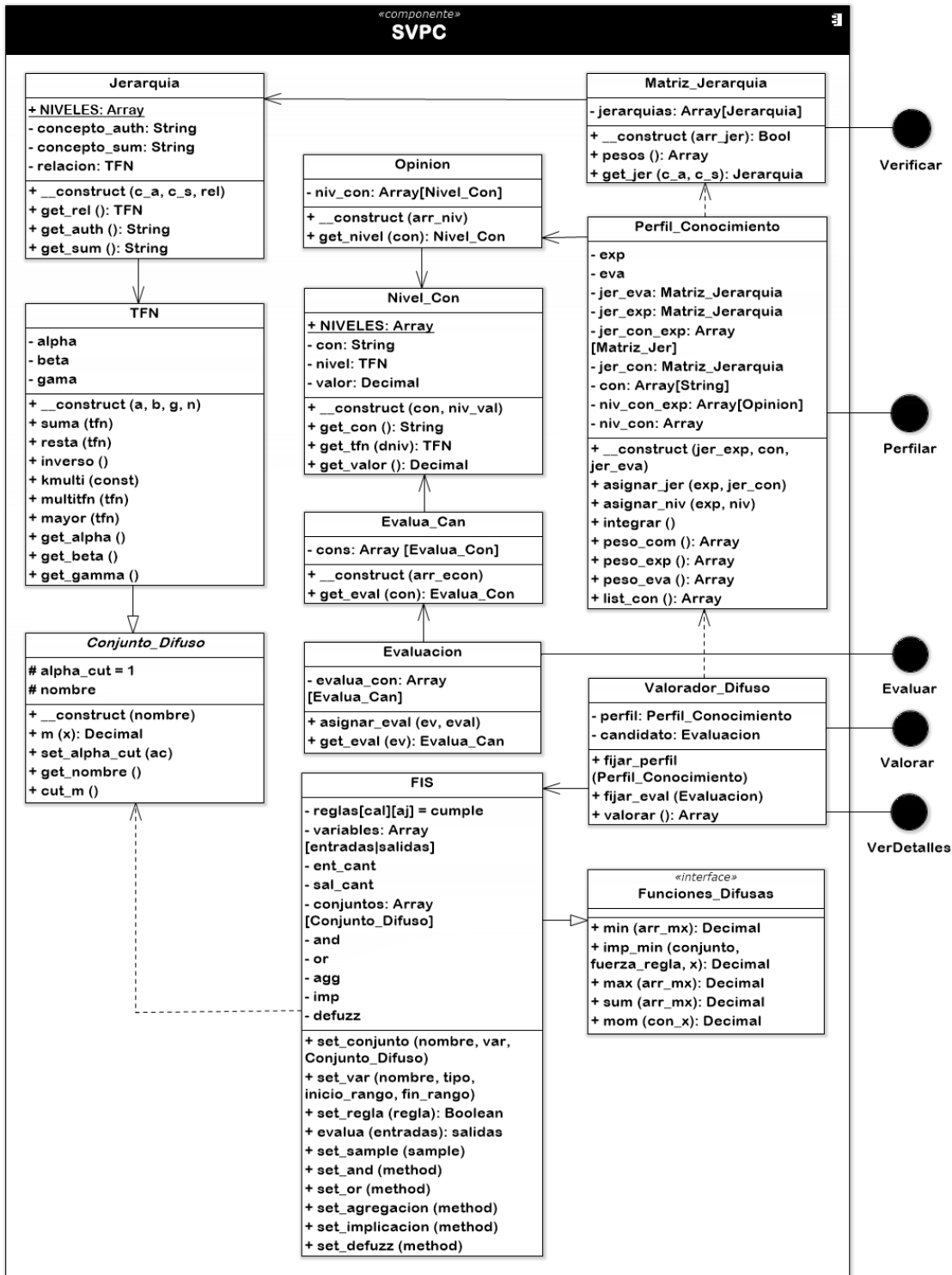


Fig. 6: Diagrama de clases del SVPC y FIS de (Rosas-Daniel, 2014)

3.6.3.2. Funcionalidad comprendida por el SCPC

Las arquitecturas middleware definen una lógica de intercambio de información siguiendo un paradigma de petición y respuesta en el que el cliente solicita al servidor que le permita hacer uso de un servicio en particular, ya sea a través de un mensaje estructurado como petición o usando RMI (invocación remota de métodos), favoreciendo la implementación de aplicaciones distribuidas a gran escala. (Pietzuch y Bacon, 2002)

El servicio de consulta de perfiles de conocimiento, junto con el servicio de valoración, conforma la capa de middleware del sistema. La aplicación móvil debe establecer contacto en primera instancia con el SCPC, el cual ordena y canaliza los parámetros recibidos, para así brindar acceso a la base de datos de perfiles de conocimiento que se alberga en el servidor.

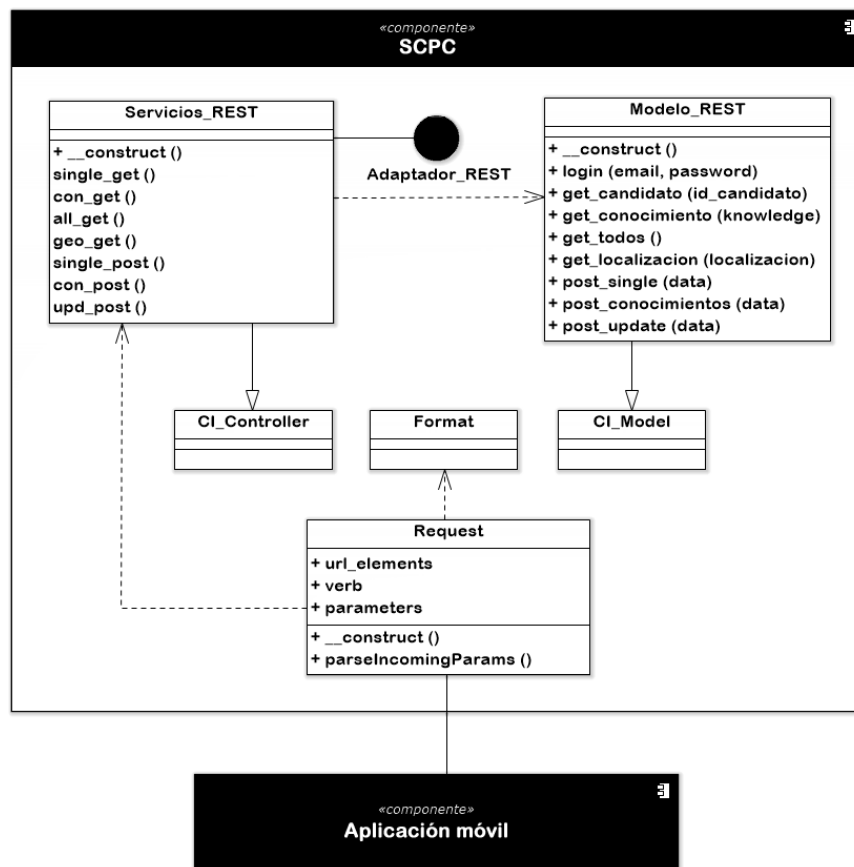


Fig. 7: Diagrama de clases del SCPC

3.6.3.3. Funcionalidad comprendida por la aplicación móvil

La figura 8 da a conocer a detalle la interacción del usuario con los distintos procedimientos del sistema, cuyas líneas de vida parten desde la perspectiva del cliente de Android y ocurren dentro de la siguiente secuencia:

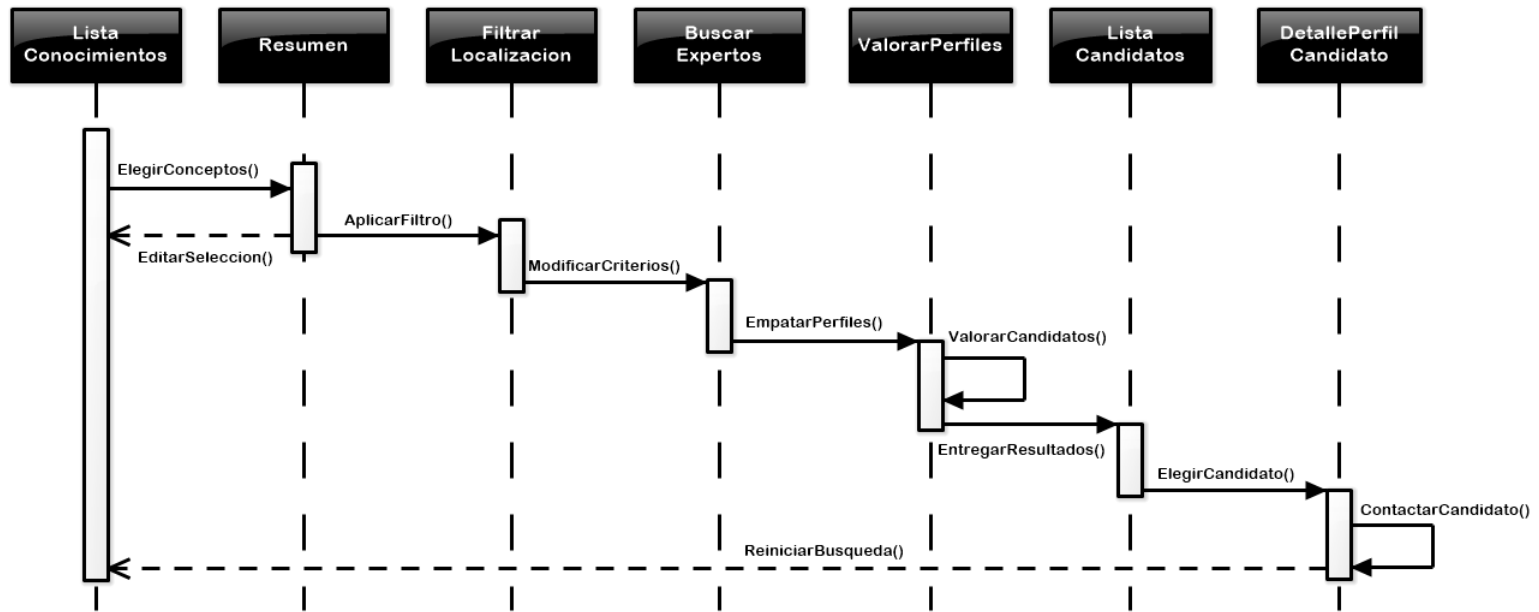


Fig. 8: Diagrama de secuencia del sistema

Tras la inclusión de esta última perspectiva arquitectónica del proyecto, quedan registrados todos los detalles de diseño en torno a los procesos de negocio que cubren la aplicación móvil de apoyo en la identificación de colaboradores y sus servicios de apoyo.

Capítulo 4

Implementación del sistema

El propósito que persigue la fase de implementación de un sistema informático es preparar el camino para el despliegue del software desarrollado con el fin de colocarlo al alcance de sus usuarios objetivos, descubriendo, acto seguido, cuales serán, *a posteriori*, los medios para su transición hacia la fase de mantenimiento. (Pataki et al., 2003)

Las siguientes páginas tratan a fondo los temas alusivos a la función de los componentes de software que juegan un papel importante en la implementación de la aplicación móvil de identificación de colaboradores, abordando los siguientes procedimientos:

- Establecimiento de los mecanismos fundamentales para el funcionamiento de la herramienta de cara al usuario, figurando entre estos el uso de una ontología de conocimiento y la definición explícita de los métodos para la extracción y almacenamiento de perfiles de conocimiento en el sector empresarial.
- Asentamiento de un ambiente de operación usando las tecnologías idóneas para lograr la implementación exitosa de la aplicación móvil, integrando una base de datos de perfiles de conocimiento junto con los servicios web de consulta y valoración de perfiles dentro del servidor.
- Desarrollo del cliente y despliegue final de ambas correspondencias del sistema, siguiendo el esquema de arquitectura detallado en el anterior capítulo.

4.1. Técnicas base para la implementación de la propuesta

En esta sección se definen dos mecanismos clave como sustento a la gestión del conocimiento organizacional.

Si bien, estas técnicas no generan un impacto de gran magnitud en el proyecto en el plano arquitectónico, el funcionamiento adecuado de la herramienta desarrollada, tal como se plantea en este trabajo de tesis, depende en su totalidad de la consolidación de estas bases.

4.1.1. Definición de una ontología de conocimiento en desarrollo y prueba del software

Una ontología es una conceptualización arbitrada y utilizada como medio de representación de la información existente en torno a un área de conocimiento en particular. (Staab y Brewster, 2004)

Hacer uso de una ontología significa organizar exhaustivamente las percepciones informales de un dominio de información y convertirlas en una especificación semi-formal, a través de una serie de representaciones intermedias que pueden ser fácilmente comprendidas tanto por los expertos del área, como por los desarrolladores de la ontología misma. (Velázquez-Mendoza, 2013)

Una de la principales metas del uso explícito de ontologías de conocimiento es, pues, delimitar formalmente el conocimiento subyacente a una determinada disciplina, en tal modo que el resultado refleje el consenso claro de la mayoría de los miembros que conforman la comunidad de expertos cualificados en dicha asignatura. (Vizcaíno et al., 2016)

Por este motivo, se ha establecido una ontología de conocimiento como un fundamento para entregar un compendio de términos y categorías inherentes al desarrollo y prueba del software, tal como se demuestra en la figura 9. Si bien esta especificación podría expandirse hasta alcanzar varias veces su extensión actual, esta primera versión permite a los programadores deducir y seleccionar temas básicos en los que podrían requerir asesoría, sin necesidad de destinar grandes esfuerzos para concretar este descubrimiento.

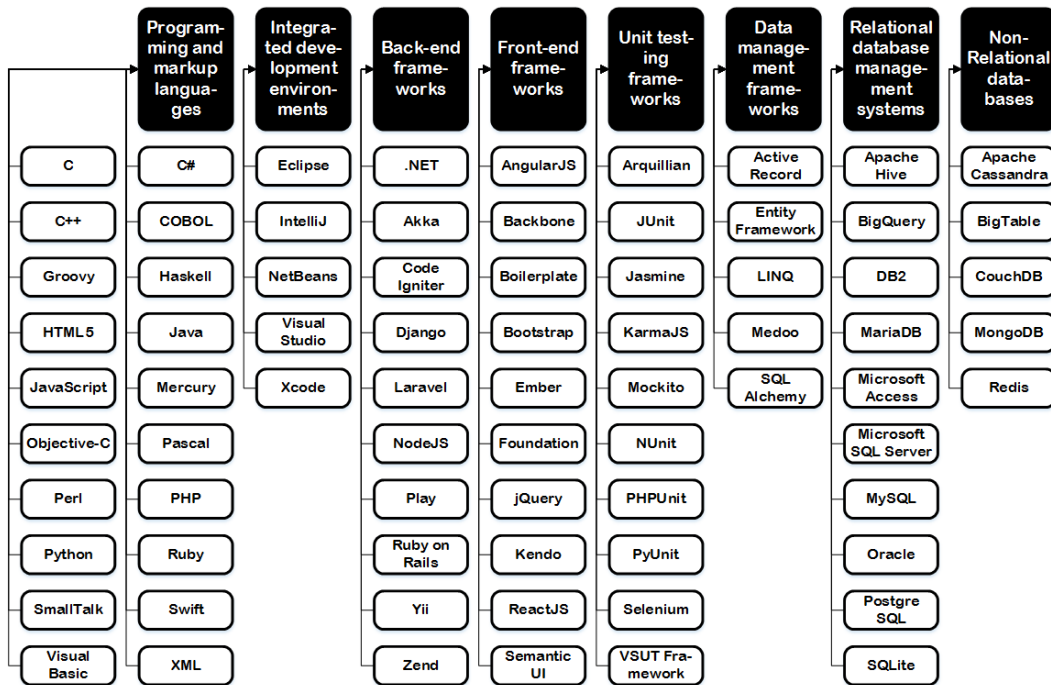


Fig. 9: Categorías y términos distinguidos por la ontología

4.1.2. Extracción y registro de perfiles de conocimiento

El interés que despierta proveer de una forma y una estructura al conocimiento ha sido abordado partiendo desde una gran variedad de disciplinas y teorías: la calidad y el impacto del *expertise* parece ser una preocupación constante para aquellos que se ven obligados a trabajar estrictamente con personas que cuenten con cualidades muy específicas. (Dochy y Valcke, 1992)

El enfoque de perfilación del conocimiento emerge como resultado de este hecho. Un perfil de conocimiento es un conjunto particular de rasgos que describen el conocimiento asociado a las capacidades que posee un individuo, como indicadores de su competitividad laboral. (Velázquez-Mendoza, 2013)

La administración de perfiles de conocimiento parte de la premisa de que las organizaciones deben construir y almacenar su conocimiento bajo un margen de trabajo arbitrado, creando índices que puedan ser localizados y consultados en cualquier momento con el propósito de reconocer y divulgar con

facilidad la información alusiva a las aptitudes del personal, dado que gestionar perfiles de conocimiento permite cotejar fácilmente las habilidades y grado de cualificación de las personas con los requisitos que estas han de cumplir para desempeñar adecuadamente una labor. (Oñate-Gómez et al., 2009)

La figura 10 ilustra en una red semántica las relaciones entre el conocimiento adquirido por un individuo y el conocimiento que se prescribe para desempeñar un rol dentro de una empresa, en el entendido de que estas requieren que los empleados o aspirantes a un puesto demuestren ser capaces de ejecutar las consignas que competen al mismo. (Rodríguez-Elías et al., 2017)

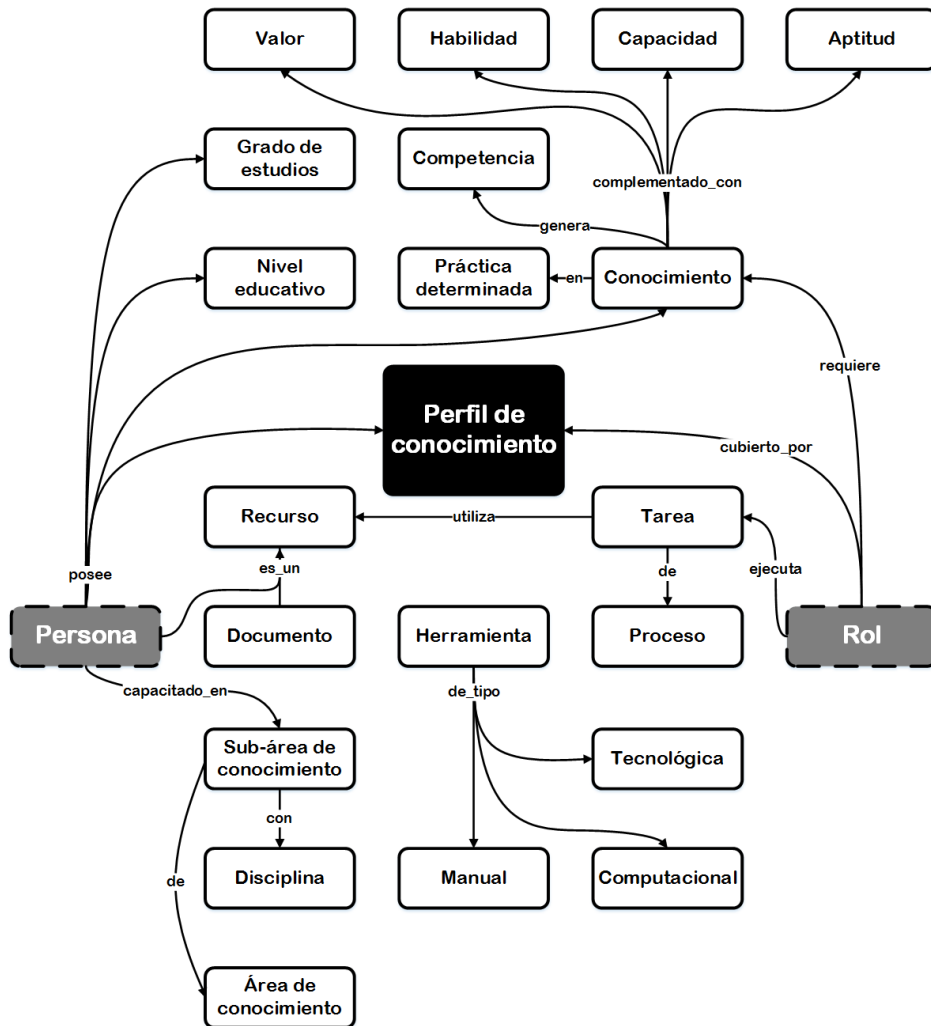


Fig. 10: Modelo de construcción de perfiles de conocimiento

Aunque es importante señalar que, por motivos prácticos, el despliegue del sistema descrito en el presente capítulo ocurre dentro de un entorno de prueba en donde se trabaja con información obtenida a partir de supuestos utilizando una base de datos de perfiles de conocimiento creada para fines exclusivamente demostrativos, el prototipo desarrollado por (Rosas-Daniel, 2014) puede resultar de utilidad en entornos de trabajo de campo, al dar soporte a los procesos de perfilación y evaluación del conocimiento del personal de una empresa empleando los cuatro pasos que pueden advertirse en la figura 11:

```

public function __construct($jer_exp,$con,$jer_eva)
{
    $this->_jer_exp = (is_array($jer_exp))? new matriz_jerarquia($jer_exp) : NULL;
    $this->_jer_eva = (is_array($jer_eva))? new matriz_jerarquia($jer_eva) : NULL;
    $this->_exp = (is_array($jer_exp))? array_keys($jer_exp) : NULL;
    $this->_eva = (is_array($jer_eva))? array_keys($jer_eva) : NULL;
    $this->_con = (is_array($con))? $con : NULL;
}

public function asignar_jer($exp,$jer_con)
{
    if( (is_string($exp) || is_integer($exp)) AND in_array($exp, $this->_exp) AND is_array($jer_con) )
    {
        $this->_jer_con_exp[$exp] = new matriz_jerarquia($jer_con);
    }
}

public function asignar_niv($exp,$niv)
{
    if( (is_string($exp) || is_integer($exp)) AND in_array($exp, $this->_exp) AND is_array($niv) )
    {
        $this->_niv_con_exp[$exp] = new opinion($niv);
    }
}

public function integrar()
{
    $WE = $this->_jer_exp->pesos();
    $SWE = array_sum($WE);
    $this->_niv_con = array();
    foreach ($this->_con as $con) {
        $Ci = new TFN(0,0,0,'Zero');
        foreach ($this->_exp as $exp) {
            $e_niv = $this->_niv_con_exp[$exp]->get_nivel($con)->get_tfn();
            $Ci = $Ci->suma($e_niv->kmulti($WE[$exp]));
        }
        $this->_niv_con[$con] = $Ci->kmulti(1/$SWE);
    }
    $this->_jer_con = array();
    foreach ($this->_con as $con1) {
        $this->_jer_con[$con1] = array();
        foreach ($this->_con as $con2) {
            $Cij = new TFN(0,0,0,'Zero');
            foreach ($this->_exp as $exp) {
                $e_jer = $this->_jer_con_exp[$exp]->get_jer($con1,$con2)->get_rel();
                $Cij = $Cij->suma($e_jer->kmulti($WE[$exp]));
            }
            $this->_jer_con[$con1][$con2] = $Cij->kmulti(1/$SWE);
        }
    }
    $this->_jer_con = new matriz_jerarquia($this->_jer_con);
}

```

Fig. 11: Creación de un perfil de conocimiento

Estas cuatro funciones de código guardan cada una un propósito distinto, cumpliéndose estos en el siguiente orden de acontecimiento:

1. `__construct()`: creación de una plantilla para un nuevo perfil de conocimiento.
2. `asignar_jer()`: definición de la jerarquía de conocimientos que contendrá el perfil.
3. `asignar_niv()`: asignación de una lista de conocimientos y niveles correspondientes.
4. `integrar()`: integración de las opiniones de los expertos de la empresa en cuanto a los niveles y jerarquías comprendidos.

4.2. Tecnologías base para la implementación de la propuesta

Esta sección introduce, en primera instancia, la descripción de conceptos que atañen a las tecnologías que actúan como directrices para el modo en que obra el sistema, para más adelante puntualizar los atributos de los componentes que se ubican del lado del servidor, y clarificar la sucesión de eventos que ocurre en la web cada vez que una nueva consulta es emitida por el cliente.

4.2.1. Base de datos de perfiles de conocimiento

Crear y mantener una base de datos concede a las organizaciones la posibilidad de resguardar y recuperar cualquier dato que resulte ser crítico para la ejecución de sus tareas, al disponer de un medio para centralizar toda su información y ratificar la obtención de resultados íntegros y exentos de ambigüedades. Estas ventajas hacen incuestionable la necesidad de las empresas por contar con una base de datos propia en forma previa a la implantación de la aplicación propuesta. Por tanto, la figura 12 muestra el esquema relacional de la base de datos utilizada por el prototipo de la aplicación, desde la cual los programadores obtienen los datos y perfiles de conocimiento de los expertos competentes.

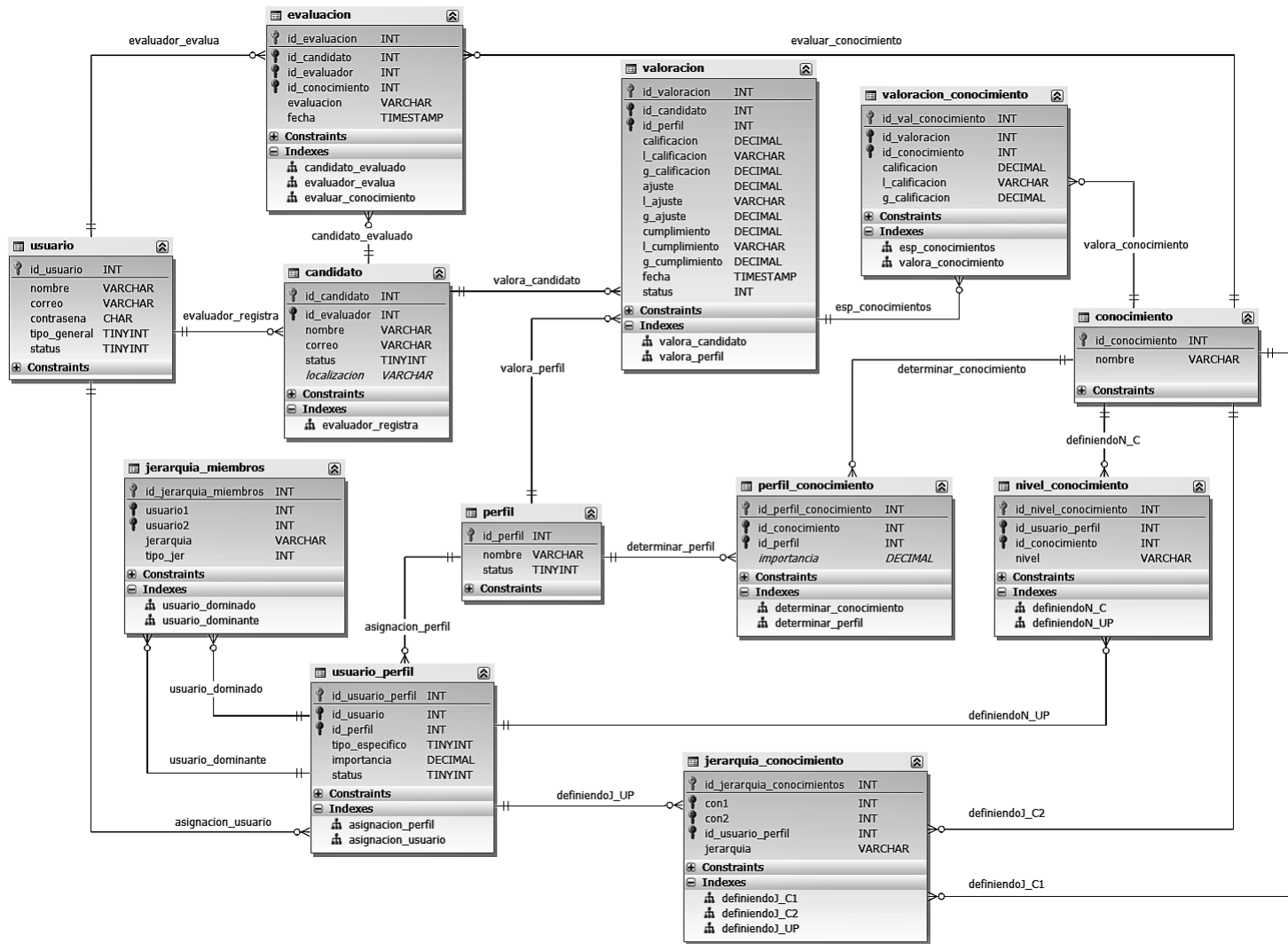


Fig. 12: Esquema relacional de la base de datos de perfiles de conocimiento

4.2.2. API REST de gestión de perfiles de conocimiento

Una API (*Application Programming Interface*) es una guía de reglas que se establece como interfaz entre aplicaciones computacionales para que estas puedan comunicarse a través de una red, permitiendo usar recursivamente una serie de funciones prediseñadas. (Merino, 2014)

Entre los pormenores que los proveedores de APIs deben especificar a los desarrolladores que deseen emplear su software, se encuentran (Spycher, 2012):

- Qué funcionalidad se provee y cuál es su disponibilidad.
- Cuáles son los límites en relación a las veces que se permite a un usuario o programa en especial utilizar la API, en aquellos casos en los que, en efecto, exista la intención de limitar su uso.
- Cómo se accede a la funcionalidad explícita, usualmente generando un contrato y entregando una llave a cada uno de aquellos desarrolladores comprometidos a seguir el reglamento prescrito.

Dentro del mismo campo, el término de “API web” se refiere a una interfaz a la que se accede vía Internet para formular peticiones mediante el método de HTTP correspondiente a la acción que se desea llamar, cuyas respuestas deben presentar una estructura pactada con anterioridad entre la API y las entidades que la consumen y apegarse a un arquetipo de comunicación. (Massé, 2012)

En el caso concurrente se dispone de REST (*Representational State Transfer* o transferencia de estado representacional) como protocolo de intercambio y manipulación de datos. REST es un conjunto coordinado de restricciones que intenta minimizar la latencia y la comunicación en red, al tiempo que maximiza la independencia y escalabilidad de las implementaciones de componentes. (Fielding y Taylor, 2002)

REST cataloga a los elementos arquitectónicos según sean datos, conectores o componentes, sin estar sujeto a ningún formato en específico, siempre y cuando todos los componentes puedan manipular correctamente los datos enviados a través de los conectores. De este modo, un servicio intermediario sólo necesita saber si es capaz de procesar las solicitudes que se le encomiendan, pudiendo omitir cualquier detalle semántico sobre los datos que estas contienen. (Jakl, 2005)

En lo concerniente a los formatos de transferencia de datos entre servicios web, JSON y XML se distinguen como las notaciones más comunes conocidas hasta hoy. JSON (*JavaScript Object Notation* o notación de objetos de JavaScript) es un formato de texto ligero, cuya ventaja frente a XML (*eXtensible Markup Language* o lenguaje de marcado extensible) radica en su facilidad de intercambio, al contener datos estructurados en vectores y registros, mientras que XML emplea estructuras de árbol cuyo procesamiento puede volverse complejo. (Vaidyanathan et al., 2011)

JSON permite la serialización de datos y su representación en cuatro tipos primitivos (cadenas de texto, números, valores booleanos y nulos) y dos estructurados (objetos y arreglos). (Bray, 2014)

Valiéndose de la flexibilidad que ofrecen los tres conceptos destacados en este apartado, el sistema suministra una sinergia de estas tecnologías al servirse de una API web que contiene servicios RESTful dedicados al procesamiento de entradas y salidas de objetos JSON a intercambiar entre cliente y servidor.

4.2.3. Servicio de valoración de perfiles de conocimiento o SVPC

Este servicio introduce un valorador difuso, gracias al cual se brinda soporte a tres actividades para la gestión de perfiles de conocimiento del personal:

- La perfilación o definición del perfil de conocimiento requerido para ejercer un rol.
- La evaluación o medición de los niveles de conocimiento con los que cuenta un candidato.
- La valoración o comparación del perfil de conocimiento del rol con el perfil de conocimiento del candidato obtenido por medio de la evaluación.

4.2.3.1. Registro de una nueva valoración

Dado que la propuesta trabaja con supuestos, asumiendo que estas valoraciones ya han sido llevadas a cabo por parte del personal experto de la empresa,

y que puede accederse a ellas al estar incorporadas dentro de una base de datos, la figura 13 se limita a mostrar el registro de una nueva valoración en su faceta más básica, tal como se instaura en el prototipo desarrollado por (Rosas-Daniel, 2014).

Este procedimiento sólo puede ser realizado una vez que se han verificado los lineamientos para definir el grado de cumplimiento obtenido por un candidato con respecto a un rol acorde a las reglas ingresadas en el valorador difuso, y que los procesos de fuzzificación del sistema de inferencia difuso o FIS han sido ejecutados.

```
public function registrar_valoracion($id_candidato, $id_perfil, $valoracion)
{
    $esp = $valoracion['especificacion'];
    foreach ($esp as $id_con => $valores) {
        $resp[$id_con]['calificacion'] = round($valores['calificacion'], 3);
        $resp[$id_con]['g_calificacion'] = round($valores['g_calificacion'], 3);
    }
    $this->db->trans_start();
    $success = $this->db->query('call P_Reg_Valoracion('.$id_candidato.', '.$id_perfil.', ".$this->array_to_string($esp)."',
        round($valoracion['calificacion'], 3).', '.$valoracion['l_calificacion']."', round($valoracion['g_calificacion'], 3).',
        round($valoracion['ajuste'], 3).', '.$valoracion['l_ajuste']."', round($valoracion['g_ajuste'], 3).',
        round($valoracion['cumplimiento'], 3).', '.$valoracion['l_cumplimiento']."', round($valoracion['g_cumplimiento'], 3).',
        $valoracion['status'].', @respuesta)');
    $success->next_result();
    $success->free_result();
    $query = $this->db->query('select @respuesta as out_param');
    $this->db->trans_complete();
    if(is_numeric($query->row()->out_param))
        return $query->row()->out_param;
    else
        return FALSE;
}
```

Fig. 13: Función del SVPC para asignar una valoración al perfil de un candidato

En caso de que se desee obtener mayores detalles sobre los procesos procedentes del servicio de valoración de perfiles de conocimiento, es altamente recomendable revisar la documentación del prototipo citado anteriormente.

4.2.4. Servicio de consulta de perfiles de conocimiento o SCPC

Este servicio proporciona al usuario la facultad de recuperar las valoraciones asignadas previamente a los perfiles de conocimiento de los candidatos, de forma que los resultados de estas puedan desplegarse en diversos dispositivos sin necesidad de aplicar ningún esfuerzo adicional.

La figura 14 muestra la función `con_get()` del SCPC, la cual da por comenzada una consulta de candidatos expertos que cuenten con los conocimientos requeridos, y cuyos perfiles se encuentren almacenados en la base de datos representada en la sección 4.2.1.

```
function con_get()
{
    $this->load->model('Modelo_rest');
    $user = $_SERVER['PHP_AUTH_USER'];
    $pass = MD5($_SERVER['PHP_AUTH_PW']);
    $log = $this->modelo_rest->login($user, $pass);
    if ($log)
    {
        $this->load->model('modelo_rest');
        if (!$this->get('know'))
        {
            $this->output
                ->set_header('HTTP/1.1 404');
        } else {
            $user = $this->modelo_rest->get_conocimiento($this->get('know'));
            if ($user == null)
            {
                $this->output
                    ->set_header('HTTP/1.1 204');
            } else {
                $this->output
                    ->set_header('HTTP/1.1 200 OK')
                    ->set_content_type('text/json', 'utf-8')
                    ->set_output(json_encode($user));
            }
        }
    } else {
        $this->output
            ->set_header('HTTP/1.1 403');
    }
}
```

Fig. 14: Función del SCPC para dar inicio a una búsqueda de conocimiento

4.2.4.1. Empate de perfiles de conocimiento existentes con perfil de necesidad extraído desde la consulta del usuario

Una vez que, mediante el código anterior, se entabla una nueva búsqueda de candidatos, se procede a invocar al modelo de REST para utilizar su función `get_conocimiento()`.

Esta función, vista a detalle en la figura 15, toma como argumento el conocimiento necesario para realizar la búsqueda en la base de datos, y devuelve un arreglo que engloba la información de aquellos candidatos existentes que cumplen con el perfil de necesidad extraído de la consulta del usuario.

Otras funciones más avanzadas del modelo de REST utilizan parámetros complementarios como la localización de los candidatos, de modo que los pro-

gramadores puedan acotar su búsqueda sólo a aquellos expertos que radiquen en el área que ellos indiquen.

```

public function get_conocimiento($knowledge)
{
    $enlace = mysqli_connect("localhost", "root", "PosgradoIS", "svpc");

    if (!$enlace) {
        echo "Error: No se pudo conectar a MySQL." . PHP_EOL;
        echo "Error de depuración: " . mysqli_connect_errno() . PHP_EOL;
        echo "Error de depuración: " . mysqli_connect_error() . PHP_EOL;
        exit;
    }

    $sql = "select ca.id_candidato, ca.nombre, ca.correo, ca.localizacion, co.nombre, e.evaluacion, e.fecha from
            candidato as ca, evaluacion as e, conocimiento as co where co.nombre = '". $knowledge. "' and
            ca.id_candidato = e.id_candidato and e.id_conocimiento = co.id_conocimiento order by ca.nombre";
    $result=$enlace->query($sql);

    if ($result->num_rows > 0){
        $arreglo = [];
        $output = $result->fetch_all();

        foreach ($output as $o){
            if(!isset($arreglo[$o[0]])){
                $arreglo[$o[0]] = [
                    'nombre_candidato'=>$o[1],
                    'correo'=>$o[2],
                    'localizacion'=>$o[3]
                ];
            }
            $arreglo[$o[0]]['conocimientos'][] = array(
                'conocimiento'=>$o[4],
                'evaluacion'=>$o[5],
                'fecha_evaluacion'=>$o[6]
            );
        }
        return $arreglo;
    } else {
        return null;
    }
    mysqli_close($enlace);
}

```

Fig. 15: Función del SCPC para encontrar perfiles candidatos según parámetros establecidos

4.3. Aplicación móvil de identificación de perfiles de conocimiento

Como pieza de desenlace del proyecto e instrumento de comprobación de la funcionalidad del sistema, surge, finalmente, la aplicación móvil de apoyo en la identificación de colaboradores en entornos de desarrollo global de software.

Se incluyen en este apartado las cuatro vistas centrales del prototipo desarrollado, así como una breve descripción de los eventos que se suscitan en cada una de ellas.

4.3.1. Selección de conocimientos a integrar en el perfil de necesidad

La primera vista de la aplicación muestra en una lista expandible las categorías y términos que abarca la ontología de conocimiento en desarrollo y prueba del software, en consonancia con lo expuesto en la sección 4.1.1.

A través de la selección de elementos dispuestos dentro de la lista que aparece en la figura 16, los programadores pueden expresar la naturaleza del problema que buscan resolver, dejando claro cual será el perfil de necesidad que los expertos deberán cubrir.

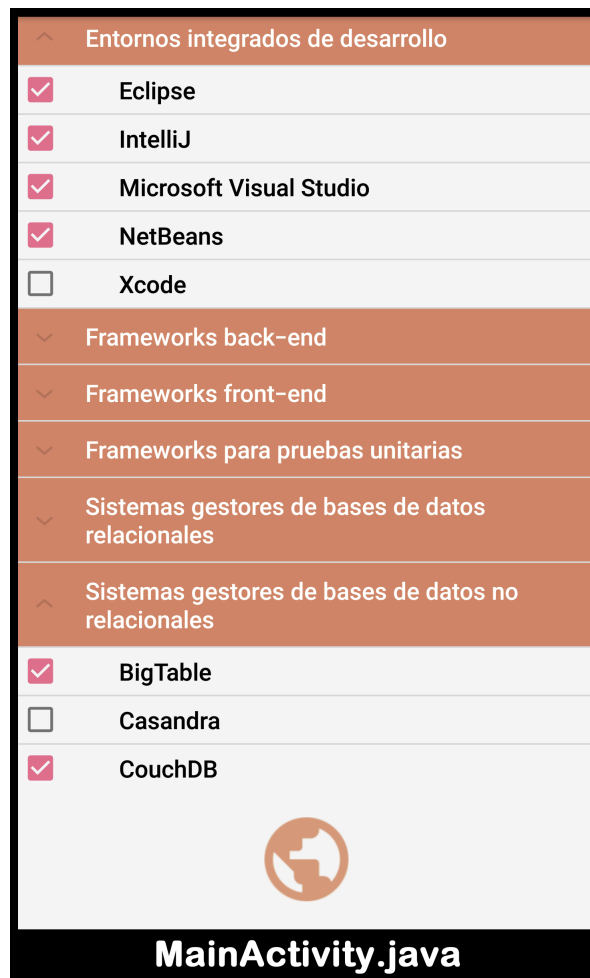


Fig. 16: Vista de categorías y conceptos

4.3.2. Sumarización de conceptos y uso de filtro geográfico para perfiles de expertos

El propósito de la vista incluida en la figura 17 es ofrecer a los programadores un punto de verificación desde el cual puedan revisar el conjunto de conocimientos seleccionado, y desplazarse de vuelta hacia la vista principal en dado caso de que se percaten de que es necesario aplicar modificaciones al mismo.

Igualmente, desde esta interfaz los programadores podrán llamar al cuadro de diálogo de aplicación de filtros, al presionar el botón ubicado en la parte central-inferior de la pantalla.

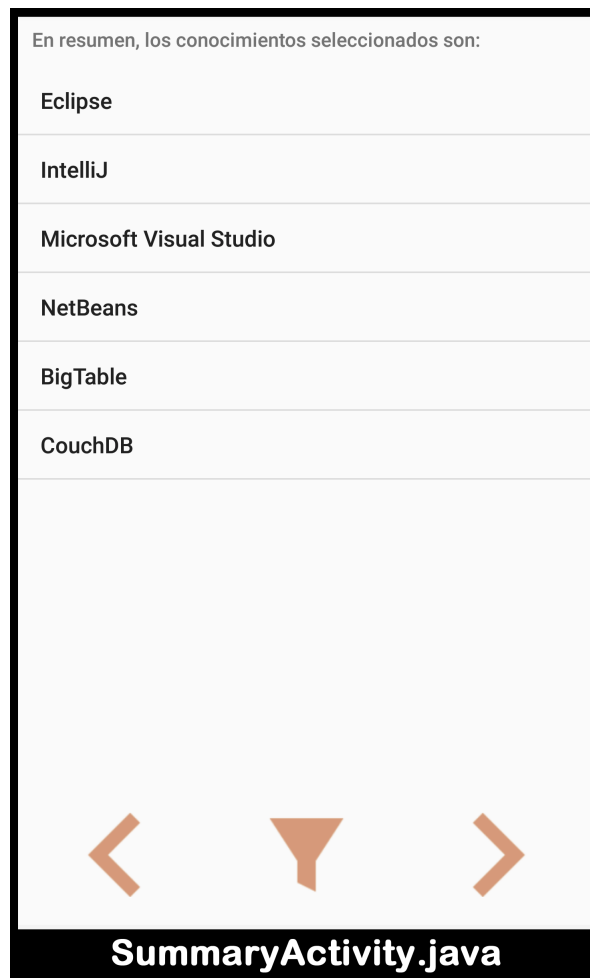


Fig. 17: Vista de resumen de selección

La figura 18 muestra el cuadro de diálogo que habilita la búsqueda de expertos según su país de origen y empresa en la que desempeñan un cargo.

La aplicación de este mecanismo de depuración trae consigo la agilización de la entrega de resultados por parte del servicio de consulta de perfiles de conocimiento, puesto que la comparación del perfil de necesidad del programador se efectúa sólo sobre una muestra reducida y puntual de perfiles candidatos, una vez que ya han sido filtrados.

Es tras concluir este punto que las funciones `get_conocimiento()` y `get_localizacion()` programadas dentro del SCPC son invocadas remotamente.



Fig. 18: Cuadro de diálogo de aplicación de filtro de localización

4.3.3. Dictamen y detalle de candidato seleccionado

En el extremo terminante de la aplicación y de la actividad del sistema en su totalidad, la figura 19 desvela la lista de expertos que podrían considerarse contribuidores potenciales, al contar con el conocimiento señalado en el perfil de necesidad ingresado en un principio.

En última instancia, el programador puede también inquirir sobre algunos detalles en alusión a los datos personales de los candidatos, como su información de contacto, disponibilidad y un resumen del *expertise* expresado en su perfil de conocimiento.

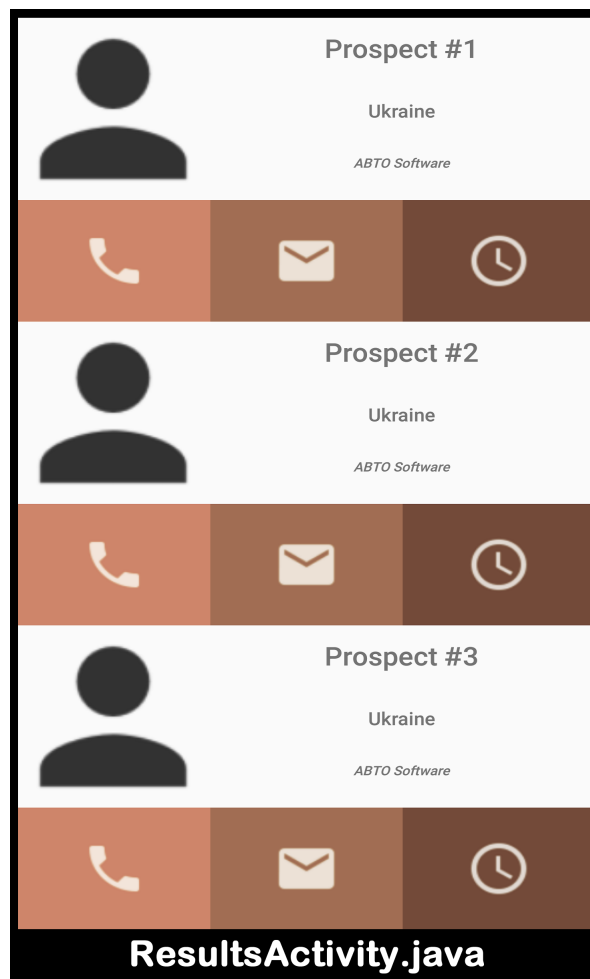


Fig. 19: Vista de candidatos disponibles

De esta forma concluye la ilustración de las interfaces del prototipo desarrollado, así como de las figuras correspondientes a las técnicas factuales que se utilizaron para su implementación.

Capítulo 5

Análisis de resultados

En representación del progreso obtenido de la investigación llevada a cabo anteriormente, así como de la transformación a la que la aplicación propuesta se ha visto sometida hacia el final de cada iteración de su ciclo de vida, el penúltimo capítulo de este manuscrito examina los resultados obtenidos a partir de una prueba piloto del prototipo desarrollado dentro de un entorno de demostración, en un intento por comprobar la utilidad objetiva de la herramienta.

Para dar pie al inicio de este análisis, se esboza un mapa conceptual que recibe el nombre de “esquema de integración de elementos”, el cual cohesiona todos los componentes del sistema, al mismo tiempo que advierte cuales son los datos de entrada y los datos de salida que estos precisan y producen, respectivamente, para lograr desenvolverse acorde al rol que, según lo estipulado durante las fases de diseño e implementación, les corresponde desempeñar.

Más adelante, tras haber dejado claro el recorrido de dicha integración, se ejemplifica a través del planteamiento de un escenario de uso viable, cuáles pueden ser las situaciones en las que la propuesta puede marcar una diferencia objetiva y derivar en un beneficio contundente para los programadores que laboran en entornos distribuidos, al mostrar el patrón de comportamiento de los actores del sistema en un despliegue hipotético.

Así pues, valiéndose del alcance de estos instrumentos, queda en entredicho el valor de los resultados recopilados, y se conduce al cierre del estudio planteado.

5.1. Integración de los elementos del sistema

A continuación se da seguimiento a las etapas de evolución por las que atraviesa la información gestionada por el sistema, al permutar esta entre sus distintos elementos desde el punto de partida hasta la consecución de los resultados.

La finalidad de este mapa de integración es exponer los principales eventos que toman lugar en tiempo de ejecución, enumerándolos acorde al orden en que estos son invocados.

Se puntualiza mediante este acto el uso de la ontología de conocimiento, tal como se reseña en la sección 4.1.1 del capítulo que antecede al actual, como una convención de términos presente tanto en la aplicación móvil como en la infraestructura web de la herramienta:

1. El cliente de Android solicita conectarse a la API web mediante un manejador de REST que hace las veces de intermediario entre cliente y servidor, tomando un objeto JSON como parámetro. Dicho objeto contiene el perfil de necesidad del programador, inferido de la lista de conocimientos que este ha seleccionado en la vista principal, y estructurado dentro de un arreglo de datos.
2. El manejador de REST canaliza la solicitud del usuario hacia el servicio de la API que este solicite, siendo en este caso el servicio de consulta de perfiles de conocimiento.
3. El SCPC recibe, interpreta y traduce la petición a lenguaje relacional, y da por comenzado el empate del perfil requerido contra los perfiles de conocimiento existentes dentro de la base de datos.
4. Concluido este proceso, la base de datos remite a aquellos perfiles de conocimiento que cumplen con las necesidades ingresadas.
5. Los datos son ahora convertidos por el SCPC de manera inversa desde un modelo relacional a un arreglo de JSON, cuyas entradas son recorridas y ordenadas acorde al grado de idoneidad que presentan los candidatos examinados, en función de las necesidades del usuario.
6. Se mapea el arreglo final dentro de un elemento del tipo *ListView* en el cliente de Android, dando por terminada la actividad del sistema.

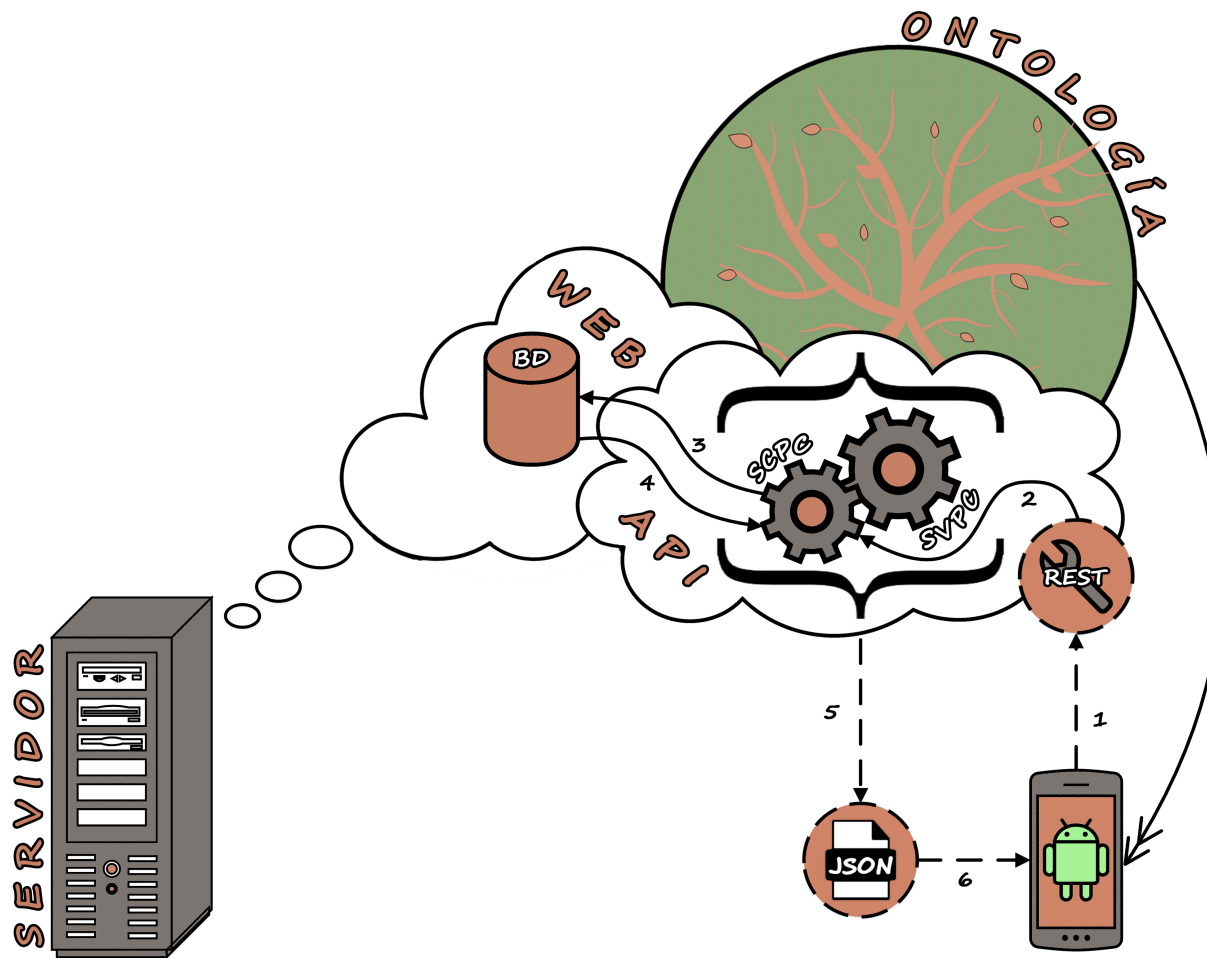


Fig. 20: Esquema de integración de elementos

5.2. Planteamiento de un escenario de uso factible

Antes de dar fin al análisis de resultados se ejemplarizan las posibilidades de la herramienta, mostrando su curso de acción dentro de una serie de imágenes en las que se señala, desde el plano hipotético, como intervendría la aplicación en los actos de interacción inaugurados en pos de la búsqueda de conocimiento entre las personas que laboran en entornos de desarrollo global de software.

Este escenario guarda cierta distancia respecto al dominio de la solución sugerida, aislando todo detalle técnico para volver a la perspectiva del negocio, exponiendo el uso del sistema desde un panorama estrechamente relacionado con la problemática de origen en su estado más puro.

- *Escenario A:* El programador Sebastián Franco se encuentra con un impedimento impuesto por una cuestión práctica que le impide finalizar uno o varios de sus deberes asignados. Dado que, incluso a nivel local, este trabajador desconoce a quién podría dirigir sus dudas, procurar asistencia remota empleando la aplicación móvil de apoyo en la identificación de colaboradores mediante perfiles de conocimiento, es su alternativa más lógica para afrontar este contratiempo.
- *Escenario B:* El programador descubre una lista de términos de entre los cuales debe seleccionar aquellos que se relacionen con las tareas que busca solventar. Posteriormente, una serie de interfaces lo guían hacia la especificación formal del perfil de conocimiento que espera encontrar, y el envío de una solicitud de asesoramiento a un experto cualificado.
- *Escenario C:* Sebastián Franco recibe un registro en el que figuran las personas instruidas en las áreas de conocimiento dentro de las que recae el problema manifestado, y que forman parte del capital humano de la empresa. El último movimiento por su parte es, pues, seleccionar a una de estas personas en cuestión. Dado que los datos de contacto disponibles se incluyen en el registro, elegir el canal a través del cual se transmitirá la misiva depende del programador y de los recursos que la empresa le provea, aclarando que el experto puede disponer del medio que considere conveniente para entregar su respuesta final.



Fig. 22: Escenario B

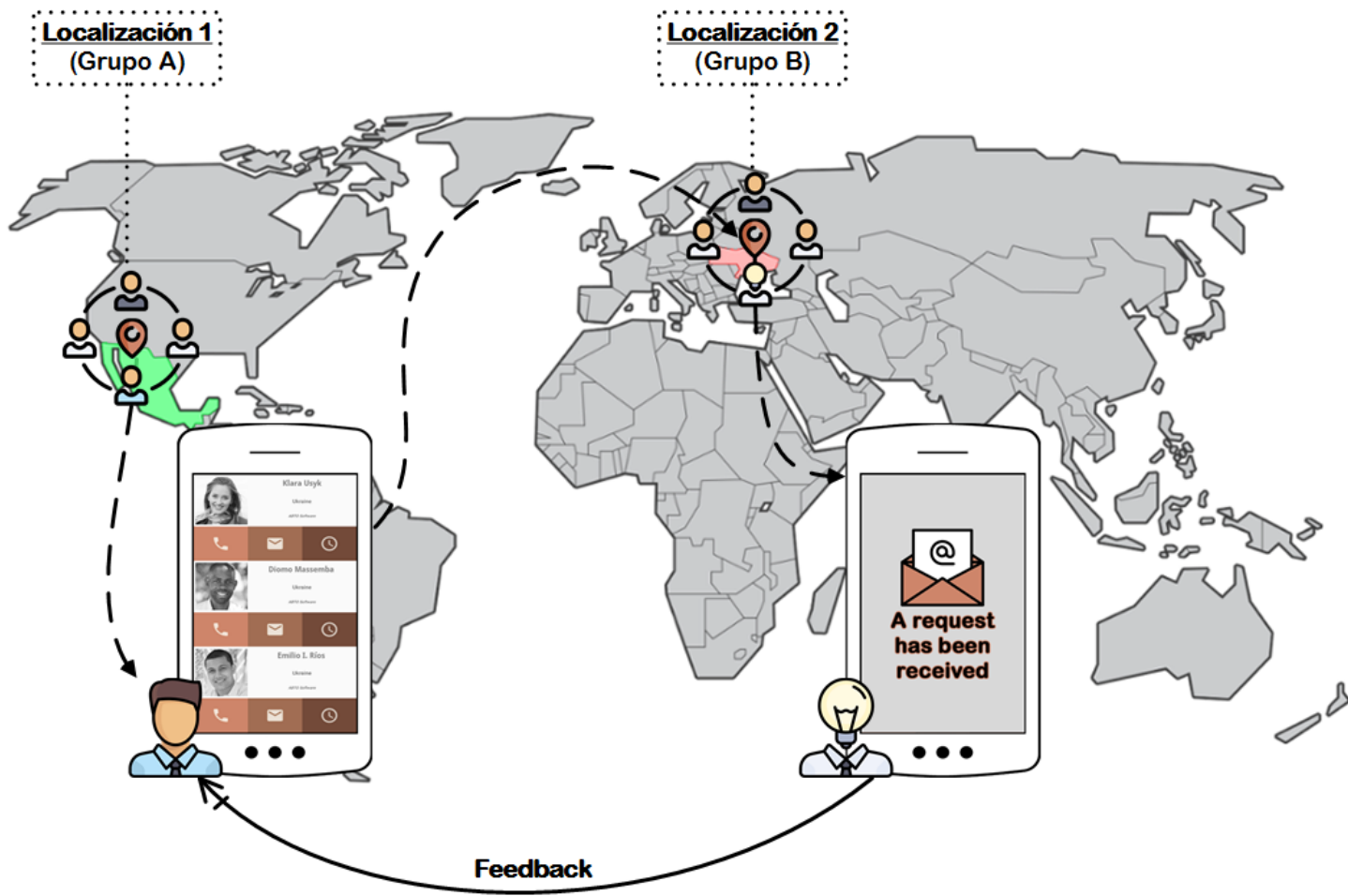


Fig. 23: Escenario C

Aunque es gracias al planteamiento de este escenario hipotético que se logra ejemplificar el uso de la aplicación en campo real, la aplicación presta lugar a una gran cantidad de permutaciones distintas en las que los antecedentes de fondo podrían variar de forma inconmensurable, produciendo en cada una de ellas un resultado, en potencia, similar al expuesto.

Capítulo 6

Conclusiones

El desarrollo global de software es un hito derivado de la globalización, el cual, sin ser estrictamente una novedad, todavía despierta una asombrosa cantidad de atención entre miles de empresarios y líderes de grupos de desarrollo, debido a las mejoras en materia de rendimiento laboral que se obtienen de él cuando se instaura propiamente dentro de un proyecto.

A pesar de que existe una amplia gama de herramientas para aminorar algunos de sus riesgos, diversas fuentes sugieren que, cuando se trata de señalar una solución ampliamente reconocida para manejar información crítica en ambientes de trabajo distribuidos, no existe unanimidad entre los trabajadores que laboran en este tipo de entornos.

Por lo consiguiente, destacan en el último capítulo de este documento las indagaciones finales que surgen alrededor de la herramienta que ha sido creada como método de subsidio para los costos que acarrearán los tiempos de interrupción a los que conduce la falta de conocimiento de los empleados, ante aquellas organizaciones cuyo núcleo laboral se distribuye entre distintas ubicaciones geográficas.

Asimismo, se recapitulan en las páginas subsecuentes las ventajas de esta utilidad, mostrando a la vez otras apreciaciones de expertos en el sector, junto con las posibles áreas de trabajo futuro que podrían complementar al producto de software actual, llegando en el acto al discernimiento conclusivo de sus aportes finales, gracias a los cuales se evidencia el logro de los objetivos

planteados en el primer capítulo del documento, y se hace posible reconocer con facilidad los beneficios que estos suponen para las empresas que hagan uso de la propuesta resaltada en el presente trabajo de tesis.

6.1. Apreciaciones complementarias

Pese a que el trabajo en equipo es intrínsecamente una actividad de carácter cooperativo, una vez que se asume la existencia de cierta distancia geográfica, por pequeña que sea, se hace ineludible que esta impacte negativamente en los valores de colaboración percibidos por parte de la empresa hacia sus equipos de trabajo. (Richardson et al., 2010)

Este patrón puede explicarse con facilidad si se toma en cuenta que en los entornos distribuidos, las relaciones entre los trabajadores son forjadas meramente por interacciones mediadas, referencias u opiniones heredadas de otros compañeros e indicadores que describen vagamente la reputación profesional de los individuos, pues las empresas suelen carecer de una infraestructura dentro de la cual puedan sobrevenir acercamientos confiables, que permitan a los equipos gozar de las características vitales de la comunicación por vía personal, como lo son la percepción de la buena voluntad y el sentido común. (Avram, 2008)

Se requiere, por ello, que las fuerzas motivacionales de los individuos den inicio, sostengan y lleven a término las diligencias necesarias para producir una contribución real, independientemente del sitio en el que radiquen. (Olivera et al., 2008)

Previo a esto, la promulgación de mecanismos que acondicionen los nuevos conocimientos, creados o adquiridos por la empresa, es determinante para su introducción en su memoria organizacional, de una manera tal que se logre maximizar su impacto y sus posibilidades de reutilización a largo plazo. (King et al., 2009)

Bajo esta jurisdicción, las tecnologías de información y la gestión de conocimiento se aglomeran para promover la distribución de la experiencia estructurada, ya que es bien sabido que el uso correcto de las TI potencia los procesos de creación, almacenamiento, acceso y transferencia del capital in-

telectual, tanto en contextos académicos como en contextos laborales y de negocios. (Ansari et al., 2011)

6.2. Contribuciones y aportaciones

La administración de perfiles de conocimiento resulta ser un activo de gran calibre para las empresas multi-nacionales, gracias al cual estas adquieren un medio para instar a sus trabajadores a compartir su experiencia y su saber con sus compañeros y, de ser necesario, a solicitar acceso al acervo profesional de los mismos.

Es en consideración a este hecho que, desde su concepción, la aplicación móvil de apoyo en la identificación de colaboradores mediante perfiles de conocimiento en grupos de desarrollo global de software acata tres reglas cuyo cumplimiento la hacen apta para auxiliar a los programadores que deben trabajar, en forma remota, con distintas personas ubicadas alrededor del mundo:

1. La aplicación debe ayudar a los programadores a formular consultas seleccionando tantos temas como encuentren preciso para resolver su problema.
2. La aplicación debe remitir a los programadores solamente hacia aquellos candidatos que cuenten con los perfiles más competentes, en el 100 % de los casos.
3. La aplicación debe permitir a los programadores escoger a quien contactarán, en el caso de que dos o más expertos aparezcan en la lista de contribuidores potenciales disponibles.

La consumación de estas reglas es avalada primeramente gracias al manejo de perfiles de conocimiento de personal, seguido por la condición pervasiva de la ontología de conocimiento en desarrollo y prueba del software definida con anterioridad, la cual cumple con la función fundamental de servir como connotación específica de qué conocimiento puede ser solicitado desde la aplicación móvil, qué expertos puede encontrar el usuario en la base de datos montada en el servidor, al hacer uso del servicio de consulta de perfiles de conocimiento, y dentro del dominio de ejecución de sistemas como el de (Rosas-Daniel,

2014), qué conocimientos pueden ser evaluados por el servicio de valoración de perfiles de conocimiento.

Es posible observar las implicaciones de la herramienta desde ángulos alternativos en cada una de estas publicaciones dedicadas a eventos académicos:

- *Propuesta de una Arquitectura para una Aplicación móvil de Apoyo en la Identificación de Colaboradores en Entornos de Desarrollo global de Software* (Córdova-Moras et al., 2016): trabajo presentado en el 6to. Simposio sobre Avances de Investigación en Ingeniería en el Estado de Sonora (AvIIES), año 2016.
- *Expert Location Tool for Global Software Development Environments based on Knowledge Profile Management* (Córdova-Moras et al., 2017): trabajo sometido en la quinta edición de la *International Conference on Software Engineering Research and Innovation* (CONISOFT), año 2017.
- *Hacia la Implementación de una Aplicación móvil de Apoyo en la Localización de Expertos dirigida a Equipos de Desarrolladores de Software distribuidos geográficamente*: trabajo por presentar en el 7mo. Simposio sobre Avances de Investigación en Ingeniería en el Estado de Sonora (AvIIES), año 2017.

6.2.1. Discernimiento final

Al permitir que el conocimiento perdure y se transmita fácilmente hacia aquellos individuos que manifiestan su interés por captarlo, la aplicación móvil de apoyo en la identificación de colaboradores mediante el perfil de conocimiento hace posible sobrellevar algunos de los mayores problemas de comunicación imputados a los entornos de trabajo distribuidos.

De esta forma, en atención directa a las interrogantes de la investigación y al planteamiento del problema descritos en la sección 1.2, la herramienta propuesta logra, de manera determinante, cumplir con el objetivo inicial perseguido por este proyecto, al consistir éste en facilitar a los programadores el acceso al expertise colectivo de la empresa para poder avanzar en sus tareas día con día, propósito que consigue gracias a los mecanismos de búsqueda y selección de perfiles de conocimiento de los cuales la aplicación hace uso.

A fin de cuentas, este software surge como una alternativa logística a soluciones más tradicionales, con el único fin de cerrar en cierta medida la brecha entre las fronteras del desarrollo de software para las personas que, dentro de medios no contiguos, se ven obligadas a lidiar con las adversidades inherentes a él.

6.3. Propuestas de trabajo futuro

Para dar fin al proceso de redacción de este manuscrito, se expone una lista de trabajos que corresponde a propuestas de terceros autores, desde las cuales es posible abstraer diversas ideas que, de ser implementadas en un futuro, supondrían para la herramienta un enriquecimiento considerable de su rentabilidad de cara a las empresas y al público objetivo:

- *Generación automática de perfiles* (Pietriková y Chodarev, 2016): Es posible inferir rasgos de conocimiento mediante el análisis estadístico de la frecuencia de elementos de lenguaje usados por un individuo sujeto a evaluación, o bien, a través de sistemas que se valgan de la evidencia mostrada en algún documento para extraer su perfil.
- *Uso de un lenguaje de dominio específico* (Cabrerero-Jojoa y Mariño-Drews, 2014): Existen diferentes lenguajes de descripción especialmente diseñados para definir ontologías. OWL (*Ontology Web Language*) destaca entre ellos como un lenguaje que describe clases, propiedades y relaciones conceptuales entre objetos, conformando un metamodelo, de tal manera que facilita la interoperabilidad y el procesamiento de contenido web.
- *Aplicación de una ontología modular* (Özacar et al., 2011): La modularización ontológica es un principio de la gestión de ontologías para lograr la reutilización, escalabilidad, mantenimiento y evolución de las mismas. Consiste en segmentar las ontologías en componentes reutilizables, al imponer la separación de sus restricciones y construir módulos encapsulados. Un enfoque modular permitiría actualizar sin esfuerzo los términos de conocimiento acordados, e incluso intercambiar entre diferentes ontologías, modificando la epistemología del sistema si se considerara necesario.

- *JTOWL como lenguaje de descripción semántica* (Yao et al., 2014): Procesar arreglos de JSON usando tecnologías de web semántica para convertirlos a ontologías OWL, puede resultar conveniente si se opta por implementar las dos opciones anteriores. JTOWL puede extender la semántica de datos utilizando razonamiento semántico y combinar múltiples recursos dentro de una ontología única. Entre estos datos se podrían incluir conceptos, propiedades, restricciones y valores que están implícitos en el objeto JSON.
- *Servicios web sensibles al contexto* (Gu, 2004): Los middleware sensibles al contexto son conscientes de su entorno, al cual se adaptan de manera inteligente según los cambios que se presenten en ellos, recurriendo a un proveedor de contexto. Si se decide emplear ontologías modulares, el uso de servicios sensibles al contexto permitiría intercalar entre varias de estas de manera automática.
- *Herramientas de análisis de perfiles de expertos en redes sociales empresariales* (Lin et al., 2009): Analizar información proveniente de redes sociales empresariales podría ser el futuro de la búsqueda de expertos, al acceder a una base de expertos mucho mayor, cuya participación explícita se limitaría a autorizar el uso de sus datos a escala mundial.

Sumado a todas estas posibles mejoras, el desarrollo del módulo tutor descrito de manera lacónica en la sección 3.3.1, mismo que fue descartado en una fase temprana del desarrollo del proyecto, se perfila como una opción a contemplar a posteridad, pues habilitaría convenientemente un nuevo canal de comunicación propio dentro del sistema.

Desde dicho módulo el experto seleccionado por el programador podría visualizar y atender las solicitudes recibidas, sin necesidad de salir de la aplicación, lo cual simplificaría el proceso de interacción entre usuarios al reducir la cantidad de pasos necesarios para concretar una nueva contribución a alguna actividad de desarrollo o prueba del proyecto.

De manera similar, se sugiere que en próximas versiones se incluya un factor de disponibilidad como nuevo parámetro para filtrar los resultados de la búsqueda, de modo que la responsabilidad de dejar en claro a los programadores qué expertos están disponibles al momento de su consulta, recaiga sobre el servicio de consulta de perfiles de conocimiento.

Empero, como sugerencia conclusiva, se subraya la integración de futuros mecanismos de software para obtener retroalimentación por parte de los usuarios y del personal experto, al ser esta de carácter indispensable para eludir la obsolescencia prematura de la herramienta, pues bien es sabido que las tecnologías de la información en la actualidad se transforman a un ritmo desmesuradamente veloz, y con esta transformación sobrevienen nuevos modelos, áreas de conocimiento, lenguajes y técnicas a emplear en el desarrollo de aplicaciones informáticas.

Bibliografía

- Ågerfalk, P. J., Fitzgerald, B., Holmström, H., Lings, B., Lundell, B. y Conchúir, E. Ó. (2005), ‘A Framework for Considering Opportunities and Threats in Distributed Software Development’, *Proc. International Workshop on Distributed Software Development, Paris, France: Austrian Computer Society* (August), 47–61.
- Ågerfalk, P. J., Fitzgerald, B., Olsson, H. H. y Ó Conchúir, E. (2008), ‘Benefits of Global Software Development: The Known and Unknown’, *Making Globally Distributed Software a Success Story* pp. 1–9.
- Alavi, M. y Leidner, D. E. (2001), ‘Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues’, *MIS Quarterly* **25**(1), 107–136.
- Ansari, M., Youshanlouei, H. R., Mood, M. M. y Jabarzadeh, Y. (2011), ‘Identifying and Ranking the Critical Success Factors in Implementation of Knowledge Management Using DELPHI Method, A Case Study of the Municipality of 22th district of Tehran’.
- Aranda, G. N. (2009), Marco para la elitización de requisitos software en procesos de desarrollo global, PhD thesis, Universidad de Castilla - La Mancha.
- Aranda, G. N., Vizcaíno, A., Palacio, R. R. y Morán, A. L. (2010), ‘What information would you like to know about your co-worker? A case study’, *Proceedings - 5th International Conference on Global Software Engineering, ICGSE 2010* pp. 135–144.
- Aspray, W., Mayadas, F. y Vardi, M. Y. (2006), Globalization and Offshoring of Software, Technical Report 3, Association for Computing Machinery.
- Avram, G. (2008), ‘Knowledge Work Practices in Global Software Deve-

- lopment’, *The Electronic Journal of Knowledge Management* **5 Issue 4(4)**, 347–356.
- Bray, E. T. (2014), The JavaScript Object Notation (JSON) Data Interchange Format Abstract, Technical report, Google, Inc.
- Cabrero-Jojoa, C. H. y Mariño-Drews, O. (2014), ‘Domain Specific Language for Handling Modular Ontologies’, *2014 9th Computing Colombian Conference, 9CCC 2014* pp. 48–53.
- Cao, Y., Liu, J., Bao, S., Li, H. y Craswell, N. (2006), ‘A Two-Stage Model for Expert Search’, *Information Retrieval* pp. 1–16.
- Charland, A. y Leroux, B. (2011), ‘Mobile Application Development: Web vs. Native’, *Communications of the ACM* **54**, 0–5.
- Cho, J. (2007), ‘Globalization and global software development’, *Issues in Information Systems* **VIII(2)**, 287–290.
- Chokkattu, J. (2015), ‘Windows Phone Market Share Drops to 1.7 percent, as Android and Huawei’s Fortunes Rise’.
- Constantinou, A. (2015), ‘State of the Developer Nation Q1 2015’.
- Córdova-Moras, L. G., Rodríguez-Elías, Ó. M., Palacio-Cinco, R. R., Serna-Encinas, M. T. y Velázquez-Mendoza, M. d. J. (2016), Propuesta de una Arquitectura para una Aplicación móvil de Apoyo en la Identificación de Colaboradores en Entornos de Desarrollo global de Software, in ‘Avances de Investigación en Ingeniería en el Estado de Sonora’, Vol. 6, Hermosillo, México.
- Córdova-Moras, L. G., Rodríguez-Elías, Ó. M., Serna-Encinas, M. T. y Palacio-Cinco, R. R. (2017), Expert Location Tool for Global Software Development Environments based on Knowledge Profile Management.
- Covili, J. R., Ochooa, S. F., Pinno, J. A., Favela, J., Mejia, D. y Morán, A. L. (2009), ‘Designing Mobile Shared Workspaces by Instantiation’, *Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2009* pp. 402–407.
- Dalkir, K. (2001), ‘Introduction to Knowledge Management’, pp. 1–30.

- Dalkir, K. (2005), *Knowledge Management in Theory and Practice*, Vol. 4, Elsevier.
- DeFranco-Tommarello, J. y Deek, F. P. (2002), ‘Collaborative software development: A discussion of problem solving models and groupware technologies’, *Proceedings of the Annual Hawaii International Conference on System Sciences* **2002-Janua(c)**, 568–577.
- Dochy, F. y Valcke, M. (1992), Validation of Knowledge Profile Dimensions: Looking for Empirical Evidence, Technical report, Open University, Herleen.
- Dossani, R. y Kenney, M. (2008), Implications of Globalization for Software Engineering, Technical report, National Academy of Sciences.
- Faßbinder, P. y Henz, V. (2009), ‘Improving Global System Development and Collaboration across Functions Experiences from Industry’, *Proceedings - 2009 4th IEEE International Conference on Global Software Engineering, ICGSE 2009* pp. 262–266.
- Fielding, R. T. y Taylor, R. N. (2002), ‘Principled Design of the Modern Web Architecture’, *ACM Transactions on Internet Technology* **2(2)**, 407–416.
- Fryer, K. y Gothe, M. (2008), ‘Global software development and delivery: Trends and challenges’.
- Galvina, Z. y Smite, D. (2011), Software Development Processes in Globally Distributed Environment, *in* ‘Computer Science and Information Technologies’, Vol. 770, University of Latvia, pp. 7–14.
- GlobalStats (2016), ‘Top Mobile Operating Systems Per Country’.
- Google Developers (2016), ‘Your First Progressive Web App’.
- Griswold, W. G. (2007), ‘Five Enablers for Mobile 2.0’, *Computer* **40(10)**, 96–98.
- Gu, T. (2004), ‘A Middleware for Building Context-Aware Mobile Services’, *Conference, 2004. VTC 2004-Spring. 2004* **5**, 2656–2660.
- Hintze, B. (2009), Bungee Connect and Model-View-Adapter (MVA): The architectural pattern for next generation interactive cloud-based web applications, Technical report, BungeeLabs.

- International Organization for Standardization (2000), ISO/IEC 9126-1:2000, Technical report, American National Standards Institute.
- Jakl, M. (2005), REST: Representational State Transfer, Technical report, Technische Universität Wien, Viena.
- Jelínek, P. (2012), Risk Management in Software Development Projects, PhD thesis.
- King, W., Chung, T. y Haney, M. (2009), ‘Knowledge Management and Organizational Learning’, *Omega* **4**(2), 167–172.
- Lanubile, F., Ebert, C., Prikładnicki, R. y Vizcaíno, A. (2010), ‘Collaboration Tools for Global Software Engineering’, *IEEE Software* **27**(2), 52–55.
- Lin, C. Y., Cao, N., Liu, S. X., Papadimitriou, S., Sun, J. y Yan, X. (2009), ‘SmallBlue: Social Network Analysis for Expertise Search and Collective Intelligence’, *Proceedings - International Conference on Data Engineering* pp. 1483–1486.
- Llamas, R., Chau, M. y Shirer, M. (2015), ‘Android and iOS Squeeze the Competition, Swelling to 96.3 % of the Smartphone Operating System Market’.
- Macdonald, C. y Ounis, I. (2006), ‘A Belief Network Model for Expert Search’, *Computing* .
- Macdonald, C. y Ounis, I. (2008), ‘Expert Search Evaluation by Supporting Documents’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **4956 LNCS**, 555–563.
- Masner, J., Šimek, P., Jarolínek, J. y Hrbek, I. (2015), ‘Mobile Applications for Agricultural Online Portals – Cross-platform or Native Development’, *Agris On-line Papers in Economics and Informatics* **7**(2), 47–54.
- Massé, M. (2012), *REST API Design Rule Book: Designing Consistent RESTful Web Service Interfaces*, O’Reilly, Sebastopol, CA.
- Merino, M. (2014), ‘¿Qué es una API y para qué sirve?’.
- Mohagheghi, P. (2004), ‘Global Software Development: Issues, Solutions, Challenges’.

- Montané Jiménez, L. G., Benítez Guerrero, E. I., Mezura Godoy, M. D. C. y Martínez López, E. (2015), ‘Studying Social Interactions in Groupware Systems’, *IEEE Latin America Transactions* **13**(10), 3488–3497.
- Mrak, M. (2000), *Globalization: Trends, Challenges and Opportunities for Countries in Transition*, Technical report, United Nations Industrial Development Organization, Vienna.
- Mudambi, R. y Venzin, M. (2010), ‘The Strategic Nexus of Offshoring and Outsourcing Decisions’, *Journal of Management Studies* **47**(8), 1510–1533.
- Mullally, A., Mckelvey, N. y Curran, K. (2011), ‘Performance Comparison of Enterprise Applications on Mobile Operating Systems’, *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)* **9**(3).
- Mullan, J. E., Kenney, M. F. y Dossani, R. (2008), ‘Mexico and the Globalization of Services: Outflanked Once Again?’, *Economía Mexicana, Nueva Época* **17**(2), 171–202.
- Naím, M. (2009), ‘Think Again: Globalization’, *Foreign Policy* (171), 28–34.
- Nidhra, S., Yanamadala, M., Afzal, W. y Torkar, R. (2012), ‘Knowledge transfer challenges and mitigation strategies in global software development - A systematic literature review and industrial validation’, *International Journal of Information Management* **33**(2), 333–355.
- Nieves-Lahaba, Y. y León-Santos, M. (2001), ‘La gestión del conocimiento: una nueva perspectiva en la gerencia de las organizaciones’, **31**(1), 3–4.
- Nishida, H. y Nguyen, T. (2013), ‘Optimal Client-Server Assignment for Internet Distributed Systems’, *IEEE Transactions on Parallel and Distributed Systems* **24**(3), 565–575.
- Nonaka, I. (1995), ‘The Knowledge-Creating company’, *Harvard Business Review* (August).
- Olivera, F., Goodman, P. S. y Tan, S. (2008), ‘Contribution Behaviors in Distributed Environments’, *MIS Quarterly* **32**(1), 23–42.
- Oñate-Gómez, C., Barrero-Ripoll, M. y Sierra-Gómez, F. (2009), ‘Interdisciplinarity and the Prior Knowledge Profile Examination La interdisciplina-

- riedad y la prueba de perfil de conocimientos previos', *Revista Electrónica de Investigación Educativa* **11**, 1–16.
- Open Handset Alliance (2007), 'Industry Leaders Announce Open Platform for Mobile Devices'.
- Oyefolahan, I. O. y Dominic, P. D. D. (2010), Conceptualization of the Antecedents and Impacts of KMS Utilization: A Preliminary Framework, *in* 'International Symposium on Information Technology - System Development and Application and Knowledge Society', Vol. 3, IEEE, pp. 1485–1489.
- Özacar, T., Öztürk, Ö. y Ünalir, M. O. (2011), 'ANEMONE: An environment for modular ontology development', *Data and Knowledge Engineering* **70**(6), 504–526.
- Palacio, R. R., Morán, A. L., González, V. M. y Vizcaíno, A. (2009), 'Providing Support for Starting Collaboration in Distributed Software Development: A Multi-Agent Approach', *2009 WRI World Congress on Computer Science and Information Engineering, CSIE 2009* **7**, 397–401.
- Palacio, R. R., Vizcaíno, A., Morán, A. L. y González, V. M. (2011), 'Tool to facilitate appropriate interaction in global software development', *IET Software* **5**(2), 157.
- Pardo-Kuklinski, H., Brandt, J. y Puerta, J. P. (2010), 'Mobile Web 2.0: La nueva industria de la comunicación móvil', *Quaderns del CAC* pp. 59–66.
- Pataki, G. E., Dilllon, J. T. y McCormack, M. (2003), Section III:6 System Implementation, *in* 'NYS Project Management Guidebook', release 2 edn, New York State Office for Technology, Albany, pp. 177–192.
- Pietriková, E. y Chodarev, S. (2016), 'Towards Programmer Knowledge Profile Generation', *Acta Electrotechnica et Informatica* **16**(1), 15–19.
- Pietzuch, P. R. y Bacon, J. M. (2002), 'Hermes: A distributed event-based middleware architecture', *Proceedings - International Conference on Distributed Computing Systems* **2002-Janua**, 611–618.
- Popa, B. (2017), 'Google's Android Overtakes Windows as the World's Number 1 Operating System'.

- Portillo-Rodríguez, J., Vizcaíno, A., Piattini, M. y Beecham, S. (2012), ‘Tools used in Global Software Engineering: A systematic mapping review’, *Information and Software Technology* **54**(7), 663–685.
- Pretel, I. y Lago, A. B. (2014), Evaluación remota de aplicaciones móviles híbridas: nueva aproximación en entornos reales, Technical report, Deusto Institute of Technology, Bilbao.
- Prikladnicki, R., Audy, J. L. N. y Evaristo, R. (2003), ‘Global Software Development in Practice Lessons Learned’, *Software Process: Improvement and Practice* **8**(4), 267–281.
- Pujol, J. M. y Sangüesa, R. (2001), NETEXPERT: Sistema basado en inteligencia artificial para la localización de expertos, Technical report, Escola Tècnica Superior d’Enginyers de Telecomunicació de Barcelona.
- Ranger, S. (2015), ‘iOS versus Android. Apple App Store versus Google Play: Here comes the next battle in the app wars’.
- Richardson, I. (2007), ‘Overview of Global Software Development, Barriers, Complexities and Project Management’.
- Richardson, I., Casey, V., Burton, J. y McCaffery, F. (2010), Global Software Engineering: A Software Process Approach, in I. Mistrik, J. Grundy, A. Hoek y J. Whitehead, eds, ‘Collaborative Software Engineering’, Springer, chapter 2, pp. 237–239.
- Richardson, I., Casey, V., McCaffery, F., Burton, J. y Beecham, S. (2012), ‘A Process Framework for Global Software Engineering Teams’, *Information and Software Technology* **54**(11), 1175–1191.
- Richter, F. (2017), ‘Chart: The Smartphone Platform War Is Over’.
- Rodríguez-Elías, O. M., Velázquez-Mendoza, M. d. J. y Rose-Gómez, C. E. (2017), An Ontology Based System for Knowledge Profile Management, in G. Alor-Hernández y R. Valencia-García, eds, ‘Current Trends on Knowledge-Based Systems’, Springer, chapter 3, pp. 49–72.
- Rosas-Daniel, J. A. (2014), Construcción de un modelo de lógica difusa para validación de perfiles de conocimiento de personal, Master’s thesis, Instituto Tecnológico de Hermosillo.

- Rosas-Daniel, J. A., Rodríguez-Elías, O. M., Velázquez-Mendoza, M. d. J. y Rose-Gómez, C. E. (2015), Construcción de un Modelo de Lógica Difusa para Validación de Perfiles de Conocimiento de Personal, *in* 'Avances de Investigación en Ingeniería en el Estado de Sonora', Vol. 5.
- Rouse, M. y Steele, C. (2011), 'Mobile operating system'.
- Salesforce Marketing Cloud (2014), 'Mobile Behavior Report', pp. 11–12.
- Schneider, S., Torkar, R. y Gorschek, T. (2013), 'Solutions in global software engineering: A systematic literature review', *International Journal of Information Management* **33**(1), 119–132.
- Scholte, J. A. (2008), 'Defining Globalisation', *The World Economy* **31**(11), 1471–1502.
- Senge, P. M. (1995), *The Fifth Discipline: The Art & Practice of the Learning Organization*.
- Senge, P. M., Cambron-McCabe, N., Lucas, T., Smitha, B., Dutton, J. y Kleiner, A. (2002), *Schools That Learn: A Fifth Discipline Fieldbook for Educators, Parents, and Everyone Who Cares about Education*.
- Serrano, N., Hernantes, J. y Gallardo, G. (2013), 'Mobile Web Apps', *IEEE Software* **30**(5), 22–27.
- Sherif, K., Hoffman, J. y Thomas, B. (2006), 'Can technology build organizational social capital? The case of a global IT consulting firm', *Information and Management* **43**(7), 795–804.
- Sin, D., Lawson, E. y Kannoorpatti, K. (2012), 'Mobile web apps - The non-programmer's alternative to native applications', *5th International Conference on Human System Interaction* pp. 8–15.
- Sørensen, E. y Mihailesc, M. I. (2010), 'Model-View-ViewModel (MVVM) Design Pattern using Windows Presentation Foundation (WPF) Technology', *MegaByte Journal* **9**(4), 1–19.
- Spycher, N. (2012), RESTful Web API für YouReputation: Entwicklung einer RESTful Web API mit PHP, Bachelorarbeit, Universität Freiburg.
- Staab, S. y Brewster, C. (2004), 'Knowledge Representation with Ontologies: The Present and Future', *IEEE Intelligent Systems* pp. 72–81.

- Tracy, K. W. (2012), ‘Mobile application development experiences on Apple’s iOS and Android OS’, *IEEE Potentials* **31**(4), 30–34.
- Tramullas, J., Garrido-Picazo, P. y Sánchez-Casabón, A.-I. (2011), ‘Groupware y software social: propuesta de marco de evaluación analítica para herramientas de software libre’, *El profesional de la información* **20**(4), 465–473.
- Uriarte, F. A. (2010), *Introduction to Knowledge Management*, ASEAN Foundation, Jakarta.
- Vaidyanathan, R., Kim, G. T., Kolarov, A., Caruso, F. y Forbes, G. (2011), ‘A Novel Cross-layer Modeling Framework for Analyzing SOA-based Information Services’, *Proceedings - IEEE Military Communications Conference MILCOM* pp. 2019–2024.
- Velázquez-Mendoza, M. d. J. (2013), Construcción de un modelo para el diseño de perfiles de conocimiento, Master’s thesis, Instituto Tecnológico de Hermosillo.
- Vizcaíno, A., García, F., Piattini, M. y Beecham, S. (2016), ‘A validated ontology for global software development’, *Computer Standards and Interfaces* **46**, 66–78.
- Yao, Y., Wu, R. y Liu, H. (2014), JTOWL: A JSON to OWL Converter, in ‘Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning - Web-KR ’14’, ACM Press, New York, New York, USA, pp. 13–14.
- Yoo, Y. (1998), Predicting Groupware Usage, in ‘System Sciences, 1998., Proceedings of the 31st Annual Hawaii International Conference on System Sciences’, IEEE, Kohala Coast, pp. 10–17.
- Zamudio, S. A., Santaolaya, R. y Fragoso, O. G. (2012), ‘Restructuring Object-Oriented Frameworks into Model-View-Adapter Architecture’, *IEEE Latin America Transactions* **10**(4), 2010–2016.
- Zhu, Y., Wu, W. y Li, D. (2016), ‘Efficient Client Assignment for Client-Server Systems’, *IEEE Transactions on Network and Service Management* **13**(4), 835–847.