

Transferencia de conocimiento en equipos de desarrollo de software – Mapeo Sistemático

Fernando Ibarra Torres, Mario Matías Urbietta, Gustavo Rossi, Alejandra Beatriz Lliteras

Abstract

The acquisition of new knowledge, by means of teaching programming languages, tools, methodologies, and development processes, is a need in the software industry to retain the experience of the programmer within the organization. This work aims analysing existing research on techniques, tools, approaches and experiences applied to the transmission and acquisition of new knowledge. Additionally, to analyse the learning process within an organizational environment, and how a company captures, maintains and transmits the knowledge. We conducted a survey to identify the software practitioner needs regarding experience and knowledge transfer. Once we could realize there is a challenge to manage the knowledge, a systematic mapping of the literature is carried out. We have considered works published between 2007 and 2019. 51 articles were selected, 71% are studies in industry, 29% are focused on university education. Our findings reveal the great challenge involved in studying the acquisition and exchange of new knowledge. Additionally, we were supported with an initial survey to validate results. Much of the literature focuses on analysing academic processes, more research is needed on the process of learning new knowledge with respect to programming languages, tools, methodologies, cycles and development processes in the software industry.

Index Terms: knowledge transfer, software development groups, systematic mapping, junior developer, new knowledge.

I. INTRODUCCIÓN

El continuo avance tecnológico y la globalización de los mercados están generando escenarios que impulsan a las industrias a la búsqueda de disminuir sus costes laborales [1] y a que adopte medidas

eficientes para la producción y distribución del conocimiento, lo que está conllevando a la reestructuración organizacional [2], esta situación obliga a investigar nuevos modelos en los que la importancia de los procesos industriales es reemplazada por la relevancia de la transmisión y adquisición de conocimientos entre empleados [3], convirtiéndose esto en punto importante de la nueva era económica [4].

Tradicionalmente según Panwong [5], la enseñanza y el aprendizaje en el ámbito educativo formal está centrada en el profesor como método básico. De esta manera los estudiantes solo aprenden el contenido de lo que el docente transpone. Por otro lado, en el ámbito laboral el papel del docente desaparece y en general, son los desarrolladores experimentados quienes deben ser la fuente del conocimiento hacia los noveles [6].

Se han comenzado a generar enfoques para hacer frente a lo expuesto, por ejemplo el desarrollo colaborativo, la programación por pares, el hackathon (reuniones entre desarrolladores para compartir experiencias y habilidades) [7], y la inmersión del aprendizaje con el uso de entornos virtuales, móviles y cursos abiertos masivos online pero una implementación de estos tipos involucra tiempo y esfuerzo [8], [9].

Actualmente la industria de software se destaca por la constante demanda de recursos humanos calificados y su alta rotación laboral implicando contextos laborales inestable en donde la transferencia de conocimiento entre pares es de suma relevancia. Los desarrolladores de software novatos son quienes más inconvenientes presentan al aprender un nuevo lenguaje de programación [10], [11], por la complejidad de aspectos técnicos, documentación mal construida y la falta de gestión en la variedad de información [12], [20], [22].

Las organizaciones se ven obligadas a buscar la manera de mantener y convertir la información generada [13], de tal forma que permita a los nuevos desarrolladores integrarse al proceso laboral y aprender [14].

Actualmente, el proceso para desarrollar software involucra una gran cantidad de decisiones, basadas en el conocimiento personal o en la experiencia [15], [16], la industria del software debe renovarse, coordinarse e investigar constantemente para mejorar estas decisiones y encontrar nuevas soluciones para reducir las curvas de aprendizaje. El conocimiento individual tiene que ser, recolectado, organizado, almacenado, compartido y aprovechado a nivel de proyecto y organización [17]. Es una tarea compleja y un reto, en vista de la variedad de conocimientos adquiridos por diferentes canales o medios [18].

Nuestro estudio se enfoca en los nuevos conocimientos del desarrollo de software que se necesita intercambiar entre

Oscar Fernando Ibarra, Universidad Técnica de Ambato, Ambato, Ecuador (email: oscar.ibarrat@info.unlp.edu.ar).

Mario Matías Urbietta, UNLP, Facultad de Informática, LIFIA, La Plata, Argentina also CONICET (email: matias.urbietta@lifia.info.unlp.edu.ar).

Gustavo Rossi, UNLP, Facultad de Informática, LIFIA, La Plata, Argentina also CONICET (email: gustavo@lifia.info.unlp.edu.ar).

Alejandra Beatriz Lliteras, UNLP, Facultad de Informática, LIFIA, La Plata, Argentina also CICPBA (email: lliteras@lifia.info.unlp.edu.ar).

miembros de los equipos de desarrollo, específicamente nuevos conocimientos en: lenguaje de programación, herramientas, framework, metodología, ciclos y procesos del desarrollo.

Complementando lo anterior, Gossen y Sena [19] [20] señalan que los mecanismos para la transmisión de conocimiento tienen que tender a ser dinámicos y evolutivos con el tiempo sin depender de la estructura organizacional sea esta informal u orgánica, el aprendizaje es considerado como uno de los nuevos paradigmas educativos y sociales, donde en una organización la situación de aprendizaje puede ocurrir en cualquier momento y lugar [21].

Estudios preliminares como el de Reyna [17] se enfocan en ámbitos educativos como la programación por pares y no en la industria del software. El presente estudio busca obtener una visión global de la investigación existente, analizar resultados y encontrar vacíos en la investigación.

La transmisión de nuevos conocimientos en lenguajes de programación, herramientas, metodologías, ciclos y procesos de desarrollo es una arista poco analizada, tal como también lo señalan en su artículo [22], con lo que el trabajo a desarrollar requerirá de grandes esfuerzos para buscar responder a la pregunta de investigación ¿Cómo hacer para renovar o adquirir nuevos conocimientos en equipos de desarrollo de software?.

En este trabajo presentamos una revisión del estado del arte sobre la adquisición de nuevos conocimientos desde el punto de vista de la siguiente pregunta de investigación: “¿Cómo hacer para adquirir nuevos conocimientos en lenguajes de programación, herramientas, metodologías, ciclos y procesos de desarrollo en equipos de desarrollo de software?”. Para ello se ha definido una cadena de búsqueda que brinda resultados con diferentes fuentes bibliográficas.

El artículo está organizado de la siguiente manera: la Sección 2 presenta resultados de una Encuesta realizada en la Industria de Software, la Sección 3 presenta el método de investigación, la Sección 4 presenta resultados del estudio, la Sección 5 presenta resultados de los métodos de transmisión de nuevos conocimientos, la Sección 6 presenta una discusión general, y finalmente, la Sección 7 presenta conclusiones.

II. ENCUESTA EN LA INDUSTRIA

En esta sección mostramos una encuesta realizada a 89 desarrolladores de software de diferente puesto laboral, lugar de trabajo y edad (todos mayores de edad).

El objetivo de esta encuesta es explorar la problemática planteada en el marco de la industria de software.

Como resultado, se obtuvo que el 41% de encuestados señalan que en la industria del software los niveles heterogéneos de conocimiento son un problema de grado alto y cerca del 53 % lo identifican como un problema medio,

1. ¿Los niveles heterogéneos de conocimiento con que grado son un problema en su organización?
89 respuestas

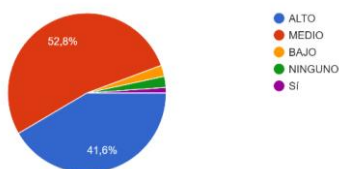


Fig. 1. Grado de afectación a causa de los niveles heterogéneos de conocimiento

4. ¿Mientras desarrolla software, que tan complejo considera usted que es adquirir un nuevo conocimiento entre desarrolladores?
89 respuestas

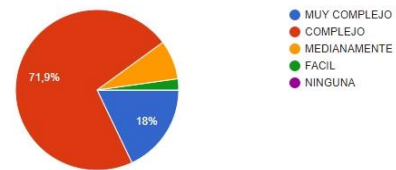


Fig. 2. Gráfico con porcentaje de dificultad para adquirir nuevos conocimientos

llegando a la conclusión que la heterogeneidad en el conocimiento es un inconveniente en las organizaciones (ver Fig. 1).

Además, identificamos que cerca del 90% de desarrolladores encuestados considera entre muy complejo y complejo la adquisición de nuevos conocimientos entre pares cuando desarrolla software, debido a la gran cantidad de información y a los diversos canales que se utilizan para llevar a cabo este proceso (ver Fig. 2).

La presente encuesta realizada a desarrolladores en actividad en la Industria, es considerado una primera herramienta de validación cuantitativa de la hipótesis presentada, más información de la encuesta se puede encontrar en el siguiente link <https://github.com/feriba/transferKnowledge.git>.

III. MÉTODO DE INVESTIGACIÓN

A. Preguntas de investigación.

El objetivo de nuestro estudio es examinar el estado del arte en el intercambio de conocimientos entre pares de un equipo de desarrollo en la industria del software intentando responder la siguiente pregunta de investigación: ¿Cómo hacer para renovar o adquirir nuevos conocimientos en equipos de desarrollo de software? Esto nos permitirá estudiar cómo es la transmisión de conocimientos y dado que nuestra pregunta de investigación es amplia la hemos descompuesto en 5 sub preguntas y que definimos textualmente a continuación y su motivación:

RQ1: ¿Con qué técnicas o métodos se adquieren nuevos conocimientos en lenguajes de programación, herramientas, metodologías, ciclos de desarrollo y procesos de desarrollo?

Se pretende obtener una visión general de cómo es el proceso de enseñanza y aprendizaje de nuevos conocimientos y el impacto que causa a nivel individual, grupal y organizacional, es decir hallar los beneficios o afectaciones de este proceso.

En este un punto es necesario conocer los modos que se utilizan para de adquirir conocimientos e ir comprendiendo que pueden existir diferentes vías de transmisión y que información se puede ir compartiendo en cada una.

RQ2: ¿Es posible adquirir conocimientos nuevos de lenguajes de programación, herramientas, metodologías, ciclos y procesos de desarrollo de otros equipos o grupos de desarrolladores de software?

Analizar la factibilidad de adquirir nuevos conocimientos de otro u otros desarrolladores y determinar que conocimiento es el transmitido. Adicionalmente identificar que habilidades y medios de comunicación son los mayormente utilizados en este proceso.

Identificando las cualidades o características de los grupos

de desarrolladores que pueden o no estar habilitados para la transmisión de conocimientos y señalando que tipo de información es la que se requiere y puede transferir.

RQ3: ¿Se puede compartir completamente el nuevo conocimiento en lenguajes de programación, herramientas, metodologías, ciclos de desarrollo y procesos de desarrollo entre desarrolladores o solo parcialmente?

Conocer si el nuevo conocimiento se puede compartir de manera completa o por fases y analizar el grado de dificultad que conlleva este proceso. Además, determinar si con la cantidad del nuevo conocimiento transmitido hasta un cierto punto se puede llegar a la resolución de problemas.

Adicionalmente, estudiar cómo influye la carencia de correctos canales de comunicación entre miembros de los equipos de desarrollo y el proceso de compartir nuevo conocimiento.

RQ4: ¿En qué etapa o fase del desarrollo es la más adecuada para transmitir el nuevo conocimiento de lenguajes de programación, herramientas, metodologías, ciclos de desarrollo y procesos de desarrollo a otro programador?

Identificar cual es la fase idónea para transmitir nuevo conocimiento entre desarrolladores y a su vez analizar las posibles consecuencias de una incorrecta transmisión en un momento inadecuado.

Además, con nuestro estudio podremos indagar las diferentes fases del proceso de desarrollo y el conocimiento que se va generando, para poder determinar la factibilidad o no de transmitirlo a otros equipos de desarrolladores.

RQ5: ¿Es un problema para las empresas la falta de conocimientos en tecnologías y conceptos?

Determinar si efectivamente es un problema para las compañías de software y conocer cuáles son las posibles afectaciones originadas por la desactualización en conocimientos informáticos. Complementariamente con nuestro estudio identificaremos los conflictos que evitan una correcta capacitación en temas y conceptos de tecnología dentro de una organización de desarrollo de software y como esto afecta al desempeño de los equipos de desarrollo y a la consecución de los objetivos institucionales.

TABLA I
BASES DE DATOS ELECTRÓNICAS PARA BÚSQUEDA

N.	BD electrónica	Seleccionada
DB1	Scopus	Si
DB2	ACM Digital Library	Si
DB3	Science Direct	Si
DB4	SpringerLink	Si
DB5	Inspec	No
DB6	ISI Web of Science	No

TABLA II
CADENA DE BÚSQUEDA

Concepto	Términos		
Desarrollador de software	de	(software AND developer)	AND
Novato, inexperto, nuevo		(new OR inexperted OR novice OR junior)	AND
Aprender		(learning)	AND
Lenguaje de programación, estrategia de desarrollo, herramienta de desarrollo, metodologías, ciclos y procesos del desarrollo, framework.	de	(programming language OR development approach OR development tool OR methodologies OR development cycles OR framework OR development process)	OR

B. Mapeo sistemático

Aunque se reportan estudios sobre el aprendizaje de conocimientos y gestión del conocimiento para el área del software. Son escasos los estudios de mapeo sistemático que se enfoquen en el nuevo conocimiento y su enseñanza o formas de aprendizaje [23].

El presente estudio de mapeo sistemático se realizó en 4 etapas: Planificación, Realización, Revisión y Filtración.

C. Ejecución del estudio de mapeo.

Con el fin de obtener una visión general de la investigación sobre la transmisión de nuevos conocimientos de lenguajes de programación, herramientas, metodologías, ciclos de desarrollo y procesos de desarrollo en el desarrollo de software, se realizan las siguientes actividades como parte del mapeo sistemático:

- Definición de la pregunta de investigación: ¿Cómo hacer para renovar o adquirir nuevos conocimientos de lenguajes de programación, herramientas, metodologías, ciclos de desarrollo y procesos de desarrollo en equipos de desarrollo de software?

- Señalamiento de la estrategia de búsqueda.

- Búsqueda de publicaciones, artículos o conferencias relevantes sobre el tema en SCOPUS para obtener resultados preliminares.

D. Búsqueda de estudios.

En nuestra investigación combinamos el método de búsqueda automática y manual para recuperar los estudios relevantes para nuestro mapeo en la fase de búsqueda. Recuperamos estudios en la base de datos de SCOPUS y la comparamos con otras bases mediante el motor de búsqueda proporcionado por las mismas (ver Tabla 1).

El método de búsqueda manual en revistas, conferencias, bibliotecas, etc. Se utilizó para revisar las características de los artículos generados como respuesta de la búsqueda automática. Los detalles presentamos en las siguientes subsecciones.

E. Estrategia de búsqueda

Las principales bibliotecas digitales que se utilizaron para buscar los artículos preliminares fueron: SCOPUS, IEEEExplore, Biblioteca digital ACM (ver Tabla 1), también se buscó manualmente artículos relacionados y que refuercen nuestro estudio en Google Académico sobre la adquisición de nuevos conocimientos orientado a la enseñanza de métodos, lenguajes y herramientas para el desarrollo de software. Adicionalmente, se revisó la información publicada en el libro "Creative working in the knowledge Economy" [24] y la revista "Intelligent Distributed Computing IX" volumen publicado en octubre 2016 [25].

Para realizar la búsqueda automática en las bibliotecas, utilizamos la cadena de búsqueda (ver Tabla 2) que consta de cuatro partes con el objetivo de cubrir los conceptos que representan el adquirir nuevos conocimientos orientados a la enseñanza de lenguajes de programación, herramientas, metodologías, ciclos y procesos en el desarrollo de software.

La primera parte de la cadena de búsqueda se relaciona con los conceptos de a quienes es dirigido el estudio (desarrollador de software), la segunda parte de la cadena de búsqueda con el nivel de conocimientos de un desarrollador

(novato, inexperto, nuevo o junior), la tercera parte de la cadena de búsqueda se refiere al concepto de adquirir o aprender y la cuarta parte está relacionada con los conocimientos que se transmite en lenguajes de programación, herramientas, metodologías, ciclos y procesos en el desarrollo de software.

En la Tabla 2 se muestra la cadena de búsqueda en donde se utiliza el operador booleano OR para unir términos alternativos entre “nuevo o inexperto o novato o junior” y “lenguaje de programación” o “herramientas” o “metodologías” o “ciclos de desarrollo” o “procesos de desarrollo”. Además, para unir cada parte principal se utiliza el operador booleano AND. La búsqueda se realizó mediante la aplicación de la cadena en los metadatos (es decir, título, resumen, abstract, introducción y palabras clave) de cada artículo.

La cadena se adaptó para ser aplicada en cada biblioteca digital al igual que en el buscador manual.

F. Búsqueda de estudios

Se utilizó un método de búsqueda automático en nuestro mapeo para recuperar estudios afines al tema. Se recuperó los estudios de la base de datos SCOPUS y se la comparó con varias de las bases de datos seleccionadas, según los términos de búsqueda establecidos dentro la sección H.

G. Alcance de la búsqueda.

El alcance de la búsqueda de este mapeo sistemático incluye el periodo de tiempo y la base de datos electrónica. En vista que con los mismos podemos obtener una integridad de los estudios primarios que son potencialmente relevantes para el tema de investigación planteado.

H. Periodo de tiempo.

El periodo de revisión incluyó estudios publicados entre el año 2007 y 2019, y para validar nuestra estrategia de búsqueda comparamos los contenidos y resultados obteniendo una muestra de 51 artículos primarios, (ver Tabla 3). Elegimos el año 2007 como inicio de la búsqueda para poder filtrar y descartar artículos muy antiguos. El mes de marzo 2019 seleccionamos como fin de la búsqueda ya que es cuando empezamos este estudio de mapeo.

I. Selección de estudios.

Para asegurar los resultados de la selección de artículos que componen nuestro mapeo sistemático, se utilizó el método automático de búsqueda y además se delimitó el periodo de años para filtrar los artículos, quedando solo aquellos publicados desde el año 2007 a marzo 2019.

1) *Criterios de selección.* En cuanto a los criterios de inclusión y exclusión que se tomaron en cuenta en nuestro mapeo sistemático tenemos:

- **Inclusión (I):** Publicaciones con citas adecuadas y que tengan un claro enfoque en la transmisión de nuevos conocimientos en la industria del software de lenguajes de programación, framework, metodologías, ciclos y procesos de desarrollo.

- **Exclusión (E):** Publicaciones que se enfoquen en su mayoría en educación, y que además no tenga una clara línea de estudio en cuanto a la transmisión de conocimientos en el marco de la industria del software.

Además, se establece un esquema de clasificación

preliminar a través de palabras clave, lectura de abstract e introducción. Una vez que se genera el listado preliminar se procede a la lectura de la publicación completa para analizarla en profundidad y volver a repetir el esquema de clasificación hasta obtener los resultados finales (ver Tabla 3).

El idioma de los estudios analizados no se considera como criterio de exclusión, ya que al configurar los motores de búsqueda de las bases de datos seleccionadas ya contemplamos ese filtro dejando por defecto el idioma inglés.

2) *Selección de estudios primarios.* Cada estudio primario que se obtuvo fue evaluado para determinar si está dentro de la temática que buscamos. Los estudios que cumplieron al menos una de las siguientes inclusiones fue seleccionada (ver Fig. 3):

- Lectura de secciones e introducción. Solo de artículos con un enfoque al software.
- Se limita la búsqueda desde el 2007 al 2019.
- Se realiza un estudio primario identificando 295 documentos que servirán como base.
- En los siguientes pasos revisamos los artículos, se excluye documentos que no representan mayor aportación, quedando 83 publicaciones.
- Finalmente, luego de una revisión completa quedan 51 artículos relevantes.

Con los pasos expuestos anteriormente consideramos que la búsqueda de publicaciones es suficientemente extensa y que las publicaciones nos brindan un buen estado del arte.

TABLA III
USO DE LOS CRITERIOS PARA SELECCIÓN DE ESTUDIOS

Paso de selección	Criterio usado
Primer paso: selección por metadatos	I
Segundo paso: selección por el título	I, E
Tercer paso: selección por el abstract	I, E
Cuarto paso: selección por el texto completo	I, E

TABLA IV
FUENTES DE SELECCIÓN

Fuente (búsqueda automatizada)	Estudios potenciales	Selección de estudios
SCOPUS	295	51
Búsqueda manual	Estudios potenciales	Selección de estudios
ACPI Limited	1	1
AIS/ICIS Ad. Office	5	1
AC Machinery	15	3
AC Machinery, Inc	7	1
CEUR-WS	2	1
Common Ground Publishing	1	1
Elsevier B.V.	7	3
Elsevier Ltd	7	2
IEEE Computer Society	11	6
IEE Engineers	1	1
IEE Engineers Inc.	37	9
IBIMA	1	1
John Wiley and Sons Inc.	1	1
Kluwer Academic Publishers	1	1
Little Lion Scientific	1	1
Nova Science Publishers, Inc.	2	1
SAE International	2	1
SAGE Publications Inc.	1	1
Science Press	1	1
Springer Verlag	11	3
Walter de Gruyter GmbH	1	1
Otros	113	10
Total	-	51

3) *Fuentes preliminares de selección.* Al aplicar nuestra estrategia de búsqueda y realizado el primer filtrado en las fuentes arroja los resultados preliminares (ver Tabla 4), por consiguiente, se seleccionaron los 295 estudios más relevantes y que cumplen los criterios de inclusión expuestos para realizar el nuevo filtrado.

Para el primer filtrado encontramos los siguientes problemas:

- Para el caso de estudios similares tipo extended en diferentes revistas se selecciona solo una versión.
- Para el caso en que un mismo estudio aparece en varias fuentes seleccionamos una sola vez de entre todas las fuentes. Se utiliza SCOPUS, ACM, entre otras fuentes.
- Ciertos resultados también lo encontramos relacionado en los campos de la gestión del conocimiento.

J. *Evaluación de la calidad.*

Como primera instancia se discutió a profundidad la calidad de los estudios seleccionados. Se revisó los artículos y se fue desarrollando listados de preselección que nuevamente eran analizados por los miembros del equipo, descartando a los que no tenía relevancia con el tema.

En esta parte se dio un peso de “1” si el contenido de la publicación tiene que ver con el tema propuesto, y se otorgó un peso negativo “-1” al artículo si el mismo se involucraba en el ámbito educativo, ya que en lo posible tratamos de no abordar temas muy específicos de E-learning.

En los primeros listados de preselección se tomó como herramienta de apoyo un cuestionario de escala de Likert para la evaluación, que contenía dos preguntas objetivas y que al momento de ir revisando los estudios se iba evaluando independientemente. La pregunta fue:

- El estudio ha sido citado por otros autores. La posible respuesta a esta pregunta fue: “SI (1)” si el estudio tiene más de 10 citas, “Parcialmente (0.5)” si el estudio ha sido citado en un rango de 1 a 10 veces; y “NO (0)” si el estudio no ha sido citado. Para esto nos apoyamos con el número de citas del artículo en Google Académico.
- El estudio ha sido publicado en revista, conferencia o libro de alto impacto tomando como referencia a marzo 2019. Las posibles respuestas fueron: “1” si es de alto impacto y “0” si no es de muy alto impacto. Para la evaluación de esta pregunta se considera el ranking de “SJR” (SCImago Journal Rank).

Con lo expuesto anteriormente, buscamos que los artículos seleccionados finalmente sean de aporte a nuestra investigación y que den soporte para el análisis del tema de investigación planteado.

K. *Extracción de datos*

Se considera que la búsqueda de documentación es extensa y por la cantidad de publicaciones el estado del arte se fundamenta para la transmisión de nuevos conocimientos en el desarrollo de software.

Para la extracción de datos, se emplea la estrategia basada en el conjunto de posibles respuestas para cada sub pregunta de investigación. Esto fue definido con anterioridad y esta estrategia asegura la aplicación de la clasificación. Esto se detalla a continuación.

La respuesta para RQ1 se la obtiene con el análisis de los marcos teóricos de la documentación finalmente

seleccionada. Al igual que del análisis de la parte introductoria de los 295 artículos primarios. Obteniendo diferentes métodos como:

- Método MRP (Multi Role Project).
- Rotación de conocimiento.
- E-learning como método tradicional.
- Conocimiento en la experiencia.
- Uso de redes sociales y medios online.
- Uso de metodologías ágiles.
- Plataformas de codificación social.

Para responder las preguntas RQ2, RQ3, se analiza los contenidos, estado del arte y resultados de los 51 artículos finales seleccionados. Donde se señala que es posible la transmisión de nuevos conocimientos entre desarrolladores de software siempre y cuando exista los canales adecuados y se utilice los métodos correctos, por ejemplo, los señalados en la respuesta de RQ1 donde se identifica el uso de medios virtuales u online, correcta documentación o software especializado [19], [26], [15].

Sobre el porcentaje o cantidad de conocimiento que se puede adquirir la limitante encontrada en la literatura revisada en este estudio [27], [28], [29], [24], [30], [31], [32], [33] se basa en la capacidad de los desarrolladores noveles por retener y practicar lo que el experto transmite y el interés que el novel le otorgue al aprendizaje de lo transmitido [1], [29], [34].

La respuesta a RQ4 se la obtiene de la experimentación y conclusiones de los proyectos realizados en la documentación revisada [7], [13], [27], [35]. Determinando que el conocimiento almacenado se puede compartir en cualquier etapa del desarrollo [23], [36]. Todo dependerá del medio o canal utilizado para la transmisión y de la capacidad de los desarrolladores de software para poder transmitir correctamente lo deseado [37], [38].

Finalmente, la respuesta para RQ5 se valida con los casos de pruebas de los estudios [10], [39], [40], [41], [42] y con una encuesta a 89 desarrolladores en diferente cargo, realizada como parte del proyecto. Se identifica que es un problema grande para las organizaciones el no poder transmitir adecuadamente el conocimiento durante el proceso de desarrollo de software (ver Fig. 1).

IV. RESULTADOS DEL ESTUDIO

Realizamos este estudio de mapeo sistemático de acuerdo con los procedimientos descritos en la sección II, a continuación, presentamos los resultados finales de la búsqueda y selección de estudios (ver Tabla 5).

A. *Resultados de búsqueda y selección*

La Tabla 5 muestra los resultados de los artículos seleccionados en nuestra búsqueda, en total se devolvieron 295 resultados con nuestra cadena y metadatos (primer paso), 51 registros fueron separados después que se revisaron los títulos (segundo paso), 161 artículos fueron excluidos después que se revisaron los abstract (Tercer paso). Posteriormente, 32 artículos se dejaron de lado al revisar el texto completo (cuarto paso), dejando finalmente 51 artículos en este mapeo sistemático (ver Fig. 3).

B. *Clasificación de resultados*

En esta sección se describe una breve calificación de los

estudios por año de publicación y lugar de publicación.

1) *Clasificación por año de publicación.* La distribución de los estudios seleccionados está comprendida entre los años 2007 y 2019. Identificando la clara tendencia que la problemática de transmisión de nuevos conocimientos entre desarrolladores de software está comenzando a surgir.

2) *Clasificación por tipo de publicación y fuente.* Cada artículo seleccionado se publicó como un documento de conferencia, periódico, revista, o capítulo de un libro. En la fig. 4 se muestra la distribución sobre los tipos de publicaciones, un 43% (22 estudios) son artículos de conferencias, 37% (19 estudios) son parte de una revista, 12% (6 estudios) son artículos independientes y el 8% (4 estudios) son capítulos de libro.

V. RESULTADOS DE LOS MÉTODOS DE TRANSMISIÓN DE NUEVOS CONOCIMIENTOS

En esta sección se describe las características más relevantes y emergentes de los métodos de transmisión de conocimientos entre desarrolladores de software, estudiados en los artículos de nuestro mapeo sistemático.

A. Transmitir nuevos conocimientos de desarrollo de software y trabajos relacionados.

Según el análisis de Karnouskos [43], con la era del internet los desarrolladores de aplicaciones tienen en sus manos una herramienta que permite ampliar sus servicios y su conocimiento. Las decisiones en el momento de aprender algo se amplía, ya que pueden utilizar casi cualquier información del mundo real, aunque en su mayoría esta dispersa y sin organización.

Nuestro estudio busca la forma de utilizar este conocimiento generado en situaciones de trabajo y conseguir el desarrollo profesional, personal y organizacional.

1) *Método MRP (Multi Role Project).* Para Panwong [5], el método MRP como instrumento para aprendizaje es de gran utilidad, se basa en proyectos aplicables, describiendo su implementación y transmitiendo conocimientos y habilidades profesionales para un mejor desempeño individual o grupal [19]. Lo que busca este modelo es proveer al equipo experiencias cercanas a las realidades profesionales [44].

El uso del modelo MRP supone un alto nivel de exigencia en la selección de los proyectos que servirán para el estudio. Una mala selección de proyectos puede ocasionar un cansancio físico y mental lo que dificultaría el aprendizaje [45].

2) *Rotación de conocimiento.* Se ha reconocido ampliamente que la creación de conocimiento pasa por los desarrolladores y es un factor crítico de éxito para la innovación en las organizaciones. En particular, en la ingeniería de software, el desarrollo de nuevos productos de software requiere los conocimientos novedosos y útiles [38].

Desde el punto de vista de Isong [46], la rotación de personal experto entre empresas puede ser una gran herramienta para transferir conocimientos externos. Dividiendo el conocimiento en tres capas, la primera es la base del conocimiento de la empresa, la segunda son los marcos de conocimiento y la estructura organizacional interna, y la tercera es la denominada “Dinámica de

conocimiento” la cual surge de la iteración entre individuos y la empresa [47].

Ahora, uno de los problemas que se enfrenta es la dependencia del apoyo laboral externo y su voluntad para transmitir conocimientos. Con esto se mide la capacidad de las empresas para atraer a personas capaces de transmitir experiencias. Las contrataciones de personal por este método pasarían por las denominadas decisiones estructurales [46]. Es importante que la industria una esfuerzos para facilitar los elementos para conseguir una colaboración a gran escala, rompiendo barreras organizativas y culturales [48]. La meta es que el conocimiento rote en la industria del software permitiendo un desarrollo conjunto [49].

3) *E-learning como método tradicional.* La enseñanza presencial combinado con la tecnología aun dominan el proceso de aprendizaje [50]. Lo que se convierte en un desafío a solucionar, en búsqueda de la innovación y desarrollo de nuevas habilidades como lo es el e-learning, que se basa en la mezcla de conocimiento entre personal calificado y juventud [36].

Los avances en tecnología nos permite mejorar casos de usos en los métodos de enseñanza [39] y la utilización del e-learning no solo para enseñar conocimiento a la gente, sino también para proporcionarles más información sobre cualquier tema que puedan necesitar ocasionalmente [51]. El uso de estas herramientas depende de las habilidades de los aprendices [35] y son estas herramientas que deben garantizar la correcta práctica de patrones de comunicación [41], [52].

En la actualidad según Ramdas [52], existe un desfaz entre la tecnología y la sociedad, hace falta enfocarse en la co-creación de conocimiento que viene hacer el proceso cooperativo y multidireccional para generar nuevos conocimientos, habilidades y competencias, de modo que el resultado sea para cada sujeto más rentable de lo que una sola persona conseguiría.

Para Minghini [53], cuando hablamos de crear conocimiento se puede dividir de acuerdo con el grado de colaboración en:

- Autónomos: No hay colaboración en absoluto.
- Colaborativo: Todos colaboran y contribuyen.

4) *Conocimiento en la experiencia.* El uso de la información que se extrae de las experiencias se puede utilizar como oportunidades para aprendizaje y la generación de nuevos conocimientos. Al aprender de las fallas internas las organizaciones pueden llegar a la raíz del problema y no atribuir los mismos a factores externos, lo que permite aumentar el nivel de atención para ver nuevas oportunidades [54], [55].

En el estudio de Moorthy [56], se identifica que lamentablemente hasta la fecha no existe un IDE Web que apoye a los desarrolladores novatos para un rápido aprendizaje de un lenguaje de programación sino solo con el transcurso del tiempo y la experiencia.

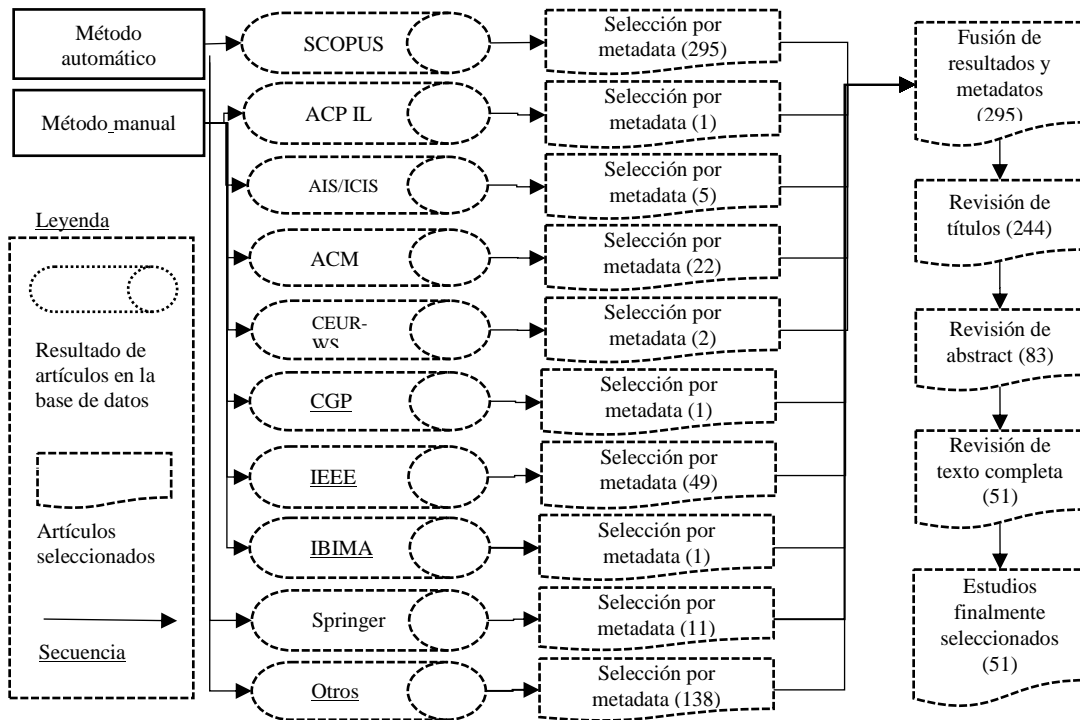


Fig. 3. Resultados de la búsqueda y selección de estudios

5) *Redes sociales y medios online.* Otra forma para compartir conocimientos son las redes sociales, entre ellos los más importantes son los sitios web, foros y blogs [57]. En las mismas se desarrollan conversaciones donde el contenido se comparte abiertamente, aparecen interacciones entre personas y entidades digitales, etc. En otras palabras, estas redes forman una verdadera comunidad digital mediante el desarrollo de estructuras de comunicación [58].

La teoría del conectivismo, promulga que el proceso de aprendizaje se enriquece con la conexión de los estudiantes, profesores y recursos en línea [59], lo que favorece a la aparición de comunidades de aprendizaje [58]. Desde otra perspectiva, el aprendizaje virtual con el uso de plataformas es una de las técnicas más utilizadas [60], en vista que brinda oportunidades de crear redes de conocimiento, aunque se tiene el inconveniente que no siempre se comparte material educativo de calidad [29].

Entre las principales ventajas que estos sistemas presentan tenemos la creación de conocimiento a través del aprendizaje, la difusión, la preservación, la captura y explicación, intercambio y comunicación colaborativa, acceso, uso y reutilización de archivos con cierto grado de conocimiento [26], [60].

6) *Uso de metodologías ágiles para enseñar y aprender.* Desde el punto de vista de Crawford y Papamichail [38] y [62], el entorno empresarial en la industria del software actual es extremadamente dinámico y competitivo. Con el fin de mantener la posición y ganar en tiempo y reducir costos es imperativo que las organizaciones sean creativas en sus esfuerzos para retener el conocimiento generado.

Las revisiones y retrospectivas al final de cada iteración proporcionan oportunidades para el equipo de desarrollo para interactuar y promover el intercambio de conocimientos, aunque esto no es del todo explotado y son pocas las organizaciones que explotan esto el inconveniente que se genera es la veracidad del conocimiento que se puede transmitir, ya que al ser ágil puede pasarse por alto ciertos filtros en la información [38].

7) *Adopción de plataformas de codificación social.* La adopción de plataformas de codificación social entre desarrolladores de software permite aumentar la colaboración y productividad, así como la reutilización de código [63]. A esta técnica se la denomina “trabajo con bancos de prueba”. Se presenta un inconveniente en la curva de aprendizaje de nuevas tecnologías por la escasa tasa de interés que los desarrolladores prestan a estas plataformas [30] por lo que no se utiliza a su máximo nivel sino únicamente para recolectar código fuente [42].

VI. DISCUSIÓN

Hoy en día, las empresas obtienen conocimiento nuevo desde diferentes medios o canales, pero de manera dispersa. Es tarea fundamental buscar la manera de recopilarlo y que sirva de base para futuras generaciones de empleados.

El desarrollo de software es un proceso complejo que involucra a diversas personas cuyos conocimientos por lo general no son aprovechados, entre estos podemos señalar: reglas de negocio, diseño de interfaces, levantamiento de requisitos y herramientas utilizadas [38].

El aprendizaje en general puede ocurrir en tres niveles, individual, grupal u organizacional. Diferentes equipos pueden interactuar en un determinado proyecto y comenzar a compartir experiencias de manera tácita lo que da lugar al nacimiento del conocimiento organizacional [64].

Un punto importante que sugiere la literatura a tener en cuenta en el área de software son las características especiales de cada desarrollador. Su capacidad de internalización, retención, asimilación y comunicación pueden realizar la

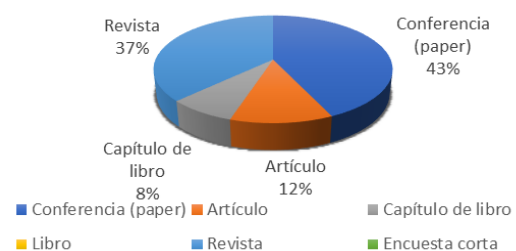


Fig. 4. Distribución de estudios seleccionados sobre el tipo de publicación

TABLA V
LISTADO DE ARTÍCULOS PRIMARIOS

N.	Autor(es)
[S1]	De Oliveira, Canedo, Faria, Amaral & Bonifacio
[S2]	Moorthy, T'ing, Wei, Zi Mei, Yee, Jia Wern & Xin Y.M.
[S3]	Sharvani V., Abhinav K., Dubey A., Arora V. & D'Souza M.
[S4]	Hussain S., Keung J., Sohail M.K., Khan A.A. & Ilahi M.
[S5]	Almaiah M.A., Alamri M.M.
[S6]	Sun X.
[S7]	Macedo A.A., Baranauskas J.A., Bulcão-Neto R.F.
[S8]	Vucnik M., Fortuna C., Solc T. & Mohorcic M.
[S9]	Politowski C., Fontoura L.M., Petrillo F. & Guéhéneuc Y.-G.
[S10]	Gossen F., Kuhn D., Margaria T. & Lamprecht A.-L.
[S11]	Papamichail M., Diamantopoulos T., & Symeonidis A.
[S12]	Hu X., Li G., Xia X., Lo D. & Jin Z.
[S13]	Gulec U., Yilmaz M., Isler V., O'Connor R.V. & Clarke P.
[S14]	Dantas R., Carvalho A., Marcílio D., Silva U. & Bonifácio R.
[S15]	Chen X., Wang L.-P., Gu Q., Wang Z., Ni C., & Wang Q.-P.
[S16]	Correa D., Isaza F.A., Mazo R. & Giraldo G.L.
[S17]	Dönmez D. & Grote G.
[S18]	Reyna J., Hanham J. & Meier P.
[S19]	Karnouskos S., Gaertner N., Verzano N., Beck F., & Other
[S20]	Huppe S., Saied M.A. & Sahraoui H.
[S21]	Yulianto B., Prabowo H., Kosala R. & Hapsara M.
[S22]	Durán E.B., Álvarez M.M. & Únzaga S.I.
[S23]	Király S., Balla T. & Kusper G.
[S24]	Rúbio T.R.P.M., Cardoso H.L. & da Costa Oliveira E.
[S25]	Lyon L.A. & Jameson K.
[S26]	Chacón J., Vargas H., Farias G., Sánchez J. & Dormido S.
[S27]	Santos A.R., Sales A., Fernandes P. & Nichols M.
[S28]	Gañán D., Caballé S., Conesa J. & Xhafa F.
[S29]	Gañán D., Caballé S., Conesa J., Oda T. & Barolli L.
[S30]	Hahm O., Baccelli E., Petersen H., Wählich M. & Schmidt T.C.
[S31]	Murphy-Hill E.
[S32]	Panwong P. & Kemavuthanon K.
[S33]	Chouseinoglou O. & Bilgen S.
[S34]	Díaz Redondo R.P., Fernández Vilas A. & Gil Solla A.
[S35]	Isong B. & Dominic E.
[S36]	Bienkowski M., Feng M. & Means B.
[S37]	Lancor L. & Katha S.
[S38]	Potkonjak V., Jovanović K., Holland O. & Uhomoihi J.
[S39]	Vitkin L.
[S40]	Hvorecky J.
[S41]	Holsapple C.
[S42]	Rodríguez J.A., Macias R., Molgo J., Guerra D. & Pi M.
[S43]	Ramdas C.V., Parimal N., Utkarsh M., Ramya K. & Smitha B.P.
[S44]	Boticki I., Wong L.-H. & Looi C.-K.
[S45]	Kolekar, S., Pai, R. M., & MM, M. P.
[S46]	Caulfield C., Maj S.P. & Veal D.
[S47]	Fullmer Jr. D. & Mavris D.N.
[S48]	Conchúir E.Ó., Ågerfalk P.J., Olsson H.H. & Fitzgerald B.
[S49]	Crawford B. & de La Barra C.L.
[S50]	Sena J., Coget J.-F. & Shani A.B.
[S51]	Yu H., Shen Z., Low C.P. & Miao C.

diferencia para que la transmisión de nuevos conocimientos sea exitosa [37].

Manejar adecuadamente el conocimiento se convierte en un recurso clave para muchas organizaciones porque puede mejorar su competitividad, siendo el conocimiento una fuente de innovación dado que la calidad del software está ligada estrictamente con la calidad del proceso con el que se desarrolló el software [53].

La mayoría del conocimiento organizacional no se produce en ambientes formales, sino informales, se estima que cerca del 80% del conocimiento organizacional se realiza en el ambiente informal [64].

Ahora, para Murphy- Hill [65] hay que tomar en cuenta los puntos fuertes del denominado aprendizaje social de persona a persona, las nuevas tecnologías pueden ayudar a los desarrolladores de software a aprender unos de otros de manera más eficiente y productiva [66]. En particular la técnica del screencasting social continuo dentro de las redes sociales y medios online, que está resultando una técnica prometedora pero no muy explotada.

Los trabajos analizados en este documento insisten en que es necesario el estudio de la transmisión de nuevo conocimiento en una determinada organización. Especialmente el del conocimiento tácito ya que no se ha estudiado mucho esto.

A. Amenazas de la validez.

Los resultados de este estudio de mapeo sistemático pueden verse afectados por el periodo de años en la búsqueda, criterios de selección, distribución de estudios o la inexactitud de la extracción de datos, el sesgo de información y además por los diferentes estilos de aprendizaje en cada individuo, que hace que no haya una única forma de transmitirle los conocimientos.

B. Búsqueda de estudios incompleto.

Puede darse el caso de existir estudios relevantes que no se recuperaron, que pueden afectar a la integridad de la búsqueda de estudios. Para mitigar este punto se buscó con énfasis y con varias cadenas de búsqueda en las principales bases de datos electrónicas, las cuales incluyen gran número de artículos, revistas, conferencias, libros, etc.

La búsqueda principal de artículos se realizó en SCOPUS y se recuperó manualmente los mismo en la respectiva base de datos indexada, también contribuyó de buena manera el analizar ciertos términos relacionados con la transmisión de nuevos conocimientos entre desarrolladores de software en Google Scholar para ampliar el campo de conocimiento.

C. Sesgo en la selección de estudios.

De cierta manera puede existir sesgo a favor de los investigadores en relación con el estudio del mapeo. El primer paso para excluir el sesgo fue establecer criterios de inclusión y exclusión para la selección de artículos. No obstante, al ser estos criterios interpretables pueden los resultados de la selección ser variados. Los investigadores llegaron a consensos en los criterios de selección y coordinaron sus esfuerzos para mitigar cualquier sesgo personal en la selección de artículos.

D. Inexactitud en la extracción de datos.

El sesgo en la extracción de datos puede resultar en la inexactitud de los datos extraídos, lo que puede afectar la clasificación y los resultados del análisis de los estudios seleccionados. Este sesgo se redujo en tres medidas. En primer lugar, los investigadores discutieron los elementos de datos que se extraen en este estudio y se logró un acuerdo sobre el significado de cada elemento de datos. Por ejemplo, en el protocolo del estudio, definimos explícitamente qué son las nociones y actividades relacionadas con la transmisión de nuevos conocimientos entre desarrolladores de software, y se resolvieron los desacuerdos entre los investigadores. En segundo lugar, se realizó una extracción doble de datos, el un investigador lo realizaba en una primera instancia, una vez finalizado entregaba al segundo investigador quien validaba la información hasta llegar a un consenso. Esta medida mejoró aún más la precisión de los datos extraídos. En tercer lugar, los datos fueron verificados por los dos investigadores, y de nuevo se discutieron y resolvieron los desacuerdos.

VII. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo presentamos una revisión de literatura que identifica los métodos que se utilizan para la transmisión de

conocimientos y sus principales dificultades y restricciones.

Más aún cuando el conocimiento no es tratado de la mejor manera en las organizaciones originando un nivel de complejidad, esto lo validamos con parte de la encuesta realizada como parte del análisis cuantitativo presentado (ver Fig. 2).

Este trabajo es parte de una investigación más completa que pretende definir un meta-modelo que ayude a la creación o desarrollo de sistemas informáticos que ayuden al aprendizaje y transmisión de nuevos conocimientos en lenguajes de programación, herramientas, metodologías, ciclos y procesos entre desarrolladores y evitar la pérdida de los mismos.

Las cinco sub preguntas de investigación planteadas se enfocan en un nuevo y emergente problema para las organizaciones del software, que es el nuevo conocimiento y su tratamiento.

La heterogeneidad de conocimientos entre desarrolladores de una organización es un problema relevante que afecta directamente al avance de proyectos de software [27], [67]. La rotación de personal que adquiere cierto grado de conocimiento y posteriormente su cambio conllevan a la pérdida de ese nuevo conocimiento que serviría como guía para una nueva generación de empleados novatos [64].

En la Fig. 1 expuesta en la introducción del presente estudio, reforzamos lo expuesto anteriormente. Es innumerable la cantidad de nuevo conocimiento que se puede transmitir, de hecho, en cada problema por resolver puede generar más una situación de solución [46].

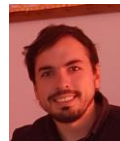
REFERENCIAS

- [1] P. J. Ågerfalk, E. Ó. Conchúir, H. H. Olsson, and B. Fitzgerald, "Global Software Development: Where are the Benefits," *Commun. ACM*, 2009.
- [2] D. Ganán, S. Caballé, J. Conesa, and F. Xhafa, "An application framework to systematically develop complex learning resources based on collaborative knowledge engineering," *Int. J. Appl. Math. Comput. Sci.*, 2015.
- [3] A. Singla, I. S. Ahuja, and A. P. S. Sethi, "Technology push and demand pull practices for achieving sustainable development in manufacturing industries," *J. Manuf. Technol. Manag.*, 2018.
- [4] T. V. Korsakova, E. A. Chelnokova, S. N. Kaznacheeva, I. B. Bicheva, A. L. Lazutina, and T. V. Perova, "Transformation of corporate culture in conditions of transition to knowledge economics," *Int. J. Environ. Sci. Educ.*, 2016.
- [5] P. Panwong and K. Kemavuthanon, "Problem-based learning framework for junior software developer: Empirical study for computer programming students," *Wirel. Pers. Commun.*, 2014.
- [6] L. A. Lyon and K. Jameson, "From clicks to code: Resources women use to learn to code in apex," in *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, 2015.
- [7] C. M. C. De Oliveira, E. D. Canedo, H. Faria, L. H. V. Amaral, and R. Bonifacio, "Improving Student's Learning and Cooperation Skills Using Coding Dojos (In the Wild!)," in *Proceedings - Frontiers in Education Conference, FIE*, 2019.
- [8] H. Yu, Z. Shen, C. P. Low, and C. Miao, "Transforming learning through agent augmented virtual world," in *Proceedings - The 8th IEEE International Conference on Advanced Learning Technologies, ICALT 2008*, 2008.
- [9] J. Chacón, H. Vargas, G. Farias, J. Sánchez, and S. Dormido, "EJS, JIL Server, and LabVIEW: An Architecture for Rapid Development of Remote Labs," *IEEE Trans. Learn. Technol.*, 2015.
- [10] D. Correa, F. A. Isaza, R. Mazo, and G. L. Giraldo, "CME – A web application framework learning technique based on concerns, micro-learning and examples," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.
- [11] L. Lancor and S. Katha, "Analyzing PHP frameworks for use in a project-based software engineering course," 2013.
- [12] O. Hahm, E. Baccelli, H. Petersen, M. Wählich, and T. C. Schmidt, "Demonstration abstract: simply RIOT: teaching and experimental research in the internet of things," in *Proceedings of the 13th international symposium on Information processing in sensor networks*, 2014, pp. 329–330.
- [13] J. Capote, C. Llantén, C. Pardo, A. González, and C. Collazos, "Gestión del conocimiento como apoyo para la mejora de procesos software en las micro, pequeñas y medianas empresas," *Rev. Ing. e Investig.*, 2008.
- [14] J. A. Rodríguez, R. Macias, J. Molgo, D. Guerra, and M. Pi, "GCS component development cycle," in *Software and Cyberinfrastructure for Astronomy II*, 2013.
- [15] L. Vitkin, "Lean Model-Based Development: Practical Approach," *SAE Int. J. Passeng. Cars - Electron. Electr. Syst.*, 2013.
- [16] R. Dantas *et al.*, "Reconciling the past and the present: An empirical study on the application of source code transformations to automatically rejuvenate Java programs," in *25th IEEE International Conference on Software Analysis, Evolution and Reengineering, SANER 2018 - Proceedings*, 2018.
- [17] J. Reyna, J. Hanham, and P. Meier, "The Internet explosion, digital media principles and implications to communicate effectively in the digital space," *E-Learning Digit. Media*, 2018.
- [18] D. Gañán, S. Caballé, J. Conesa, T. Oda, and L. Barolli, "Evaluation of using software infrastructure and multimedia technologies for the creation of complex learning resources," in *Proceedings - 2014 International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS 2014*, 2014.
- [19] F. Gossen, D. Kühn, T. Margaria, and A.-L. Lamprecht, "Computational thinking: learning by doing with the Cinco adventure game tool," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 2018, vol. 1, pp. 990–999.
- [20] J. Sena, J.-F. Coget, and A. B. (Rami) Shani, "Designing for Agility as an Organizational Capability: Learning from a Software Development Firm," *Int. J. Knowledge, Cult. Chang. Manag. Annu. Rev.*, 2016.
- [21] T. R. P. M. Rúbio, H. L. Cardoso, and E. da Costa Oliveira, "MAESTROS: Multi-agent simulation of rework in open source software," *Stud. Comput. Intell.*, 2016.
- [22] E. B. Durán, M. M. Álvarez, and S. I. Únzaga, "Ontological model for the personalization of U-learning applications," in *2016 8th Euro American Conference on Telematics and Information Systems, EATIS 2016*, 2016.
- [23] W. C. Hu, T. T. Wei, C. F. Yang, and H. C. Jiau, "Reactive Programming Collaborative Agents for Self-Managing Software Teams," in *Proceedings - 2016 International Computer Symposium, ICS 2016*, 2017.
- [24] B. Gros, "Retos y tendencias sobre el futuro de la investigación acerca del aprendizaje con tecnologías digitales," *Rev. Educ. a Distancia*, 2016.
- [25] J. Hordern, "Creative working in the knowledge economy," *Br. J. Educ. Stud.*, vol. 66, no. 2, pp. 274–276, 2018.
- [26] V. Potkonjak, K. Jovanović, O. Holland, and J. Uhomoihibi, "Distance learning and skill acquisition in engineering sciences: Present state and prospects," *Multicult. Educ. Technol. J.*, 2013.
- [27] K. Moororthy *et al.*, "Is facebook useful for learning? A study in private universities in Malaysia," *Comput. Educ.*, 2019.
- [28] S. Király, T. Balla, and G. Kúspér, "Code School: Learn to code by practicing."
- [29] A. Raicu and G. Raicu, "Innovation in engineering education through computer assisted learning and virtual university model," in *IOP Conference Series: Materials Science and Engineering*, 2015.
- [30] C. Caulfield, S. P. Maj, and D. Veal, "Learning Java with Sun SPOTS," in *Advanced Techniques in Computing Sciences and Software Engineering*, Springer, 2010, pp. 251–256.
- [31] I. Boticki, L. H. Wong, and C. K. Looi, "Experiences in implementing and using a technological framework for mobile collaborative learning of mathematics and Chinese," *Int. J. Mob. Learn. Organ.*, 2012.
- [32] X. Sun, "Towards Learning-augmented Languages," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 959–961.
- [33] M. Bienkowski, M. Feng, B. Means, and others, "Enhancing teaching and learning through educational data mining and learning analytics: An issue brief," *US Dep. Educ. Off. Educ. Technol.*, vol. 1, pp. 1–57, 2012.
- [34] X. Hu, G. Li, X. Xia, D. Lo, and Z. Jin, "Deep code comment generation," in *Proceedings of the 26th Conference on Program Comprehension*, 2018, pp. 200–210.
- [35] P. Gil, "Short Project-Based Learning with MATLAB Applications to Support the Learning of Video-Image Processing," *J. Sci. Educ. Technol.*, 2017.

- [36] M. Blaga, R. Harpa, I. R. Radulescu, and Z. Stepjanovic, "E-learning for textile enterprises innovation improvement," in *IOP Conference Series: Materials Science and Engineering*, 2017.
- [37] M. G. Colombo, E. Piva, and C. Rossi-Lamastra, "Authorising employees to collaborate with communities during working hours: When is it valuable for firms?," *Long Range Plann.*, 2013.
- [38] B. Crawford and C. L. de La Barra, "eXtreme programming meets knowledge creation and creativity," in *Knowledge Management and Innovation in Advancing Economies: Analyses and Solutions - Proceedings of the 13th International Business Information Management Association Conference, IBIMA 2009*, 2009.
- [39] M. Almaiah and M. Al Amri, "PROPOSING A NEW TECHNICAL QUALITY REQUIREMENTS FRAMEWORK FOR MOBILE LEARNING APPLICATIONS," *J. Theor. Appl. Inf. Technol.*, vol. 96, 2018.
- [40] V. Sharvani, K. Abhinav, A. Dubey, S. Jain, V. Arora, and M. D'Souza, "FrAppLe: A Framework for Apprenticeship Learning," in *Proceedings of the 12th Innovations on Software Engineering Conference (formerly known as India Software Engineering Conference)*, 2019, p. 26.
- [41] S. Hussain, J. Keung, M. K. Sohail, A. A. Khan, and M. Ilahi, "Automated framework for classification and selection of software design patterns," *Appl. Soft Comput. J.*, 2019.
- [42] M. Vucnik, C. Fortuna, T. Solc, and M. Mohorcic, "Integrating Research Testbeds into Social Coding Platforms," in *2018 European Conference on Networks and Communications, EuCNC 2018*, 2018.
- [43] S. Karnouskos *et al.*, "Experiences in integrating Internet of Things and cloud services with the robot operating system," in *Proceedings - 2017 IEEE 15th International Conference on Industrial Informatics, INDIN 2017*, 2017.
- [44] D. Dönmez and G. Grote, "Two sides of the same coin – how agile software development teams approach uncertainty as threats and opportunities," *Inf. Softw. Technol.*, 2018.
- [45] R. P. Díaz Redondo, A. Fernández Vilas, J. J. Pazos Arias, and A. Gil Solla, "Collaborative and role-play strategies in software engineering learning with web 2.0 tools," *Comput. Appl. Eng. Educ.*, 2014.
- [46] B. Isong and E. Dominic, "An investigation of software development activities among undergraduate students: A case study," in *Proceedings - 2014 International Conference on Computational Science and Computational Intelligence, CSCI 2014*, 2014.
- [47] I. Jøranli, "Managing organisational knowledge through recruitment: searching and selecting embodied competencies," *J. Knowl. Manag.*, 2018.
- [48] Y. Ma, G. Luo, X. Zeng, and A. Chen, "Transfer learning for cross-company software defect prediction," *Inf. Softw. Technol.*, 2012.
- [49] O. Chouseinoglou and S. Bilgen, "Introducing critical thinking to software engineering education," in *Studies in Computational Intelligence*, 2014.
- [50] D. Fullmer Jr. and D. N. Mavris, "Concurrent product data modeling for collaborative design environments," in *Proceedings of the 2010 Conference on Grand Challenges in Modeling & Simulation*, 2010.
- [51] J. Hvorecky, "An Integral Approach to Online Education: An Example.," *Proc. Int. Conf. e-Learning*, 2013.
- [52] C. V. Ramdas, N. Parimal, M. Utkarsh, S. Sumit, K. Ramya, and B. P. Smitha, "Application of sensors in Augmented Reality based interactive learning environments," in *Proceedings of the International Conference on Sensing Technology, ICST*, 2012.
- [53] M. Minghini *et al.*, "FOSS4G as a key building block for case-based learning in geographic information education," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 2017.
- [54] A. Carmeli and A. Dothan, "Generative work relationships as a source of direct and indirect learning from experiences of failure: Implications for innovation agility and product innovation," *Technol. Forecast. Soc. Change*, 2017.
- [55] B. Yulianto, H. Prabowo, R. Kosala, and M. Hapsara, "Harmonik = ++(Web IDE)," in *Procedia Computer Science*, 2017.
- [56] K. Moorthy *et al.*, "Is facebook useful for learning? A study in private universities in Malaysia," *Comput. Educ.*, vol. 130, pp. 94–104, 2019.
- [57] J. Cruz-Benito, O. Borrás-Gene, F. J. García-Penalvo, A. F. Blanco, and R. Theron, "Learning Communities in Social Networks and Their Relationship with the MOOCs," *Rev. Iberoam. Tecnol. del Aprendiz.*, 2017.
- [58] S. Kolekar, R. M. Pai, and M. M. Manohara Pai, "Adaptive user interface for e-learning applications based on learning styles using Web Logs analysis: A hybrid cloud architecture," in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2016.
- [59] A. A. Macedo, J. A. Baranauskas, and R. F. Bulcão-Neto, "The Evolution of a Healthcare Software Framework: Reuse, Evaluation and Lessons Learned," in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2018, pp. 1–9.
- [60] U. Gulec, M. Yilmaz, V. Isler, R. V. O'Connor, and P. Clarke, "Adopting virtual reality as a medium for software development process education," 2018.
- [61] A. R. Santos, A. Sales, P. Fernandes, and M. Nichols, "Combining Challenge-Based Learning and Scrum Framework for Mobile Application Development," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '15*, 2015.
- [62] M. Papamichail, T. Diamantopoulos, I. Chrysovergis, P. Samlidis, and A. Symeonidis, "User-perceived reusability estimation based on analysis of software repositories," in *2018 IEEE International Workshop on Machine Learning Techniques for Software Quality Evaluation, MaLTeSQuE 2018 - Proceedings*, 2018.
- [63] A. Heredia, J. García-Guzmán, A. Amescua, and M. I. Sánchez-Segura, "Interactive knowledge asset management: Acquiring and disseminating tacit knowledge," *J. Inf. Sci. Eng.*, 2013.
- [64] C. W. Holsapple, "Handbook on knowledge management: Knowledge matters," *Int. Handb. Inf. Syst.*, 2003.
- [65] E. Murphy-Hill, "The future of social learning in software engineering," *Computer (Long Beach, Calif.)*, 2014.
- [66] S. Huppe, M. A. Saied, and H. Sahrouri, "Mining complex temporal API usage patterns: An evolutionary approach," in *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017*, 2017.
- [67] I. A. Mian and others, "Analysis of self-assessment skills of novice software Developers working in professional software Development industry," Bahria University Islamabad Campus, 2018.



Oscar Fernando Ibarra Torres nació en Ambato, Ecuador, en 1985. Se recibió Magister en Gerencia de Sistemas de Información de la Universidad Técnica de Ambato (UTA). Desde 2019, es profesor en la Universidad Técnica de Ambato (UTA), Ambato, Ecuador. Es autor de 6 artículos de revistas de investigación. Es desarrollador de software de varios sistemas académicos tanto de pregrado y posgrado. Sus intereses de investigación incluyen metodologías de desarrollo, desarrollo de software.



Mario Matías Urbieta nació en Formosa, Argentina, en 1981. Se recibió en B.S. y P.h.D en Ciencias Informáticas de la Universidad Nacional de La Plata (UNLP). Desde 2008, es asistente de investigación en el Laboratorio de Investigación y Formación en Informática Avanzada (LIFIA), La Plata, Buenos Aires, Argentina. Es autor de 1 capítulo de libro, 10 artículos de revistas y 20 artículos de conferencia. Sus intereses de investigación incluyen la separación de concerns en aplicaciones web, ingeniería de requisitos Agile y aumento web. Enseña conceptos de programación orientada a objetos e ingeniería web a nivel de licenciatura y doctorado en UNLP y Universidad Abierta Interamericana.



Gustavo Rossi recibió su diploma de doctorado en PUC-Rio, Brasil en 1996. Su tesis doctoral fue el desarrollo del enfoque de diseño OOHDm, uno de los métodos maduros para el desarrollo de aplicaciones web. Él es actualmente profesor titular en la Facultad de Informática, Universidad Nacional de la Plata. Ha sido profesor visitante en el Universidades de Lyon y Montpellier en Francia y ha recibido el Habilitation pour Diriger Recherches (HDR) en INSA-Lyon. El ha estado como parte del comité de PC de las conferencias más importantes de su campo de investigación como ACM, WWW, ICWE, y el ACM hipertexto. Es miembro del consejo de redacción de IEEE. Internet Computing, IEEE IT Professional, WWW Journal, Journal of Web Ingeniería, Revista de Servicios y Aplicaciones de Internet y la Revista Internacional de Sistemas de Información Cooperativa.



Alejandra Beatriz Lliteras nació en la ciudad de La Plata, provincia de Buenos Aires, Argentina. Es Magister en Tecnología Informática Aplicada en Educación y Licenciada en Informática. Es Profesora Adjunta dedicación exclusiva en Universidad Nacional de La Plata, Facultad de Informática y pertenece al centro LIFIA. Es Investigadora Asistente Asociada CICPBA. Es docente en grado y posgrado de temas relacionados a bases de datos. Sus áreas de interés en investigación incluyen Ingeniería de Software, HCI, Computación Móvil, Aprendizaje Móvil y Ciencia Ciudadana.