



ISSN: 1646-9895

Revista Ibérica de Sistemas e Tecnologias de Informação
Iberian Journal of Information Systems and Technologies

M a r z o 1 8 • M a r c h 1 8



©AISTI 2018 <http://www.aisti.eu>

Nº 26

Edição / Edición

Nº 26, 03/2018

Tiragem / Tirage: 1000

Preço por número / Precio por número: 17,5€

Subscrição anual / Suscripción anual: 30€ (2 números)

ISSN: 1646-9895

Depósito legal:

Indexação / Indexación

Academic Journals Database, CiteFactor, Dialnet, DOAJ, DOI, EBSCO, GALE, IndexCopernicus, Index of Information Systems Journals, ISI Web of Knowledge, Latindex, ProQuest, QUALIS, SciELO, SCImago, Scopus, SIS, Ulrich's.

Propriedade e Publicação / Propriedad y Publicación

AISTI – Associação Ibérica de Sistemas e Tecnologias de Informação

Rua Quinta do Roseiral 76, 4435-209 Rio Tinto, Portugal

E-mail: aistic@gmail.com

Web: <http://www.risti.xyz>

Director

Álvaro Rocha, Universidade de Coimbra

Coordenadores da Edição / Coordinadores de la Edición

Jezreel Mejia, Centro de Investigación en Matemática (CIMAT)

Mirna Muñoz, Centro de Investigación en Matemática (CIMAT)

Tomas San Feliu, Universidad Politécnica de Madrid

Conselho Editorial / Consejo Editorial

Carlos Ferrás Sexto, Universidad de Santiago de Compostela

Gonçalo Paiva Dias, Universidade de Aveiro

Jose Antonio Calvo-Manzano Villalón, Universidad Politécnica de Madrid

Luís Paulo Reis, Universidade do Minho

Manuel Pérez Cota, Universidad de Vigo

Ramiro Gonçalves, Universidade de Trás-os-Montes e Alto Douro

Conselho Científico / Consejo Científico

Adolfo Lozano-Tello, Universidad de Extremadura, ES

Adrián Hiebra Pardo, Universidad de Santiago de Compostela, ES

Adriano Pasqualotti, Universidade de Passo Fundo, BR

Alberto Fernández, Universidad Rey Juan Carlos, ES

Alberto Bugarín, Universidad de Santiago de Compostela, ES

Alejandro Medina, Universidad Politécnica de Chiapas, MX

Alejandro Rodríguez González, Universidad Politécnica de Madrid, ES

Alejandro Peña, Escuela de Ingeniería de Antioquia, CO

Alexandre L'Erario, Universidade Tecnológica Federal do Paraná, BR

Alma María Gómez-Rodríguez, Universidad de Vigo, ES

Álvaro E. Prieto, Universidad de Extremadura, ES

Ana Azevedo, Instituto Politécnico do Porto, PT

Ana Cristina Ramada Paiva, FEUP, Universidade do Porto, PT

Ana Isabel Veloso, Universidade de Aveiro, PT

Ana Maria Correia, ISEGI, Universidade Nova de Lisboa, PT

Ana Paula Afonso, Instituto Politécnico do Porto, PT

Anabela Mesquita, Instituto Politécnico do Porto, PT

Angelica Caro, Universidad del Bío-Bío, CL
Ania Cravero, Universidad de La Frontera, CL
Antoni Lluís Mesquida Calafat, Universitat de les Illes Balears, ES
Antonia Mas Pichaco, Universitat de les Illes Balears, ES
António Coelho, FEUP, Universidade do Porto, PT
António Godinho, ISLA-Gaia, PT
Antonio Jesus Garcia Loureiro, Universidad de Santiago de Compostela, ES
António Pereira, Instituto Politécnico de Leiria, PT
Armando Mendes, Universidade dos Açores, PT
Arnaldo Martins, Universidade de Aveiro, PT
Arturo J. Méndez, Universidad de Vigo, ES
Baltasar García Pérez-Schofield, Universidad de Vigo, ES
Benjamim Fonseca, Universidade de Trás-os-Montes e Alto Douro, PT
Bráulio Alturas, ISCTE - Instituto Universitário de Lisboa, PT
Brenda L. Flores-Rios, Universidad Autónoma de Baja California, MX
Brígida Mónica Faria, ESTSP, Instituto Politécnico do Porto, PT
Carlos Costa, Universidade de Aveiro, PT
Carlos Rabadão, Instituto Politécnico de Leiria, PT
Carlos Carreto, Instituto Politécnico da Guarda, PT
Carlos Morais, Instituto Politécnico de Bragança, PT
Carlos Vaz de Carvalho, Instituto Politécnico do Porto, PT
Carmen Galvez, Universidad de Granada, ES
César Gonçalves, Universidade do Algarve, PT
Ciro Martins, Universidade de Aveiro, PT
Cristina Alcaraz, Universidad de Málaga, ES
Daniel Castro Silva, Universidade de Coimbra, PT
Daniel Polónia, Universidade de Aveiro, PT
Daniel Riesco, Universidad Nacional de San Luis, AR
David Fonseca, Universitat Ramon Llull, ES
David Ramos Valcarcel, Universidad de Vigo, ES
Dora Simões, Universidade de Aveiro, PT
Eduardo Sánchez Vila, Universidad de Santiago de Compostela, ES
Emiliano Reynares, CIDISI - UTN FRSF - CONICET, AR
Enric Mor, Universitat Oberta de Catalunya, ES

Eusébio Ferreira da Costa, Escola Superior de Tecnologias de Fafe, PT
Feliz Gouveia, Universidade Fernando Pessoa, PT
Fernando Bandeira, Universidade Fernando Pessoa, PT
Fernando Diaz, Universidad de Valladolid, ES
Fernando Moreira, Universidade Portucalense, PT
Francisco Restivo, Universidade Católica Portuguesa, PT
Gerardo Gonzalez Filgueira, Universidad da Coruña, ES
Gerardo Rodriguez, Universidad de Salamanca, ES
Germano Montejano, Universidad Nacional de San Luis, AR
Guilhermina Lobato Miranda, Universidade de Lisboa, PT
Hélder Zagalo, Universidade de Aveiro, PT
Hélia Guerra, Universidade dos Açores, PT
Henrique Gil, Instituto Politécnico de Castelo Branco, PT
Henrique Santos, Universidade do Minho, PT
Higino Ramos, Universidad de Salamanca, ES
Hugo Paredes, Universidade de Trás-os-Montes e Alto Douro, PT
Isabel Pedrosa, Instituto Politécnico de Coimbra, PT
Isaura Ribeiro, Universidade dos Açores, PT
Isidro Calvo, Universidad del País Vasco (UPV/EHU), ES
Ismael Etxeberria-Agiriano, Universidad del País Vasco (UPV/EHU), ES
Ivan Garcia, Universidad Tecnológica de la Mixteca, MX
Javier Garcia Tobio, CESGA-Centro de Supercomputacion de Galicia, ES
Jezreel Mejia, Centro de Investigación en Matemática (CIMAT), MX
João Pascual Faria, FEUP, Universidade do Porto, PT
João Paulo Costa, Universidade de Coimbra, PT
João Tavares, FEUP, Universidade do Porto, PT
Joaquim José Gonçalves, Instituto Politécnico do Cávado e do Ave, PT
Joaquim Madeira, Universidade de Aveiro, PT
Joaquim Reis, ISCTE - Instituto Universitário de Lisboa, PT
Jörg Thomaschewski, University of Applied Sciences OOW - Emden, DE
Jorge Dias Villegas, Universidad de la Frontera, CL
Jose Alfonso Aguilar, Universidad Autonoma de Sinaloa, MX
José Augusto Fabri, Universidade Tecnológica Federal do Paraná, BR
José Braga de Vasconcelos, Universidade Atlântica, PT

José Cascalho, Universidade dos Açores, PT
José Felipe Cocón Juárez, Universidad Autónoma del Carmen, MX
Jose J. Pazos-Arias, Universidad de Vigo, ES
José Luís Silva, Universidade da Madeira, PT
José Paulo Lousado, Instituto Politécnico de Viseu, PT
José Luis Pestrana Brincones, Universidad de Málaga
José Luís Reis, ISMAI - Instituto Superior da Maia, PT
Jose M Molina, Universidad Carlos III de Madrid, ES
Jose Maria Zavala Perez, Eticas Research & Consulting, ES
José Martins, Universidade de Trás-os-Montes e Alto Douro, PT
Jose Maria de Fuentes, Universidad Carlos III de Madrid, ES
Jose R. R. Viqueira, Universidade de Santiago de Compostela, ES
José Silvestre Silva, Academia Militar, PT
Josep M. Marco-Simó, Universitat Oberta de Catalunya, ES
Juan D'Amato, PLADEMA-UNCPBA-CONICET, AR
Juan M. Santos Gago, Universidad de Vigo, ES
Juan Manuel Fernández-Luna, Universidad de Granada, ES
Juan-Manuel Lopez-Zafra, Universidad Complutense de Madrid, ES
Leonardo Bermon, Universidad Nacional de Colombia, CO
Leila Weitzel, Universidade Federal Fluminense, BR
Lilia Muñoz, Universidad Tecnológica de Panamá, PA
Luis Alvarez Sabucedo, Universidad de Vigo, ES
Luís Correia, Universidade de Lisboa, PT
Luis de Campos, Universidad de Granada, ES
Luis Enrique, Sicaman Nuevas Tecnologías S.L., ES
Luis Fernandez-Sanz, Universidad de Alcalá, ES
Luís Ferreira, Instituto Politécnico do Cávado e do Ave, PT
Luis Vilán-Crespo, Universidad de Vigo, ES
Luisa María Romero-Moreno, Universidad de Sevilla, ES
Luisa Miranda, Instituto Politécnico de Bragança, PT
Lus Sussy Bayona Ore, Universidad Nacional Mayor de San Marcos, PE
Magdalena Arcilla Cobián, Universidade Nacional de Educación a Distancia, ES
Manuel Jose Fernandez Iglesias, Universidad de Vigo, ES
Marcelo Mendonça Teixeira, Universidade Federal Rural de Pernambuco, BR

Marco Painho, ISEGI, Universidade Nova de Lisboa, PT
Maria Hallo, Escuela Politécnica Nacional, EC
María J. Lado, Universidad de Vigo, ES
Maria João Castro, Instituto Politécnico do Porto, PT
Maria João Ferreira, Universidade Portucalense, PT
Maria João Gomes, Universidade do Minho, PT
Maria José Angélico, Instituto Politécnico do Porto, PT
Maria José Marcelino, Universidade de Coimbra, PT
Maria José Sousa, Universidade Europeia, PT
Marisol B. Correia, Universidade do Algarve, PT
Maristela Holanda, Universidade de Brasília, BR
Martín Llamas Nistal, Universidad de Vigo, ES
Matías García Rivera, Universidad de Vigo, ES
Mercedes Ruiz, Universidad de Cádiz, ES
Miguel A. Brito, Universidade do Minho, PT
Miguel Bugalho, Universidade Europeia, PT
Miguel Casquilho, IST, Universidade de Lisboa, PT
Mirna Ariadna Muñoz Mata, Centro de Investigación en Matemáticas (CIMAT), MX
Nelson Rocha, Universidade de Aveiro, PT
Nuno Lau, Universidade de Aveiro, PT
Nuno Ribeiro, Universidade Fernando Pessoa, PT
Orlando Belo, Universidade do Minho, PT
Oscar Mealha, Universidade de Aveiro, PT
Paula Peres, Instituto Politécnico do Porto
Paula Prata, Universidade da Beira Interior, PT
Paulo Martins, Universidade de Trás-os-Montes e Alto Douro, PT
Paulo Pinto, FCT, Universidade Nova de Lisboa, PT
Pedro Abreu, Universidade de Coimbra, PT
Pedro Miguel Moreira, Instituto Politécnico de Viana do Castelo, PT
Pedro Nogueira Ramos, ISCTE - Instituto Universitário de Lisboa, PT
Pedro Sánchez Palma, Universidad Politécnica de Cartagena, ES
Pedro Sanz Angulo, Universidad de Valladolid, ES
Pilar Mareca Lopez, Universidad Politécnica de Madrid, ES
Raul Laureano, ISCTE - Instituto Universitário de Lisboa, PT

Renata Spolon Lobato, UNESP - Universidade Estadual Paulista, BR
Reinaldo Bianchi, Centro Universitário da FEI, BR
Rita Santos, Universidade de Aveiro, PT
Roberto Rodríguez Echeverría, Universidad de extremadura, ES
Rodolfo Miranda Barros, Universidade Estadual de Londrina, BR
Rogério Eduardo Garcia, Universidade Estadual Paulista, BR
Rubén González Crespo, Universidad Internacional de La Rioja, ES
Rui Cruz, IST, Universidade de Lisboa, PT
Rui José, Universidade do Minho, PT
Rui Pedro Marques, Universidade de Aveiro, PT
Santiago Gonzales Sánchez, Universidad Inca Garcilaso de la Vega, PE
Sergio Gálvez Rojas, Universidad de Málaga, ES
Sérgio Guerreiro, Universidade Lusófona de Humanidades e Tecnologias, PT
Silvia Fernandes, Universidade do Algarve, PT
Solange N Alves de Souza, Universidade de São Paulo, BR
Tomás San Feliu Gilabert, Universidad Politécnica de Madrid, ES
Valéria Farinazzo Martins, Universidade Presbiteriana Mackenzie, BR
Victor Hugo Medina Garcia, Universidad Distrital Francisco José de Caldas, CO
Vitor Carvalho, Instituto Politécnico do Cávado e do Ave, PT
Vitor Santos, ISEGI, Universidade Nova de Lisboa, PT
Wagner Tanaka Botelho, Universidade Federal do ABC, BR
Xose A. Vila, Universidad de Vigo, ES

Tendencias en Tecnologías de Información y Comunicación

Trends in Information and Communication Technologies

Jezreel Mejia¹, Mirna Muñoz¹, Tomas San Feliu²

{jmejia, mirna.munoz }@cimat.mx, tomas.sanfeliu@upm.es

¹Centro de Investigación en Matemáticas- Unidad zacatecas, Parque Quantum, Ciudad del Conocimiento, Avenida Lasec, andador Galileo Galilei, manzana 3 lote 7, C. P. 98160, Zacatecas, México

²ETS de Ingenieros Informáticos :: Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte (Madrid), España

DOI: 10.4304/risti.35.0

Introducción

Los artículos presentados en este número especial fueron seleccionados de los artículos aceptados en el Sexto Congreso Internacional de Mejora de Procesos Software (CIMPS) 2017, realizado en la Ciudad de Zacatecas, Zacatecas, México del 18 al 20 de Octubre del 2017. Este congreso ha tenido una tasa de aceptación del 30% de los artículos enviados.

En este número especial de la Revista Ibérica de Sistemas y Tecnologías de la Información (RISTI), aborda temas relacionados a “Tendencias en Tecnologías de Información y Comunicación”. Los artículos presentan versiones extendidas y mejoradas de las versiones originales que han sido presentados en el congreso CIMPS 2017.

Tendencias en Tecnologías de Información y Comunicación

Las Tecnologías de Información y Comunicación (TICs) evolucionan a una velocidad imparable, tal es el hecho, en el cual personas como organizaciones hacen uso de éstas, demandando cada vez más su utilización en cualquier aspecto de la vida cotidiana, esto trae consigo una constante investigación en áreas a fines a las TICs, así como la creación de nuevos dominios de interés.

En el contexto de la proliferación y utilización de diversos modelos o estándares para la implementación de mejora de procesos implica esfuerzo, tiempo y costo, debido, a que cada uno de ellos contiene su propia estructura, provocando que cada implementación sea muy compleja. En este sentido, tanto en modelos o estándares orientados a grandes empresas como a pequeñas o muy pequeñas como es el estándar ISO/IEC 29110 (Cano et al, 2015), se deben tener en cuenta factores existentes como son: tamaño de la organización, cultura organizacional, uso correcto del estándar o modelo, compromiso de la alta gerencia, participación y compromiso del equipo (Davila & Pessoa, 2015). Además en modelos más robustos como CMMI-Dev v1.3, informes generados indican que las organizaciones siguen enfrentando problemas para la mejora de procesos de software y aumentan el riesgo de fracaso en pequeñas y medianas organizaciones (Larrucea et al, 2016).

En este mismo sentido, surgen nuevos modelos o paradigmas dada la necesidad de enfocar y reforzar la cooperación entre el profesional de TI y los desarrolladores de software lo que se conoce como “DevOps”, que es una mezcla de dos palabras, desarrolladores y Operaciones (Virmani, 2015).

En cuanto al desarrollo de proyectos de software, además de tener en cuenta aspectos como calendario, presupuesto, es crucial que se defina un modelo durante la fase de análisis de requerimientos, para ello, es recomendable utilizar patrones. Sin embargo, una mala definición del patrón establecido genera los llamado bad smells que refleja un diseño pobre que impacta negativamente en las propiedades del sistema de software, los cuales pueden ocurrir desde el código fuente (Source Makings, Code Smells 2017) a la arquitectura misma del sistema.

Otro aspecto importante durante el desarrollo de software es evaluar la efectividad, eficiencia y satisfacción del software, para ello, se han desarrollado métodos de evaluación. Estos métodos evalúan características específicas dependiendo de las aplicaciones, tales como: Web (Saltiveri, 2013), móviles (Molano, 2015) y colaborativas (Quiñonez, 2016).

Con respecto al análisis de datos, hoy en día las organizaciones se enfrentan al crecimiento constante de información enfrentándose en que tipo de información es prioritaria en las organizaciones de cualquier entorno, además de como gestionar esta gran cantidad de datos. Como respuesta a ello, uno de los procesos que se esta utilizando es un proceso de Data warehousing que permite un análisis de datos existentes dentro de cualquier organización, ya que uno de sus objetivos es la creación de conocimiento a través de un conjunto de estrategias y herramientas tecnológicas (Golfarelli et al, 2004; Prieto & Piattini, 2015).

Finalmente, otro ámbito donde es prioritaria la utilización de análisis de datos es en las llamadas ciudades inteligentes, donde las ciudades y los territorios se consideran elemento clave de cualquier país (Nam & Pardo, 2011). Otro aspecto importante en el conocimiento generado a partir del análisis de información es la capacidad de absorción en la teoría del aprendizaje organizacional (Levitt & March, 1988), es decir, con el objetivo de encontrar una ventaja competitiva en ellas.

Los artículos publicados en este número de RISTI tratan de cubrir parte de las Tendencias en Tecnologías de Información y Comunicación que actualmente están siendo abordadas desde el punto de vista de Ingeniería del Software.

Estructura

El primer artículo, presenta un estudio que analiza los factores del entorno que influyen en la adopción del estándar ISO/IEC 29110, en su perfil básico, en 4 pequeñas empresas desarrolladoras de software, utilizando la metodología de estudio de caso múltiple. Del conjunto de factores del entorno analizados, se encontró como resultado que 5 de ellos influenciaron positivamente en todas las empresas sin importar el contexto, 2 ellos negativamente y 2 de ellos tuvieron relación entre factores debido al contexto.

El segundo artículo, estudia los factores relacionados con características del software, la organización y los usuarios para la aceptación del software de gestión ya que el uso de aplicaciones de software generalmente en su adopción no siempre es efectiva. En particular, en la gestión de proyectos, donde existen algunas dificultades en el contexto de las empresas medianas en Lima para utilizar las herramientas de gestión de proyectos.

El tercer artículo, propone la identificación de alternativas de caminos a través de una herramienta para lograr una efectiva implementación en el modelo CMMI-DEV Nivel 2, considerando sus objetivos de negocio de la organización, ya que en la actualidad, uno de los principales obstáculos de las organizaciones de desarrollo de software en la implementación del Modelo CMMI-Dev es como iniciar su implementación y aumentar la probabilidad de éxito en una evaluación SCAMPI.

El cuarto artículo, muestra la implementación de un enfoque DevOps (Developer Operations) en Centros de Datos que permite establecer estrategias para la gestión de operaciones automatizadas, ingeniería de software y aseguramiento de la calidad. Este artículo presenta la implementación de un enfoque evolucionado de DevOps nombrado como DevSecOps+ Risk Management, que además, permite establecer estrategias para gestionar la seguridad informática y gestión de riesgos.

En el quinto artículo presenta en análisis de bad smells arquitectónicos que son causa frecuente de acumulación de deuda técnica; término que se refiere al costo en que se incurre por utilizar un enfoque de diseño o desarrollo apresurado y descuidado. Los resultados obtenidos en este artículo, muestran que la mayoría de los bad smells definidos existen en la práctica, y que la estrategia de detección propuesta por los autores reduce en un amplio margen el tiempo requerido para detectar bad smells mediante una revisión de código de forma manual.

El sexto artículo, presenta una propuesta de evaluación de usabilidad para Aplicaciones Colaborativas (AC) con interfaces de usuario multimodal, aplicada a los cuatro elementos de una AC: comunicación, colaboración, coordinación y regulación. La propuesta contempla la evaluación en tres etapas: planeación, ejecución y análisis y reporte. La propuesta de evaluación se aplicó a una AC con interfaz multimodal que permite la elaboración de diagramas de flujo.

El séptimo artículo, presenta una nueva metodología para procesos data warehousing, que integra la especificación de requisitos de información, modelamiento relacional, modelo de desarrollo combinado a partir de las propuestas de Kimball y Hefesto, un proceso aumentado de extracción-transformación y carga que incorpora explícitamente una fase de validación de indicadores, y finalmente visualizaciones integradas e interactivas para el análisis multidimensional de los indicadores obtenidos.

En el octavo artículo, presenta un marco acuñado como SmartLand-LD con un enfoque de Web Semántica que apoya la extracción, la transformación de datos vinculados, la integración y la explotación de datos masivos y heterogéneos, con aplicación en la toma de decisiones relacionadas con el desarrollo inteligente y sostenible de los territorios. Este marco permite la construcción de gráficos de conocimiento, la interoperabilidad semántica y las técnicas colaborativas de ciencia abierta, por lo que, permite el mapeo e integración de datos sobre indicadores de desarrollo sostenible en territorios de alta biodiversidad.

Finalmente, en el noveno artículo, se presenta una re-conceptualización del constructo ‘capacidad de absorción’ (ACAP) en las organizaciones para facilitar la simulación de los efectos de la adquisición, asimilación y explotación de conocimiento aprovechable del entorno de la organización, con el objetivo de generar ventaja competitiva e innovación en las organizaciones.

Agradecimientos

Agradecemos al Director de la Revista Álvaro Rocha por otorgar la oportunidad para dar a conocer los trabajos mas relevantes presentados en el congreso CIMPS 2017. Asimismo, expresamos nuestra gratitud a los autores de los trabajos publicados, lo cual hizo posible este número especial en RISTI. Damos la enhorabuena a todos aquellos autores cuyos artículos han sido publicados en este número especial de RISTI.

Referencias

- Cano, C., Melgar, A., Dávila, A., & Pessoa, M. (2015). Comparison of Software Process Models. A Systematic Literature Review. In XLI Conferencia Latinoamericana en Informática (CLEI 2015) (pp. 19–23). Arequipa.
- Dávila, A., & Pessoa, M. (2015). Factors driving the adoption of ISO/IEC 29110: A case study of a small software enterprise. In Proceedings - 2015 41st Latin American Computing Conference, CLEI 2015 (pp. 1–8). <https://doi.org/10.1109/CLEI.2015.7360042>
- Larrucea, X., O'Connor, R. V., Colomo-Palacios, R., & Laporte, C. Y. (2016). Software process improvement in very small organizations. IEEE Software, 33(2), 85–89. <https://doi.org/10.1109/MS.2016.37>
- Virmani M. (2015). Understanding DevOps & Bridging the gap from continuous integration to continuous delivery. In Fifth international conference on Innovative Computing Technology (INTECH 2015). pp. 78–82
- “Source Makings, Code Smells.” (2017). <https://sourcemaking.com/refactoring/>
- Saltiveri, T. G., Fransi, E. C., & Salla, Y. M.. (2013). Análisis de usabilidad de cooperativas del sector de la fruta y aceite en el área de Lleida. RISTI - Revista Ibérica de Sistemas e Tecnologías de Informação, 11, 45–59.

- Molano, J. I. R., Yara, E. S., & García, L. K. J. (2015). Model for measuring usability of survey mobile apps, by analysis of usability evaluation methods and attributes. In *Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on* (pp. 1–6). IEEE.
- Quiñonez, Y., Luzardo, G. & Granda, R. (2016). Implementación de un sistema multi-táctil en ambientes educativos para promover y facilitar la evaluación del trabajo colaborativo en el aula. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, 17, 57–65.
- Golfarelli, M., Rizzi, S. & Cella, I. (2004). Beyond Data Warehousing: What's Next in Business Intelligence. In: *Proceedings of the 7th ACM international workshop on data warehousing and OLAP*, pp. 1-6. Washington DC, USA, ISBN 978-0-07-067752-4.
- Prieto, A. & Piattini, M. (2015). Propuesta de marco de mejora continua de gobierno TI en entidades financieras. *Revista ibérica de Sistemas y Tecnologías de Información RISTI*, N.º 15, 06/2015, pp. 51–67. DOI: 10.17013/risti.15.51-67.
- Nam, T., & Pardo, T. A. (2011). Conceptualizing smart city with dimensions of technology, people, and institutions. In *Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times* (pp. 282–291).
- Levitt, B., & March, J. G. (1988). Organizational learning. *Annual review of sociology*, 319–340.

Índice / Index

EDITORIAL

Tendencias en Tecnologías de Información y Comunicación	ix
<i>Jezreel Mejía, Mirna Muñoz, Tomas San Feliu</i>	

ARTIGOS / ARTICLES

Influencia de Factores de Entorno en la Implementación del estándar ISO/IEC 29110.....	1
<i>Stuardo Lucho, Karin Melendez, Abraham Dávila</i>	
Impacto de los factores sociológicos de los usuarios en la aceptación y uso de software de gestión de proyectos en la mediana empresa en Lima.....	17
<i>César Aguilera, María Teresa Villalobos, Abraham Dávila</i>	
Identificando caminos para iniciar la implementación de mejora de procesos de software en organizaciones de desarrollo de software	31
<i>Jezreel Mejía, Marcos Gonzalez, Jose Calvo-Manzano, Tomas San Feliu</i>	
Implementación de un enfoque DevSecOps + Risk Management en un Centro de Datos de una organización Mexicana	43
<i>Oswaldo Díaz, Mirna Muñoz</i>	
Caracterización y Detección Automática de Bad Smells MVC	54
<i>Perla Velasco-Elizondo, Lucero Castañeda-Calvillo, Alejandro García-Fernández, y Sodel Vazquez-Reyes</i>	
Proceso de evaluación de usabilidad en aplicaciones colaborativas con interfaces de usuario multimodal.....	68
<i>Gabriela Sánchez Morales, Carmen Mezura-Godoy, Itzel Alessandra Reyes Flores, Edgard Benítez-Guerrero</i>	
Una Metodología para Procesos Data Warehousing Basada en la Experiencia.....	83
<i>Wilson Castillo-Rojas, Fernando Medina Quispe, Francisco Fariña Molina</i>	
Hacia la Interoperabilidad Semántica para el Manejo Inteligente y Sostenible de Territorios de Alta Biodiversidad usando SmartLand-LD.....	104
<i>Nelson Piedra, Juan Pablo Suárez</i>	
Un modelo basado en agentes para simular la capacidad de absorción en organizaciones	122
<i>Orlando López-Cruz</i>	

Influencia de Factores de Entorno en la Implementación del estándar ISO/IEC 29110

Stuardo Lucho¹, Karin Melendez², Abraham Dávila³

stuardo.lucho@pucp.edu.pe, kmelendez@pucp.edu.pe, abraham.davila@pucp.edu.pe

¹ Departamento de Ingeniería, Pontificia Universidad Católica del Perú, Lima 32, Lima, Perú.

² Departamento de Ingeniería, Pontificia Universidad Católica del Perú, Lima 32, Lima, Perú.

³ Departamento de Ingeniería, Pontificia Universidad Católica del Perú, Lima 32, Lima, Perú

DOI: 10.17013/risti.26.1-16

Resumen: El estándar ISO/IEC 29110 es usado por algunas empresas peruanas con el fin de mejorar la calidad del software que elaboran y ser más productivos. Sin embargo, existen factores que influyen en el equipo durante la mejora de procesos de software, por lo cual, identificar adecuadamente estos factores, permitirá mejorar el nivel de adopción en futuras implementaciones. El objetivo de este estudio es analizar los factores del entorno que influyen en la adopción del estándar ISO/IEC 29110, en su perfil básico, en 4 pequeñas empresas desarrolladoras de software, utilizando la metodología de estudio de caso múltiple. Cada empresa trabajó bajo su propio contexto y definió sus propios objetivos de mejora. Del conjunto de factores del entorno analizados, se encontró como resultado que 5 de ellos influenciaron positivamente en todas las empresas sin importar el contexto, 2 ellos negativamente y 2 de ellos tuvieron relación entre factores debido al contexto.

Palabras-clave: ISO/IEC 29110, factores de adopción, mejora de proceso software.

Influence of Environment Factors in the Implementation of the ISO / IEC 29110 standard

Abstract: The ISO/IEC 29110 standard is used by some Peruvian companies in order to improve the quality of the software they produce and be more productive. However, there are factors that influence the team during the software process improvement, for which, properly identifying these factors will allow improving the level of adoption in future implementations. The purpose of this study is to analyze the environmental factors that influence the adoption of the ISO/IEC 29110 standard, in its basic profile, in 4 small software development companies, using the multiple case study methodology. Each company worked in its own context and defined its own improvement objectives. From the set of environmental factors analyzed, it was found that 5 of them positively influenced all the companies regardless of the context, 2 of them negatively and 2 of them were related between factors due to the context.

Keywords: 29110, adoption, factors, spi.

1. Introducción

Las empresas en la industria del software realizan la definición y documentación de sus actividades para: (i) gestionar sus proyectos de software de inicio a fin (Humphrey, 1995), y (ii) para proporcionar una comprensión general a todos aquellos involucrados en el desarrollo de software sobre las tareas de las cuales son responsables (Unterkalmsteiner et al., 2012). Estas actividades pueden mejorarse siguiendo diversos referentes internacionales como estándares o modelos de procesos (Vasconcellos et al., 2016) (Miramontes, Muñoz, Calvo-Manzano, & Corona, 2016). Algunos modelos requieren mayor cantidad de recursos humanos y económicos para poder implementarlos, como el modelo CMMI o el estándar ISO/IEC 15504 (Huang & Zhang, 2010); mientras que otros, como MoProSoft o el estándar ISO/IEC 29110, están más orientados a las pequeñas empresas artículos (O'Connor & Laporte, 2011)(Cano, Melgar, Dávila, & Pessoa, 2015).

Al adoptar un modelo de procesos, existen factores que se deben considerar para obtener beneficios o evitar retrasos, tales como el tamaño de la empresa (Horvat, Rozman, & Györkös, 2000) (Basri & O'Connor, 2010b), la cultura organizacional, los recursos a invertir, los resultados esperados (Ebert, 1999)(Larrucea, O'Connor, Colomo-Palacios, & Laporte, 2010), el uso correcto de un modelo de referencia, la participación de la alta gerencia de la organización; y especialmente la alta proporción de personas que forman parte del equipo de trabajo de la organización (Basri & O'Connor, 2010a). Todos estos factores han sido estudiados y organizados por diversos autores en categorías tales como: factores organizacionales, alta dirección y del entorno, entre otros; mientras que otros autores los han agrupado en modelos conceptuales, como los presentados en (Hameed, Counsell, & Swift, 2012), (Bayona, Calvo-Manzano, & Feliu, 2013), (Bayona, Calvo-Manzano, Cuevas, & Feliu, 2010) y (Sulayman, Urquhart, Mendes, & Seidel, 2012).

Con respecto a lo indicado, en el proyecto ProCal-ProSer (Díaz, De Jesús, Melendez, & Dávila, 2016) se estudia los factores que influyen en la adopción del estándar ISO/IEC 29110 por parte de pequeñas empresas. Los factores identificados como parte del proyecto ProCal-ProSer permitirán definir estrategias de adopción en la industria del software.

En este artículo se presenta una análisis complementario al presentado en un artículo previo (Lucho, Melendez, & Dávila, 2018) sobre los factores del entorno que influyen en un grupo de empresas que implementaron el estándar ISO/IEC 29110. El documento está organizado de la siguiente manera: Sección 2, muestra el marco de referencia; Sección 3, metodología de investigación; Sección 4, unidades de análisis; Sección 5, el análisis e interpretación de los resultados y la Sección 6, la discusión final y el trabajo futuro.

2. Marco de referencia

En este capítulo se revisaron conceptos y estudios relacionados con modelos de procesos y factores de mejora. Además, se realizó una búsqueda de trabajos similares que permitan guiarnos y realizar una investigación más completa.

2.1. Modelos de proceso de software

Existen diversas maneras de mejorar la calidad del producto de software, tal como se indica en los estudios (Nirmala & LathaMaheswari, 2016), (Svensson et al., 2012)

y (Razón, Durán, González, Cortérighs, & Miranda, 2014). Mejorar la calidad en el proceso de desarrollo de software (Elhag, Elshaikh, Mohamed, & Babar, 2013) basado en modelos de procesos de software, es una forma de lograr mejorar la calidad en el producto.

Entre los modelos más representativos se encuentran: CMMI (SEI, 2010) y el estándar ISO/IEC 12207 (ISO/IEC, 2008). Sin embargo, ambos modelos tienen problemas de adopción y generan rechazo cuando se intenta implementar en pequeñas organizaciones (Huang & Zhang, 2010), (Brodman & Johnson, 1997), (Ruiz et al., 2011). Dada la situación descrita, se desarrollaron nuevos modelos que se ajustan mejor a las pequeñas empresas y algunos de ellos fueron utilizados en Perú: MoProSoft (Oktaba et al., 2005), Competissoft (Piattini, 2008), MPS.BR (Kalinowski et al., 2015); y, el estándar ISO/IEC 29110 (ISO/IEC, 2011a).

De los modelos mencionados, la ISO/IEC 29110 es un estándar desarrollado con el objetivo de mejorar la calidad de los productos de software a través de la mejora de los procesos de las pequeñas empresas (Laporte, O'Connor, & Paucar, 2016). El estándar define una pequeña organización (PO) como una organización, empresa, departamento o proyecto que tiene hasta 25 profesionales dedicados a proyectos de software (ISO/IEC, 2011a). El estándar está organizado por perfiles tales como: básico, intermedio y avanzado; y en cada perfil se establece un conjunto de procesos (ISO/IEC, 2011a). El perfil básico se detalla en el documento (ISO/IEC, 2011c) y está compuesto por dos procesos: (i) gestión de proyecto (GP), el cual tiene como objetivo llevar a cabo sistemáticamente las tareas del desarrollo del software, con el propósito de cumplir los objetivos del proyecto en relación con calidad, tiempo y presupuesto. (ii) Implementación del software (IS), el cual tiene como objetivo establecer la relación sistemática entre las actividades de análisis, construcción, diseño, integración y prueba del producto software según los requisitos de software.

Cada proceso y sus principales actividades se muestran en la Figura 1 y la Figura 2.

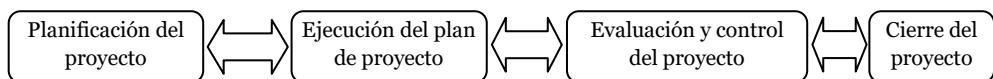


Figura 1 – Actividades principales del proceso de GP (ISO/IEC, 2011c)



Figura 2 – Actividades principales del proceso de IS (ISO/IEC, 2011c)

2.2. Modelos de factores en la mejora de procesos de software

Las empresas que adoptan una mejora de procesos de software (MPS), se dice que puede considerarse como una innovación en tecnologías de la información (Anaconda, Bastidas, Pino, & Pardo, 2015).

En este contexto, existen trabajos de investigación para pequeñas organizaciones (PO) como (Mejia Jezreel et al., 2012) y (M Jezreel et al., 2013), donde se proponen métodos para identificar los factores clave a considerar si la empresa desea implementar una mejora de procesos de software exitosa. Algunos factores identificados fueron: mantenimiento del software, calidad del producto y gestión del cambio.

Del mismo modo, en la literatura se han recopilado un conjunto de factores que pueden influir positiva o negativamente al realizar una MPS. Estos estudios incluyen: (i) El análisis realizado en (Cuevas, Calvo-Manzano, & García, 2014) que muestra los puntos clave que deben tenerse en cuenta en las diferentes fases del desarrollo del software y en las personas que implementan el MPS. (ii) Los 18 factores de éxito identificados para lograr una mejora de procesos de software, en pequeñas y medianas empresas paquistaníes, descritas en (Sulayman, Mendes, Urquhart, Riaz, & Tempero, 2014), en el contexto de aplicaciones de desarrollo web. (iii) El estudio de (Lee, Shiue, & Chen, 2016) que analiza la influencia de otros factores como la cultura organizacional, la distribución del conocimiento entre los empleados, entre otros. (iv) Los inconvenientes asociados con la adopción de una metodología ágil descrita en (Paulk, 2011), donde se identifican un conjunto de factores tales como: el tamaño del proyecto, problemas culturales, entre otros. (v) El modelo conceptual de (Hameed et al., 2012), donde se recogieron 107 factores de la literatura actual, que pueden influir en la adopción de una innovación en tecnología de la información. Los factores se agruparon en 5 categorías: innovación (20 factores), organización (41 factores), entorno (16 factores), CEO (8 factores) y aceptación del usuario (22 factores).

2.3. Trabajos relacionados

Una de las ventajas de utilizar el estándar ISO/IEC 29110 es que permite a las empresas mejorar sus procesos de software sin tener la complejidad de otros estándares o modelos (Larrucea et al., 2010); en esta línea, hay estudios que analizan los potenciales beneficios y barreras de las pequeñas empresas en adoptar el estándar en un futuro, tales como (Basri & O'Connor, 2010a) y (Sánchez-Gordón & O'Connor, 2016). Además, en los artículos (O'Connor & Laporte, 2011), (O'Connor & Sanders, 2013), (Pauca, Laporte, Arteaga, & Bruggmann, 2015) y (O'Connor, 2014) se estudian algunas buenas prácticas y directrices a tomar en cuenta cuando una pequeña organización adopta el estándar ISO/IEC 29110, tales como: "Paquetes de implementación", asistencia cercana al equipo para la implementación del estándar y el uso de herramientas de soporte para la gestión de procesos de IS y GP. Todos los artículos mencionados previamente, en esta sección, se consideran de gran relevancia pues contribuyen a comprender los factores que se analizan en el contexto de nuestro Proyecto ProCal-ProSer (Díaz et al., 2016).

Por otro lado, hay otro grupo de artículos que analizan los factores que influyen en las pequeñas empresas, al implementar una mejora de procesos de software utilizando el estándar ISO/IEC 29110. Uno de ellos (Wongsai, Siddoo, & Wetprasit, 2015), recopila y clasifica 37 factores (de éxito y barreras de adopción) que pueden influenciar en la adopción del estándar ISO/IEC 29110. Nuestra investigación, a diferencia de la anterior, utiliza otro grupo de factores identificados en (Hameed et al., 2012) no necesariamente aplicados al estándar ISO/IEC 29110; sin embargo, en la sección de discusión, se analiza si se encontró alguna relación entre los factores identificados en (Wongsai et al., 2015) y los analizados en este estudio.

Finalmente, los artículos de (Sanchez-Gordon, O'Connor, & Colomo-Palacios, 2015) y (O'Connor, 2012) muestran el punto de vista de algunas PO con respecto a la adopción del estándar ISO/IEC 29110, la información obtenida a través del uso de la metodología de teoría fundamentada y entrevistas. Como resultado, se muestran factores que influyen en la adopción tales como: nivel de interés, nivel de aceptación y barreras presentes en una pequeña empresa para adoptar una mejora de proceso de software usando el estándar ISO/IEC 29110. Nuestra investigación tiene similitud con ambos estudios; sin embargo, el propósito de este estudio es conocer qué factores tienen una mayor o menor influencia en la mejora de los procesos de software (basado en el modelo conceptual de (Hameed et al., 2012)), a diferencia de los estudios previos que obtienen los factores como resultado de la metodología utilizada.

3. Metodología de investigación

En el presente estudio, se ha utilizado la metodología de investigación cualitativa “estudio de caso múltiple”, debido a que se realiza un análisis exploratorio de la mejora de procesos en diferentes organizaciones, pero cada una como unidad de análisis independiente.

El protocolo que se ha seguido para el estudio de caso es el definido en (Genero, Cruz-Lemus, & Piattini, 2014), asimismo, se tiene como referencia el artículo de (Dávila & Pessoa, 2015), que muestra un caso práctico de implementación del protocolo, en un análisis de factores de adopción en una compañía. El proceso seguido consta de 4 fases y varias etapas (ver Tabla 1).

Fase	Etapas
<i>P1: Diseño del estudio de caso</i>	(P1-1) Objetivo del estudio (P1-2) Qué se estudia (P1-3) Teoría requerida (P1-4) Preguntas de investigación (P1-5) Métodos de recolección (P1-6) Selección de datos
<i>P2: Preparar recolección de la data</i>	(P2-1) Definición de cómo se realiza la recolección
<i>E1: Recolectar la data</i>	(E1-1) Recolección de la data
<i>E2: Analizar e interpretar la data</i>	(E2-1) Análisis de la data (E2-2) Interpretación de la data
<i>R1: Reportar los resultados</i>	(R1-1) Llevar a cabo el informe

Tabla 1 – Fases y etapas del proceso del estudio de caso (Genero et al., 2014)

En los párrafos siguientes, se desarrollan las etapas del estudio de caso relacionado con la fase de planificación (P):

(P1-1) Objetivo del estudio: analizar cualitativamente la forma en que los factores del entorno influyen en la adopción del estándar ISO/IEC 29110. El trabajo se realiza en equipos de trabajo de cuatro PO, las cuales son descritas en la Sección 4. Para lograr el

objetivo, se realizó un ciclo de mejora del proceso de software en todas las unidades de análisis utilizando el perfil básico del estándar ISO/IEC 29110, que involucra los procesos de implementación del software y gestión de proyectos. Dentro del proyecto ProCal-ProSer (Díaz et al., 2016), se estableció trabajar con modelos de factores utilizados en el contexto de mejora de procesos y modelos de factores de innovación tecnológica, con el objetivo de evaluar su aplicación en varias PO. Para este estudio en particular, se estableció trabajar con el modelo conceptual definido en (Hameed et al., 2012), que agrupa los factores de adopción en 5 categorías como se muestra en la sección 2.2. Este artículo usa y analiza los 16 factores del entorno definidos en el modelo y se propone como trabajo futuro el estudio del resto de los factores.

(P1-2) ¿Qué se estudia? las unidades de análisis son cuatro PO. Los equipos de trabajo tienen menos de 25 personas, por lo que no hubo restricciones para implementar la mejora del proceso basada en el estándar ISO/IEC 29110 en lo que respecta al perfil básico. La descripción detallada de las empresas y sus características se presenta en la Sección 4. En consecuencia, el objeto de estudio se basa en un grupo de instancias de análisis, cada una bajo su propio contexto, lo que permite definir un estudio de caso múltiple holístico (Genero et al., 2014).

(P1-3) Teoría requerida: ver la Sección 2.

(P1-4) Preguntas de investigación: la pregunta de investigación para el presente estudio es: ¿Cómo influyen los factores del entorno en la adopción de las prácticas de perfil básico definidas en la ISO/IEC 29110-5-1-2?

(P1-5) Métodos de recolección: la información recopilada fue de tipo cualitativo y se obtuvo en dos etapas. (i) Primera etapa: por cada empresa se realizó una evaluación diagnóstica, utilizando documentos e informes, para comparar el nivel de cumplimiento de sus prácticas versus las definidas en el estándar ISO/IEC 29110-5-1-2, tanto al inicio como al final de la mejora. El proceso de recopilación de datos, para esta primera etapa, se puede considerar de tercer grado, de acuerdo con lo especificado en (Genero et al., 2014). (ii) Segunda etapa: en todas las empresas se realizaron entrevistas semiestructuradas, para lograr un estudio descriptivo y exploratorio de los factores del entorno que influyeron en la mejora de procesos de software. Las preguntas de la entrevista fueron formuladas en base a la descripción de cada factor y fueron revisadas y aprobadas por un revisor externo con más de 10 años de experiencia en cuestiones de MPS y calidad de los procesos de software. Detalles adicionales de la entrevista se muestran en la Sección 5.

(P1-6) Selección de datos: se realizó en dos etapas acorde a (P1-5). (i) Primera etapa: el informe del nivel de adopción basado en la evaluación de la ISO/IEC 15504 y realizado por un evaluador externo, el cual pudo determinar qué prácticas del estándar fueron adoptados y cuáles no, y hasta qué punto (ver Sección 5). (ii) Segunda etapa: las entrevistas fueron refinadas y consolidadas para obtener un diagnóstico de los factores relacionados al entorno que influenciaron en la adopción de la norma (ver la Sección 5).

(P2-1) Definición de cómo se realiza la recolección: se llevó a cabo en dos etapas. La primera, donde se evaluó el nivel de adopción del estándar y la segunda, a través de entrevistas semiestructuradas, cuyo foco central fue conocer qué factores del modelo de (Hameed et al., 2012) influyeron positiva o negativamente en la adopción del perfil básico.

Los pasos restantes se analizaron de la siguiente manera: (E1-1), (E2-1) y (E2-2) en la sección 5, (R1-1) en todo el artículo.

4. Unidades de análisis

Se tienen cuatro unidades de análisis para el estudio de caso múltiple. Por cuestiones de confidencialidad, las empresas se llamaron alfa, beta, gamma y delta, las cuales se describen en la Tabla 2.

Ítem	Alpha	Beta	Gamma	Delta
<i>Descripción</i>	Área de software dentro de una organización educativa. Desarrolla proyectos a ser utilizados en entornos educativos.	PO del sector privado que desarrolla soluciones de hardware, software y redes para la industria del juego.	PO que ofrece soluciones en diversos sectores, principalmente comerciales, minoristas, de salud, minería, ingeniería y educación.	PO que ofrece soluciones para iPhone, iPad, Android y Windows Phone.
<i>Equipo</i>	1 gerente de proyecto, 2 desarrolladores junior y 1 diseñador gráfico	1 gerente de proyecto, 3 desarrolladores senior y 1 junior, 2 analistas y 1 gerente de pruebas	1 gerente de proyecto, 2 analistas y 3 desarrolladores senior	1 gerente general, 1 subgerente, 4 desarrolladores senior y 2 administrativos.
<i>Experiencia empresarial</i>	5 años	13 años	8 años	10 años
<i>Experiencia en MPS</i>	Sin experiencia previa	Sin experiencia previa	Sin experiencia previa	Sin experiencia previa
<i>Apoyo externo en MPS</i>	Estudiante de posgrado	Estudiante de posgrado	Estudiante universitario (remunerado)	Estudiante universitario (remunerado)
<i>Duración de la MPS</i>	4 meses	6 meses	6 meses	6 meses

Tabla 2 – Unidades de análisis

5. Análisis y resultados

En esta sección se analizan los resultados obtenidos del nivel de adopción a la norma y los factores que influenciaron.

5.1. Análisis del nivel de adopción

El proceso de recolección de datos empezó con una evaluación diagnóstica al inicio del proyecto, en la cual se evaluaron los procesos GP y IS (del perfil básico) para conocer el estado actual de las empresas y establecer sus propuestas de mejora. La evaluación se realizó utilizando la ISO/IEC 15504-2, como se especifica en (ISO/IEC, 2011b).

La evaluación diagnóstica fue realizada por un evaluador externo calificado junto con el gerente de proyecto de cada compañía y presentando la documentación requerida por cada práctica o grupo de prácticas.

Como resultado de la evaluación inicial se obtuvo el grado de adherencia de las empresas en cada uno de los procesos implementados, en base a cuatro niveles definidos por la ISO/IEC 15504-2: N (no cumplido), P (parcialmente cumplido), L (ampliamente cumplido) y F (totalmente logrado). Asimismo, se colocaron porcentajes de adhesión a la guía; aquellos que representan el grado de cumplimiento basado en las prácticas adoptadas por la empresa. Estos porcentajes fueron colocados por los autores para facilitar la comparación y el análisis entre la evaluación inicial y la final. La calificación práctica básica del proceso de evaluación del proceso se muestra en el Apéndice A.

Las metas de mejora se establecieron en base a la evaluación diagnóstica y fueron diferentes para cada empresa: Alfa tuvo como objetivo alcanzar al menos un nivel L en ambos procesos del perfil básico; mientras que Beta pretendía alcanzar un nivel P en GP y un nivel L en IS y finalmente, Gamma y Delta apuntaban a alcanzar el nivel F en IS y GP. La implementación se llevó a cabo en todas las empresas utilizando el perfil básico (GP y IS) de la ISO/IEC 29110-5-1-2.

El proceso seguido para la evaluación inicial fue replicado para la evaluación final y los resultados se presentan en las columnas EI y EF en la Tabla 3. Se puede observar en dicha Tabla, que Alfa logró una mejora significativa en GP, pasando de una calificación N con un cumplimiento del 10.6% a una L con 62.5%.

Id Org.	Gestión de proyectos			Implementación del software		
	Adherencia (%)	Variación de nivel	Variación de capacidad	Adherencia (%)	Variación de nivel	Variación de capacidad
	EI → EF	EI → EF	EI → EF	EI → EF	EI → EF	EI → EF
Alfa	10.6 → 62.5	N → L	0 → 0	37.9 → 42.8	P → L	0 → 0
Beta	14.6 → 48.1	N → P	0 → 0	21.9 → 49.4	P → P	0 → 0
Gamma	89.2 → 96.2	F → F	1 → 1	52.9 → 71.8	L → L	0 → 0
Delta	27.9 → 84.6	P → L	0 → 1	29.9 → 77.4	P → L	0 → 0

Tabla 3 – Comparación de la evaluación inicial y final

5.2. Análisis de los factores

Después de la evaluación inicial y final, se analizaron los factores que influenciaron en la adopción a través de una entrevista semiestructurada (ver Sección 3, P1-5), la cual se realizó a los gerentes de proyecto de cada empresa. Los factores relacionados con el entorno se muestran en la Tabla 4, junto con una breve descripción. Para cada factor,

el investigador propuso una pregunta que permitiría saber durante la entrevista si el factor influyó en la mejora de procesos de software y, de ser así, en qué nivel. Estas preguntas se muestran en el Apéndice B.

ID	Factor	Descripción
E01	Presión competitiva	Presión ejercida por los competidores del mismo rubro.
E02	Presión externa	Influencias que surgen de varias fuentes dentro del entorno competitivo que rodea a la organización.
E03	Apoyo del gobierno	El gobierno apoya con incentivos financieros, reducción de impuestos, proyectos pilotos para beneficiar la innovación.
E04	Soporte del vendedor	Asistencia del vendedor de un producto o servicio con el que cuenta la empresa.
E05	Soporte de los socios	Grado de Apoyo de los socios de la empresa a las decisiones corporativas.
E06	Disposición de los socios	Disponibilidad de los socios de apoyar la mejora de procesos.
E07	Incertidumbre del entorno	La empresa tiene poca información sobre su entorno externo, por lo tanto, su comportamiento es errático.
E08	Vinculación vertical	Actividades de coordinación y comunicación desde los puestos más altos hasta los más bajos de la empresa.
E09	Defensa de los socios	Los socios de la empresa apoyan la mejora de procesos.
E10	Presión del gobierno	Presión ejercida por el gobierno hacia la adopción de estándares y la obtención de certificaciones.
E11	Número de competidores	Cantidad de competidores identificados por la empresa.
E12	Experiencia externa	La empresa solicita consultoría externa de un proveedor externo.
E13	Eficacia de la consultoría	La consultoría solicitada brindó una solución y/o ayudó a esclarecer los problemas presentes.
E14	Confianza en los socios	Confianza brindada por los socios en el proceso de adopción de la mejora de procesos.
E15	Globalización	Mayor cantidad de empresas de software en el mundo, clientes de la empresa en otros países, nuevos mercados.
E16	Influencia social	La percepción del público (u otra empresa) sobre la posición de la empresa.

Tabla 4 – Factores del entorno

Asimismo, cada factor fue evaluado de acuerdo con los siguientes criterios: (i) Influencia en la mejora del proceso: positiva o negativa. (ii) Nivel de influencia, en una escala de Likert, de 1 a 5, donde 1 es influencia muy baja y 5 una influencia muy alta. (iii) Justificación de la influencia del factor. (iv) ¿qué problemas se presentaron sobre este factor durante la mejora? (v) Criticidad del problema, en una escala Likert, de 1 a 5, donde 1 es muy baja criticidad y 5 muy alta criticidad. (vi) ¿cómo lidió con este problema?

La Tabla 5 presenta el consolidado de las entrevistas, considerando dos columnas para cada empresa: el tipo de influencia (Inf.) y el nivel de esa influencia (Nivel). Si el factor

evaluado no tuvo influencia, se coloca N.A, si la influencia fue positiva se coloca (+) y si fue negativa se coloca (-). En esta investigación utilizamos la síntesis narrativa, de acuerdo con lo establecido en (Popay et al., 2006), para analizar la información recolectada de los factores del entorno, que a través de entrevistas, fueron recopiladas y mapeadas en la Tabla 5.

Factor ID	Alfa		Beta		Gamma		Delta	
	Inf.	Nivel	Inf.	Nivel	Inf.	Nivel	Inf.	Nivel
E01	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
E02	+	4	+	3	N.A.	N.A.	N.A.	N.A.
E03	+	3	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
E04	+	1	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
E05	+	3	+	5	+	2	+	5
E06	+	3	N.A.	N.A.	+	3	+	5
E07	N.A.	N.A.	+	1	N.A.	N.A.	N.A.	N.A.
E08	+	4	+	4	+	5	+	5
E09	-	2	-	3	+	4	+	3
E10	N.A.	N.A.	+	2	+	2	N.A.	N.A.
E11	N.A.	N.A.	+	2	N.A.	N.A.	N.A.	N.A.
E12	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
E13	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
E14	+	5	+	4	+	3	+	4
E15	+	1	+	2	+	3	+	4
E16	N.A.	N.A.	N.A.	N.A.	+	5	N.A.	N.A.

Tabla 5 – Resultado de la evaluación de factores

En base a los resultados obtenidos, se analizaron los factores, de manera individual o agrupados por similitud de respuestas:

1. Los factores E01, E12 y E13, no influenciaron en ninguna empresa.
2. El factor E02, presión externa, no tuvo influencia en Gamma ni Delta, pero sí en Alfa y Beta. No se puede generalizar una razón para explicar este resultado por ambas empresas, pero sí, por separado. Alfa suele postular sus proyectos como parte de convocatorias al Estado, lo cual se ve reflejado en E03, por lo cual existe influencia competitiva del entorno en su MPS. Para Beta, la influencia viene de sus competidores, tal como se ve en relación con el factor E11, siendo esta la única PO que tuvo esa influencia.
3. El factor E03 y E04 solo influenciaron en Alfa, lo cual está acorde a su propio contexto pues es una unidad de software dentro de una entidad educativa; mientras que las otras PO, son desarrolladoras de software.
4. Los factores E05, E08, E09, E14 y E15, han influenciado en todas las empresas sin importar el contexto donde se desempeñaba. En particular, en Alfa el factor

E14 tuvo un impacto positivo muy alto y el comentario al respecto fue: “la razón es que el líder técnico es también el especialista en software y el gerente de área confió plenamente sobre sus capacidades para implementar la mejora en los procesos y obtener los resultados buscados”.

5. El único factor que influyó negativamente en el estudio fue E09 y esto sucedió en las empresas Alfa y Beta. Al analizar un extracto de la entrevista a la compañía Beta con respecto a este factor se tuvo: “El gerente de la empresa sabía sobre la mejora de los procesos, pero al no ser una persona técnica (no era ingeniero), prefirió no entrar en los detalles de la implementación de la mejora y lo dejó en manos del líder del proyecto”. Por este motivo, al no apoyar ni estar involucrado el gerente directamente la MPS, generó retrasos en la implementación de la mejora.

5.3. Amenazas del estudio

Se debe considerar que la mejora de procesos fue guiada por un miembro del equipo del proyecto ProCal-ProSer en gamma y delta, lo que no sucedió en alfa y beta. Del mismo modo, con respecto al análisis de los factores, las entrevistas se hicieron a los líderes del equipo, tal vez los miembros del equipo tenían una opinión diferente y se hubiera obtenido otros niveles de influencia para los mismos factores.

6. Discusión final y trabajo futuro

La adopción de una mejora de procesos de software implica una serie de pasos y buenas prácticas que deben seguirse para alcanzar los objetivos establecidos. En el caso de pequeñas empresas, en base a los estudios previos y los resultados obtenidos, la mejora de procesos tendrá mayor éxito si existe un compromiso total desde los cargos más altos de la empresa hasta los más bajos.

Luego de analizar los 16 factores del entorno, se puede concluir que, 5 de ellos influyeron en todas las organizaciones, independientemente del contexto en el que se encontraban. Dentro de estos cinco factores, tres de ellos, E05 soporte de los socios, E08 vinculación vertical y E14 confianza en los socios, están relacionados fuertemente con el apoyo de la alta gerencia a la mejora de procesos de software, lo cual está acorde con la investigación de (Wongsai et al., 2015) (Sección 2.2), donde dos factores mapeados, “compromiso y participación” y “política organizacional”, ambos relacionados también con el apoyo de la alta gerencia, se consideran factores de éxito. Además, estos tres factores también están relacionados con la investigación de (Sanchez-Gordon et al., 2015) donde se menciona que la “falta de necesidad percibida” y “necesidad de recursos” son barreras para lograr la adopción de una mejora de procesos. En este estudio, todas las empresas pudieron superar estas barreras pues contaban con los recursos para implementar la mejora de procesos.

Los otros dos factores que influenciaron en todas las empresas, E09 defensa de los socios y E15 globalización, no han sido identificados previamente como factores de éxito o barreras de adopción en ningún artículo revisado, por lo tanto, ambos deberían tomarse en consideración al realizar una MPS o estudios similares. De los dos antes mencionados, tomar particular atención a E09 defensa de los socios, pues fue el único que influyó negativamente en dos empresas. De igual forma, se ha podido verificar que algunos factores están relacionados con otros factores por el contexto particular de

la empresa, como es el caso de EO2, que tuvo influencia positiva en dos empresas y la razón era su relación con EO3 en Alfa y con E11 en Beta.

Como trabajo futuro, se propone realizar el procedimiento descrito en este documento, pero centrándose en los otros 4 grupos de factores del modelo de (Hameed et al., 2012): innovación, organización, CEO y aceptación del usuario. Además, se propone realizar un análisis de correlación entre el grado de adopción después de la mejora de procesos de software y los factores que han influenciado en la adopción de la mejora; así como, una posible correlación entre los mismos factores.

Agradecimientos

Este trabajo ha sido realizado dentro del proyecto ProCal-ProSer financiado por Innóvate Perú bajo el Contrato 210 - FINCYT-IA-2013 y por el Departamento de Ingeniería y el Grupo de Investigación y Desarrollo de Ingeniería de Software (GIDIS) de la Pontificia Universidad Católica del Perú.

Referencias

- Anaconda, D., Bastidas, M. I., Pino, F. J., & Pardo, C. (2015). Innova-Procedure : A procedure to guide the innovation of software development processes in VSEs. In *Proceedings of Computing Colombian Conference (10CCC) 2015 10th*, (pp. 108-115). Bogota, Colombia: IEEE.
- Basri, S., & O'Connor, R. V. (2010a). Understanding the perception of very small software companies towards the adoption of process standards. In *Communications in Computer and Information Science book* (vol. 99, (pp. 153-164). Berlin: Springer. https://doi.org/10.1007/978-3-642-15666-3_14
- Basri, S. Bin, & O'Connor, R. V. (2010b). Organizational commitment towards software process improvement: An Irish software VSEs case study. In *Proceedings of International Symposium on Information Technology - System Development and Application and Knowledge Society, ITSIM'10* (Vol. 3, pp. 1456-1461). Kuala Lumpur, Malaysia: IEEE. <https://doi.org/10.1109/ITSIM.2010.5561489>
- Bayona, S., Calvo-Manzano, J. A., & Feliu, T. S. (2013). Review of Critical Success Factors Related to People in Software Process Improvement. In *Proceedings of 20th EuroSPI Conference* (pp. 179-189). Berlin: Springer. https://doi.org/10.1007/978-3-642-39179-8_16
- Bayona, S., Calvo-Manzano, J., Cuevas, G., & Feliu, T. S. (2010). Taxonomía de Factores Críticos para el Despliegue de Procesos Software. *REICIS, Revista Española de Innovación, Calidad E Ingeniería Del Software*, 6(3), 6-22. Retrieved from <http://oa.upm.es/8242/>
- Brodman, J., & Johnson, D. (1997). A Software Process Improvement Approach Tailored for Small Organizations and Small Projects. In *Proceedings of 19th International Conference on Software Engineering* (pp. 661-662). Boston, MA, USA: ICSE. <https://doi.org/10.1109/ICSE.1997.610455>

- Cano, C., Melgar, A., Dávila, A., & Pessoa, M. (2015). Comparison of Software Process Models: A Systematic Literature Review. In *Proceedings of XLI Conferencia Latinoamericana en Informática (CLEI 2015)* (pp. 19-23). Arequipa, Peru: SPC.
- Cuevas, G., Calvo-Manzano, J. A., & García, I. (2014). *Some Key Topics to be Considered in Software Process Improvement* (pp. 119-142). <https://doi.org/10.4018/978-1-4666-5182-1.ch008>
- Dávila, A., & Pessoa, M. (2015). Factors driving the adoption of ISO/IEC 29110: A case study of a small software enterprise. In *Proceedings of 2015 41st Latin American Computing Conference, CLEI 2015* (pp. 1-8). Piscataway, New Jersey, USA: IEEE. <https://doi.org/10.1109/CLEI.2015.7360042>
- Díaz, A., De Jesús, C., Melendez, K., & Dávila, A. (2016). ISO/IEC 29110 Implementation on two Very Small Software Development Companies in Lima: Lessons Learned. *IEEE Latin America Transactions*, 14(5), 2504-2510. Retrieved from <https://sites.google.com/a/pucp.pe/procal-proser/>
- Ebert, C. (1999). Technical controlling and software process improvement. *Journal of Systems and Software*, 46, 25-39. [https://doi.org/http://dx.doi.org/10.1016/S0164-1212\(98\)10086-9](https://doi.org/http://dx.doi.org/10.1016/S0164-1212(98)10086-9)
- Elhag, A., Elshaikh, M. A., Mohamed, R., & Babar, M. I. (2013). Problems and future trends of software process improvement in some Sudanese software organizations. In *Proceedings of International Conference on Computer, Electrical and Electronics Engineering. ICCEEE 2013* (pp. 263-268). Khartoum, Sudan: IEEE. <https://doi.org/10.1109/ICCEEE.2013.6633945>
- Genero, M., Cruz-Lemus, J., & Piattini, M. (2014). *Métodos de investigación en ingeniería del software*. Madrid: RA-MA Editorial.
- Hameed, M. A., Counsell, S., & Swift, S. (2012). A conceptual model for the process of IT innovation adoption in organizations. *Journal of Engineering and Technology Management - JET-M*, 29(3), 358-390. <https://doi.org/10.1016/j.jengtecman.2012.03.007>
- Horvat, R. V., Rozman, I., & Györkös, J. (2000). Managing the complexity of SPI in small companies. *Software Process: Improvement and Practice*, 5, 45-54. [https://doi.org/10.1002/\(SICI\)1099-1670\(200003\)5:1<45::AID-SPIP110>3.0.CO;2-2](https://doi.org/10.1002/(SICI)1099-1670(200003)5:1<45::AID-SPIP110>3.0.CO;2-2)
- Huang, D., & Zhang, W. (2010). CMMI in Medium & Small Enterprises: Problems and Solutions. In *Proceedings of 2nd IEEE International Conference on Information Management and Engineering* (pp. 171-174). <https://doi.org/10.1109/ICIME.2010.5478220>
- Humphrey, W. S. (1995). *A Discipline for Software Engineering*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. Retrieved from [https://books.google.com.pe/books?id=MvnSAwAAQBAJ&lpg=PA3&dq=discipline of software - The State of the House&pg=PA3#v=onepage&q&f=false](https://books.google.com.pe/books?id=MvnSAwAAQBAJ&lpg=PA3&dq=discipline%20of%20software%20-%20The%20State%20of%20the%20House&pg=PA3#v=onepage&q&f=false)
- ISO/IEC. (2008). *ISO/IEC 12207: 2008 System and Software Engineering - Software Life Cycle Processes*. Geneva: ISO/IEC. Retrieved from <https://www.iso.org/obp/ui/#iso:std:iso-iec:12207:ed-2:v1:en>

- ISO/IEC. (2011a). *ISO/IEC TR 29110-1 Lifecycle profiles for Very Small Entities (VSEs) – Part 1: Overview*. Geneva: ISO/IEC.
- ISO/IEC. (2011b). *ISO/IEC TR 29110-1 Lifecycle profiles for Very Small Entities (VSEs) – Part 3: Assessment guide*. Geneva: ISO/IEC.
- ISO/IEC. (2011c). *ISO/IEC TR 29110-1 Lifecycle profiles for Very Small Entities (VSEs) – Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile* (Vol. 2011). Geneva: ISO/IEC.
- Jezreel, M., Mirna, M., Edrisi, M., Diego, E., Al Rumeni, A., & Juan, M. (2013). Leading the effort of software process improvement in SMEs. In *Proceedings of 8th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–7). Lisbon, Portugal: IEEE.
- Jezreel, M., Mirna, M., Pablo, N., Edgar, O., Alejandro, G., & Sandra, M. (2012). Identifying findings for software process improvement in SMEs: An experience. In *Proceedings of 9th Electronics, Robotics and Automotive Mechanics Conference, CERMA 2012* (pp. 141–146). Cuernavaca, Mexico: Curran Associates, Inc. (<https://doi.org/10.1109/CERMA.2012.30>)
- Kalinowski, M., Weber, K., Santos, G., Franco, N., Duarte, V., & Travassos, G. (2015). Software Process Improvement Results in Brazil Based on the MPS-SW Model. *Software Quality Professional*, 17(4), 15–28.
- Laporte, C. Y., O'Connor, R. V., & Paucar, L. H. G. (2016). The implementation of ISO/IEC 29110 software engineering standards and guides in very small entities. In *Evaluation of Novel Approaches to Software Engineering* (Vol. 599, pp. 162–179). Berlin: Springer. https://doi.org/10.1007/978-3-319-30243-0_9
- Larrucea, X., O'Connor, R. V., Colomo-Palacios, R., & Laporte, C. Y. (2010). Software Process Improvement in Very Small Organizations. *IEEE Software*, 33(April), 85–89. <https://doi.org/10.1109/MS.2016.42>
- Lee, J.-C., Shiue, Y.-C., & Chen, C.-Y. (2016). Examining the impacts of organizational culture and top management support of knowledge sharing on the success of software process improvement. *Computers in Human Behavior*, 54, 462–474. <https://doi.org/10.1016/j.chb.2015.08.030>
- Lucho, S., Melendez, K., & Dávila, A. (2018). Analysis of environmental factors in the adoption of ISO/IEC 29110. Multiple case study. In: J. Mejia, M. Muñoz, Á. Rocha, Y. Quiñonez, & J. Calvo-Manzano (Eds.), *Trends and Applications in Software Engineering* (pp. 82–93). Cham: Springer International Publishing.
- Miramontes, J., Muñoz, M., Calvo-Manzano, J. A., & Corona, B. (2016). Establecimiento del estado del arte sobre el aligeramiento de procesos de software. *RISTI - Revista Ibérica de Sistemas E Tecnologias de Informação*, (17), 16–25. <https://doi.org/10.17013/risti.17.16-25>
- Nirmala, D., & LathaMaheswari, T. (2016). Automated testcase generation for software quality assurance. In *Proceedings of 10th International Conference on Intelligent Systems and Control (ISCO)* (pp. 1–6). India: IEEE. <https://doi.org/10.1109/ISCO.2016.7727003>

- O'Connor, R. V. (2012). Evaluating management sentiment towards ISO/IEC 29110 in very small software development companies. *Communications in Computer and Information Science*, 290, 277–281. https://doi.org/10.1007/978-3-642-30439-2_31
- O'Connor, R. V. (2014). Early Stage Adoption of ISO/IEC 29110 Software Project Management Practices: A Case Study. In *Proceedings of International Conference on Software Process Improvement and Capability Determination* (pp. 226–237). Dublin, Ireland: Springer. https://doi.org/10.1007/978-3-319-13036-1_20
- O'Connor, R. V., & Laporte, C. Y. (2011). Using ISO/IEC 29110 to harness process improvement in very small entities. In *Proceedings of 18th European Software Process Improvement Conference*. (Vol. 172, pp. 225–235). Roskilde, Denmark: Springer. <https://doi.org/10.1007/978-3-642-22206-1>
- O'Connor, R. V., & Sanders, M. (2013). Lessons from a Pilot Implementation of ISO/IEC 29110 in a Group of Very Small Irish Companies. In *Proceedings of 13th International Conference on Software Process Improvement and Capability dEtermination (SPICE)* (pp. 243–246). Bremen, Germany: Springer. https://doi.org/10.1007/978-3-642-38833-0_23
- Oktaba, H., Alquicira, C., Su, A., Martinez, A., Quintanilla, G., Mara, R., & Flores, M. (2005). *Modelo de Procesos para la Industria de Software: MoProSoft*. Mexico: Secretaría de Economía.
- Paucar, L. G., Laporte, C. Y., Arteaga, J., & Bruggmann, M. (2015). Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru. *Software Quality Professional*, 17(2), 16–29.
- Paulk, M. C. (2011). *On Empirical Research into Scrum Adoption*. Pittsburgh: Carnegie Mellon University.
- Piattini, M. (2008). *Competisoft. Mejora de procesos software para pequeñas y medianas empresas y proyectos*. Madrid: RA-MA Editorial.
- Popay, J., Roberts, H., Sowden, A., Petticrew, M., Arai, L., Rodgers, M., & Duffy, S. (2006). *Guidance on the Conduct of Narrative Synthesis in Systematic Reviews. A Product from the ESRC Methods Programme*. Lancaster: Lancaster University <https://doi.org/10.1111/j.1523-536x.1995tb00261.x>
- Razón, O. C., Durán, J. G., González, J. A. G., Cortérighs, F. I., & Miranda, J. M. (2014). Cómo Guiar a las Pymes en la Mejora de Procesos Software. *RISTI - Revista Ibérica de Sistemas E Tecnologias de Informação*, (spe1), 17–30. <https://doi.org/10.4304/risti.e1.17-30>
- Ruiz, J. C., Osorio, Z., Mejia, J., Muñoz, M., Chávez, A. M., & Olivares, B. (2011). Definition of a hybrid measurement process for the models ISO/IEC 15504-ISO/IEC 12207:2008 and CMMI Dev 1.3 in SMEs. In *Proceedings of Electronics, Robotics and Automotive Mechanics Conference* (pp. 421–426). Cuernavaca, Mexico: IEEE. <https://doi.org/10.1109/CERMA.2011.74>
- Sánchez-Gordón, M. L., & O'Connor, R. V. (2016). Understanding the gap between software process practices and actual practice in very small companies. *Software Quality Journal*, 24(3), 549–570. <https://doi.org/10.1007/s11219-015-9282-6>

- Sanchez-Gordon, M. L., O'Connor, R. V., & Colomo-Palacios, R. (2015). Evaluating VSES viewpoint and sentiment towards the ISO/IEC 29110 standard: A two Country grounded theory study. *Communications in Computer and Information Science*, 526, 114–127. https://doi.org/10.1007/978-3-319-19860-6_10
- SEI (2010). *CMMI ® para Desarrollo, Versión 1.3. Software Engineering Institute*. Retrieved from <http://www.sei.cmu.edu/library/assets/whitepapers/Spanish Technical Report CMMI V 1 3.pdf>
- Sulayman, M., Mendes, E., Urquhart, C., Riaz, M., & Tempero, E. (2014). Towards a theoretical framework of SPI success factors for small and medium web companies. *Information and Software Technology*, 56(7), 807–820. <https://doi.org/10.1016/j.infsof.2014.02.006>
- Sulayman, M., Urquhart, C., Mendes, E., & Seidel, S. (2012). Software process improvement success factors for small and medium Web companies: A qualitative study. *Information and Software Technology*, 54(5), 479–500. <https://doi.org/10.1016/j.infsof.2011.12.007>
- Svensson, R. B., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., & Feldt, R. (2012). Quality Requirements in Industrial Practice - An Extended Interview Study at Eleven Companies. *IEEE Transactions on Software Engineering*, 38(4), 923–935. <https://doi.org/10.1109/tse.2011.47>
- Unterkalmsteiner, M., Gorschek, T., Islam, M., Cheng, C. K., Permadi, R., & Feldt, R. (2012). Evaluation and Measurement of Software Process Improvement—A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 38(2), 398–424. <https://doi.org/10.1109/TSE.2011.26>
- Vasconcellos, F., Landre, G., Cunha, J., Oliveira, J. L., Ferreira, R. A., & Vincenzi, A. M. (2016). Approaches to strategic alignment of software process improvement: A systematic literature review. *Journal of Systems and Software*, 123, 45–63. <https://doi.org/10.1016/j.jss.2016.09.030>
- Wongsai, N., Siddoo, V., & Wetprasit, R. (2015). Factors of influence in software process improvement: An ISO/IEC 29110 for very-small entities. In *7th International Conference on Information Technology and Electrical Engineering: Envisioning the Trend of Computer, Information and Engineering, ICITEE* (pp. 12–17). Chiang Mai, Thailand: IEEE.

Apéndice.

Disponibile en:

https://drive.google.com/drive/folders/oB_K_llz5juqPMll5TC1sNkliWGc

Impacto de los factores sociológicos de los usuarios en la aceptación y uso de software de gestión de proyectos en la mediana empresa en Lima

César Aguilera¹, María Teresa Villalobos², Abraham Dávila¹

{cesar.aguilera, mtvillalobosa, abraham.davila}@pucp.edu.pe

¹ Departamento de Ingeniería, Pontificia Universidad Católica del Perú, Lima 32, Lima, Perú.

² Departamento de Ciencias, Pontificia Universidad Católica del Perú, Lima 32, Lima, Perú

DOI: 10.17013/risti.26.17-30

Resumen: (Contexto) El uso de aplicaciones de software generalmente en su adopción no siempre es efectiva. En particular, en la gestión de proyectos, donde existen algunas dificultades en el contexto de las empresas medianas en Lima para utilizar las herramientas de gestión de proyectos. (Objetivo) Estudiar los factores relacionados con características del software, la organización y los usuarios para la aceptación del software de gestión. (Método) Se diseñó y ejecutó una encuesta basada en un estudio realizado en el mismo tema. (Resultados) Se obtuvieron respuesta de 77 gerentes de proyectos. (Conclusión) Los factores como la funcionalidad, el tamaño de la organización, la complejidad del proyecto y el uso del software son los más representativos. En el estudio también se encuentra que: (i) existe una relación fuerte y significativa entre el uso del software y la percepción del desempeño del gerente del proyecto y (ii) los factores como la capacitación, la experiencia y el nivel educativo no tienen ningún efecto.

Palabras-clave: gestión de proyectos, software de gestión de proyecto, aceptación de tecnología de la información, modelo de aceptación tecnológica.

Impact of the sociological factors of the user on the acceptance and use of project management software in the medium-sized company in Lima

Abstract: (Context) The use of software applications generally in their adoption is not always effective. In particular, in project management, where there are some difficulties in the context of medium-sized companies in Lima to use project management tools. (Objective) Study the factors related to the characteristics of the software, the organization and the users for the acceptance of the management software. (Method) A survey based on a study conducted on the same topic was designed and executed. (Results) Response was obtained from 77 project managers. (Conclusion) Factors such as functionality, the size of the organization, the complexity of the project and the use of software are the most representative. In the study also finds that: (i) exists a strong and significant relationship between the

use of software and the perception of the project manager's performance and (ii) factors such as training, experience and educational level have no effect.

Keywords: project management, project management software, information technology acceptance, technology acceptance model.

1. Introducción

La gestión de proyectos es un tema cada vez más discutido en las organizaciones, aunque no se trata de una novedad como disciplina (Abreu, 2008). Entendemos como un proyecto a todo esfuerzo para alcanzar un objetivo específico a través de un conjunto único de tareas interrelacionadas y de la utilización eficaz de recursos (Gido & Clements, 2007).

Los proyectos son concebidos y realizados para desarrollar productos y servicios para ganar eficiencia operativa (Anantatmula, 2008). De acuerdo con Willians (2005), las empresas están cada vez más orientadas a proyectos y el gasto global en proyectos es de muchos billones de dólares anualmente.

En la naturaleza de los proyectos se pueden identificar diferentes características como la duración o la complejidad, y debido a estos factores; alguno de ellos puede influir en la determinación del uso de software de gestión de proyectos (Anantatmula, 2008). El software de gestión de proyectos ha incorporado durante los últimos años, muchas funcionalidades que facilita el trabajo de un equipo gestor de proyecto o la gestión de proyectos. Para Gido y Clements (2007), el gerente de proyectos tiene la responsabilidad de liderar la planificación, organizar y controlar los esfuerzos desplegados para que los objetivos del proyecto sean alcanzados. En grandes organizaciones, es común que los equipos participen de diversos proyectos simultáneamente y, para administrar esa multiplicidad de proyectos, cada vez más se utilizan softwares de gestión de proyectos como herramienta de soporte a las tareas de los gerentes (Da Silva Radaiesk, Feier Fróes & Lindstrom Bandeira).

Algunas herramientas o técnicas tales como WBS (estructura de desglose del trabajo), CPM (método de ruta crítica) y PERT (técnicas de revisión y evaluación de proyectos) son características comunes en cualquier software de gestión de proyectos (Meredith & Mantel, 2006). Además de otras características como su funcionalidad mediante el uso de Internet, el trabajo colaborativo y la documentación adjunta al proyecto (Straub, 2009).

La tecnología de la información (TI) implica el desarrollo, mantenimiento y uso de sistemas informáticos, software y redes para el procesamiento y distribución de datos (Information technology, 2012). Con el aumento del uso de la TI, dentro de una empresa, la TI se vuelve de una gran importancia en la implementación de los procesos diarios y la toma de decisiones rápida y efectiva (Jabbouri, Siron, Zahari & Khalid, 2016).

La participación de las áreas de TI ha sido importante para incorporar estas herramientas en las aplicaciones de gestión de proyectos, lo que ha permitido a los directores de proyectos maximizar el uso de los métodos de gestión de proyectos (Anantatmula, 2008). Sin embargo, el problema principal es que no se conocen los factores organizativos y de usuario que influyen en la aceptación del software de gestión de proyectos; este problema principal es el efecto de poca o ninguna comprensión del proceso de adopción de nueva tecnología y de que se desconocen las características comunes y los factores sociológicos de los usuarios.

En este artículo se presenta un análisis sobre los aspectos sociológicos de los usuarios en la aceptación y uso de software de gestión de proyectos, que complementa el artículo (Aguilera, Villalobos & Dávila, 2018). El documento se organiza de la siguiente manera: la Sección 2, menciona los modelos que tradicionalmente se han utilizado para explicar la adopción de TI a nivel individual; en la Sección 3, se describe el protocolo de investigación; en la Sección 4, se presentan los resultados de la encuesta; en la Sección 5, se prueban las hipótesis; en la Sección 6, se establecen una consideración final, conclusión y trabajo futuro.

2. Antecedentes

La adopción de TI ha sido ampliamente estudiada para encontrar fundamentos teóricos, factores, roles y estructuras organizacionales. Toda esta investigación ha sido necesaria porque la adopción de la tecnología es un proceso complejo y social que involucra factores contextuales, emocionales y cognitivos (Straub, 2009).

La teoría del comportamiento se usa para predecir si una persona tiene la intención de hacer algo, los investigadores necesitan saber: (i) si la persona está a favor de hacerlo (“actitud”); (ii) cuánto siente la persona la presión social para hacerlo (“norma subjetiva”); y (iii) sea la persona que se siente en control de la acción en cuestión (“control del comportamiento percibido”).

La Teoría de la Acción Razonada (TRA) es un modelo general de la predicción del comportamiento humano introducido originalmente por Fishbein en 1967 y luego perfeccionado, desarrollado y probado por Fishbein y Ajzen en 1975, que define las relaciones entre la creencia, actitud, norma, la intención y la conducta o comportamiento (Ajzen & Fishbein, 1980; Davis, 1985; Straub, 2005).

La Teoría de Comportamiento Planificado (TPB, por sus siglas en inglés) es un modelo muy similar a la TRA, excepto que una construcción adicional se tiene en cuenta: el control del comportamiento percibido (PBC, por sus siglas en inglés), que se refiere a la percepción control sobre el rendimiento de un determinado comportamiento (Ajzen, 1985).

Los modelos más utilizados, desde la perspectiva donde los individuos son altamente racionales y toman sus decisiones para maximizar el valor o la utilidad (Slappendel, 1996), y por lo tanto convertirse en la fuente principal de innovación y cambios organizacionales, son: el Modelo de Aceptación de Tecnología (TAM), la Teoría de Comportamiento Planificado (TPB) y la Teoría Unificada de Aceptación y Uso de Tecnología (UTAUT) (Olivera & Fraga, 2011).

Puesto que la adopción de tecnologías de la información presupone un uso nuevo de estas tecnologías y sistemas, o la introducción de ellos en una organización, modelos de adopción se basan en modelos de innovación, en el sentido de que esta última implica una novedad percibida dentro de la organización, tal como una idea, artefacto o una práctica (Rogers, 2003).

En 2003, Liberatore y Pollack-Johnson (2003) desarrollaron una investigación con el objetivo de analizar y evaluar los factores que influyen en la extensión del uso, el tipo de uso (planificación versus planificación y control) y la selección de software de gestión de proyectos. En esta investigación, los autores realizaron una encuesta con 240 miembros del

Project Management Institute (PMI) actuando en diferentes ramas de negocio de Estados Unidos (EUA). Los resultados encontrados sugieren que el tamaño y la complejidad de los proyectos influyen en el uso del software de gestión de proyectos. También indican que la extensión del uso del software de gestión de proyectos es influenciada por el número de actividades de un proyecto típico, número de proyectos que son tácticamente administrados (simultáneamente), porcentaje de trabajo involucrado en la gestión de proyectos e histórico del uso de software de gestión de proyectos en la organización.

En 2008, Ali, Anbari y Money (2008) presentan una investigación hacia la comprensión y el entendimiento de los factores que afectan el uso de la tecnología de la información en materia de gestión de proyectos.

La presente investigación sobre el impacto de los factores organizacionales y de usuarios en la aceptación y uso de software de gestión busca verificar si los resultados encontrados en el estudio realizado por Ali, Anbari y Money (2008) se aplican a la realidad de la mediana empresa en Lima Metropolitana.

3. Protocolo de investigación

En esta sección presentamos el protocolo de investigación basado en las hipótesis y metodología seguidas para este estudio.

3.1. Marco teórico e Hipótesis

En la Figura 1 presenta el modelo de investigación examinado en este estudio. La investigación modelo evalúa el alcance del uso de software de gestión de proyecto por profesionales del proyecto y el impacto de su uso en su rendimiento percibido (Ali, Anbari & Money, 2008).

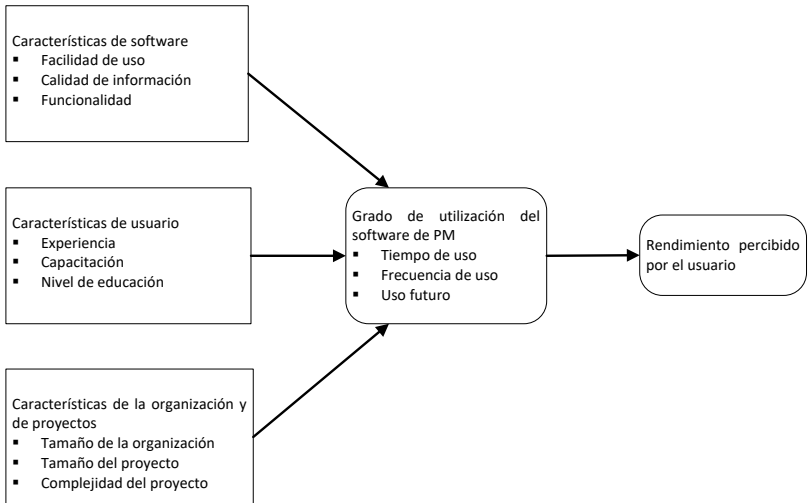


Figura 1 – Modelo de investigación

El modelo postula que la aceptación del software de administración de proyectos es una función de calidad de la información percibida, funcionalidad del software, facilidad de uso, complejidad, tamaño del proyecto, tamaño de la organización, nivel de formación del usuario, nivel de educación y experiencia. El modelo también propone que el uso del software de gestión de proyecto tiene un impacto directo y positivo en el rendimiento percibido por el gerente de proyecto (Ali, Anbari & Money, 2008). Por otro lado, se busca identificar qué factores sociológicos de los usuarios influye en mayor medida en el uso del software de gestión de proyectos. El modelo de investigación propone las siguientes hipótesis:

- H1: la facilidad de uso percibida tiene una relación positiva con el uso del software de gestión de proyectos
- H2: la funcionalidad percibida tiene una relación positiva con el uso del software de gestión de proyectos
- H3: la calidad de la información percibida tiene una relación positiva con el uso del software de gestión de proyectos
- H4: el tamaño de la organización tiene una relación positiva con el uso del software de gestión del proyecto.
- H5: el tamaño del proyecto tiene una relación positiva con el uso del software de gestión de proyectos.
- H6: la complejidad del proyecto tiene una relación positiva con el uso del software de gestión de proyectos
- H7: el nivel de capacitación del gerente de proyectos tiene una relación positiva con el uso del software de gestión de proyectos
- H8: el nivel de experiencia del gerente de proyectos tiene una relación positiva con el uso de software de gestión de proyectos
- H9: el nivel de educación del gerente de proyecto tiene una relación positiva con el uso de software de gestión de proyectos
- H10: el uso de software de gestión de proyectos tiene una relación positiva con el rendimiento del gerente de proyecto

3.2. Metodología

En este estudio usamos una encuesta como protocolo de investigación. El instrumento utilizado para la evaluación fue traducido del inglés a español y las preguntas revisadas para facilitar su comprensión. Posteriormente, algunos elementos se modificaron para aumentar su claridad. El instrumento final se construyó usando una escala de Likert de 5 puntos (1. Casi nunca, 2. A veces, 3. Normalmente, 4. Casi siempre, 5. Siempre). Se utilizó una encuesta web como medio de recopilación de datos.

Debido a la naturaleza exploratoria del estudio, se eligió un muestreo no probabilístico por conveniencia, teniendo como referencia a los gerentes de proyecto de las áreas de informática de tamaño mediano en la ciudad de Lima Metropolitana.

4. Resultados de la encuesta

Los datos se obtuvieron de una muestra independiente compuesta por profesionales de gestión de proyectos obtenidos a partir de los contactos del investigador, a partir de la base de datos de contactos en el Perú de la empresa Dharma Consulting y de una lista

de contactos proporcionada por la Sección de Ingeniería Informática de la Pontificia Universidad Católica de Perú.

Participaron 108 profesionales de gestión de proyectos en la encuesta (completadas) obteniendo 77 respuestas que indicaban que utilizaban algún tipo de software de gestión de proyectos que representa el 70% de la muestra; el 30% restante no utilizaba ninguna herramienta. Además, de ese 70% de quienes usan alguna herramienta, fue Microsoft® Project la herramienta más usada como herramienta de gestión de proyectos. Más del 57% de los encuestados usan Microsoft Project, 10% utiliza Microsoft Project y Project Server (versión de Microsoft Project para la administración de las carteras de proyectos), 6% usa Primavera Project Management Systems con Microsoft Project y el 5% solo utiliza Project Server.

La Tabla 1 muestra el nivel de educación de la muestra de investigación, donde más del 76.9% tiene un título profesional, un postgrado de maestría o doctorado, y el resto de los participantes tenían grado de bachiller, habían completado la universidad o tenían instrucción técnica. Además, el 49.4% de los participantes tienen más de cinco años en el campo de la gestión de proyectos y el 48.0% de ellos tiene más de cinco años de experiencia en el uso de software de gestión de proyectos. La Tabla 2 muestra los datos con respecto a la edad de los participantes, donde el 95.4% de ellos están en el rango de 20 a 50 años.

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Doctorado	3	2,8	2,8
	Magister	39	36,1	38,9
	Titulado	41	38,0	76,9
	Bachiller	18	16,7	93,6
	Egresado de universidad	4	3,7	97,3
	Instrucción técnica	3	2,8	100,0
	Total	108	100,0	

Tabla 1 – Nivel de educación

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	20-30	28	25,9	25,9
	31-40	53	49,1	75,0
	41-50	22	20,4	95,4
	51-60	5	4,6	100,0
	Total	108	100,0	

Tabla 2 – Edad de los participantes

El análisis de confiabilidad del instrumento de medición fue omitido porque fue tomado de una investigación previa. Sin embargo, los altos valores de alfa de Cronbach obtenidos en cada una de las secciones indican y confirman la consistencia de las preguntas. Como primer paso, se realizó un Análisis Factorial, que incluyó una rotación Varimax para cada una de las secciones del cuestionario de las preguntas de investigación. Luego, se identificaron los factores para cada una de las secciones de la encuesta: rendimiento, uso, características del software, autoeficacia, características de la tarea y apoyo administrativo. Para determinar el número de factores a retener, se aplicó el Criterio de Kaiser, es decir, retener aquellos factores con un valor propio mayor que 1. Una vez que se obtuvieron los factores, se realizó el análisis de regresión de los factores identificados frente a las variables de uso. Tal como se realizó en la investigación original.

4. Prueba de las hipótesis de investigación.

Los factores obtenidos se muestran en la Tabla 3. Al analizar las varianzas de ANOVA, se pudo validar el modelo y establecer los valores de β factores que lo componen. El análisis detallado de cada uno de los resultados obtenidos para cada una de las hipótesis se resume en la Tabla 4. Cinco de las diez hipótesis de investigación fueron compatibles y otras cinco no fueron compatibles.

	Factor	Variable
Rendimiento	F1	Rendimiento
	F2	Rendimiento de la gestión
Grado de Uso	F1	Área de conocimiento (Extent of use)
	F2	Proyectos complejos [-]
	F3	Grupos de procesos [-] (Extent of use)
	F4	Proyectos cortos / pequeños
	F5	Uso (facilidad de uso)
Características del software	F1	Calidad de la información
	F2	Integración
	F3	Funcionalidad
	F4	Facilidad de Uso [-]
Autoeficacia	F1	Con soporte
	F2	Sin soporte [-]
Características de la tarea o Complejidad del proyecto	F1	Tareas rutinarias
	F2	Tareas establecidas (documentadas)
	F3	Tareas interdependientes [-]
	F4	Tareas pioneras (equivocability)
Apoyo de la gestión	F1	Apoyo Alta Gerencia

Tabla 3 – Resumen de factores

Respecto a los factores de carga de Características del software se agruparon en cuatro factores, debido principalmente a que el factor de integración se encuentra unificado y no definido en interna y externa como en la investigación original. Otra diferencia es que respecto al factor de Facilidad de uso se encontró una correspondencia inversa.

Sobre los factores de Autoeficacia, se obtuvo dos factores de manera similar a la investigación original. Sin embargo, una diferencia con ese estudio es que la autoeficacia sin soporte tiene una correspondencia inversa.

Acerca de los factores de la Complejidad del proyecto o también denominado Características de las tareas, se obtuvo la misma cantidad de factores que en la investigación original. Con la diferencia que para el factor denominado Tareas interdependientes (problemas no rutinarios y que involucran más de un área y proceso) se encontró una correspondencia inversa.

Con respecto al Apoyo de la gestión, de manera similar a la investigación original se coincide con un único factor denominado Apoyo de la Alta Gerencia

Hipótesis	Resultado	p-valor	R ²
H1	No soportado	0.772	0.120
H2	Soportado	0.000	16.50
H3	Soportado	0.001	15.44
H4	Parcialmente soportado	0.058	5.05
H5	No soportado	-	-
H6	Soportado	0.004	11.14
H7	No soportado	-	-
H8	No soportado	-	-
H9	No soportado	-	-
H10	Soportado	0.000	24.57

Tabla 4 – Resultado de las hipótesis modelo

En comparación con el estudio de investigación original, tenemos los siguientes resultados:

- La hipótesis 1 (H1) no fue soportada, lo que indica que no se encontró una relación estadísticamente significativa ($\beta = 3.4\%$, $p < 0.772$, $R^2 = 0.12$). Este hallazgo se contradice con una encuesta realizada por The Access Group en 2013 que establece que los más buscado después de las características del software de gestión de proyectos son la fiabilidad, la facilidad de integración y la facilidad de uso (The Access Group, 2013).
- La hipótesis 2 (H2) puede confirmar que para los participantes de este estudio es más relevante que el software cumpla con la funcionalidad solicitada antes de la satisfacción en el uso. En ese mismo sentido en 2015, la encuesta “Informe de investigación de usuarios de gestión de proyectos” de Capterra indica que el

factor más importante para elegir la compra del software de gestión de proyectos es la funcionalidad (40%) seguida de la facilidad de uso (24%) (Capterra, 2015).

- La hipótesis 3 (H3) es soportada. La encuesta de Capterra en 2015 señala que los principales desafíos que impulsa a las personas a usar software de gestión de proyectos es controlar el tiempo y los costos del proyecto (62%) (Capterra, 2015). Lo que puede interpretarse como la necesidad de poder tener información sobre estos dos aspectos en los proyectos.
- La hipótesis 4 (H4) está parcialmente respaldada en la investigación. De acuerdo con esta declaración, tenemos que PwC en una encuesta llamada “Perspectivas y Tendencias: Programa Actual y Prácticas de Gestión de Proyectos” (PriceWaterhouseCoopers, 2007), indica que el 77% de las empresas participantes utilizan software de gestión de proyectos. De este conjunto, aquellas organizaciones que tienen un mayor nivel de madurez tienen mayor probabilidad de usar software de gestión de proyectos: el 95% de las organizaciones de la categoría de mayor madurez utilizan software de gestión de proyectos, en comparación con solo el 55% dentro de la categoría de menor nivel de madurez. En un informe anterior de PriceWaterhouseCoopers (2004), se menciona que si una organización alcanza un cierto nivel de madurez, donde los procesos de gestión de proyecto están institucionalizados, el uso de software aumentará significativamente el rendimiento general del proyecto”.
- La hipótesis 5 (H5) no posee una correlación estadísticamente significativa con el uso de software de gestión de proyectos. En 2003, Liberatore y Pollack-Johnson obtuvieron resultados que sugieren que el tamaño y la complejidad de los proyectos influyen en el uso del software de gestión de proyectos.
- La hipótesis 6 (H6) es soportada. Los participantes en este estudio identifican que la complejidad del proyecto está directamente relacionada con el uso del software de gestión de proyectos.
- La hipótesis 7 (H7) no es soportada, no se pudo comprobar la existencia de una relación entre el nivel de capacitación del gerente de proyecto y el uso del software de gestión de proyectos. En la encuesta realizada por PriceWaterhouseCoopers (2007), encontraron una correlación entre la certificación de gestión de proyectos (certificaciones PMI y Prince 2) y el rendimiento del proyecto. En el mismo estudio de PwC, se descubrió que el uso del software de gestión de proyectos está relacionado con el alto rendimiento del proyecto. El 77% de las empresas participantes utilizaban software de gestión de proyectos. En un informe anterior realizado por la misma compañía, en 2004, se encontró la misma correlación con un nivel de uso del 87%.
- Las hipótesis 8 (H8) y 9 (H9) no fueron soportadas, no se encontró ninguna correlación con respecto a la experiencia y la educación con el uso del software de gestión de proyecto. En 2012, en otro estudio de PwC, se menciona que los gerentes de unidades de negocios comprometidos tienen la correlación más alta con el éxito del proyecto. Otro tipo de personal clave que contribuye al éxito del proyecto incluye gerentes de calidad calificados, y personal de control de calidad capacitado y calificado.
- En un estudio realizado sobre los factores de éxito de un gerente de proyecto (Saadé, Dong, & Wan, 2015) encontraron tres factores importantes como el

compromiso, educación (credenciales de certificación y habilidades de escritura) y experiencia (duración de proyectos culminados y tamaño de equipos de trabajo gestionados).

- Por otro lado en una estudio sobre adopción tecnológica (Palos-Sánchez, Arenas-Márquez, & Aguayo-Camacho, 2017) se pudo confirmar que la formación profesional, grado en que una empresa instruye a sus empleos, ejerce un efecto positivo sobre la intensidad de uso de sistemas (software).
- La hipótesis 10 (H10). De acuerdo con H10, se encontró que el uso de software de gestión de proyectos tiene una relación positiva y estadísticamente significativa con el rendimiento percibido por un gerente de proyecto, debido a los valores obtenidos ($\beta = 50.6\%$, $p < 0.0001$, $R^2 = 24.57$). En otras palabras, el uso de software de gestión de proyectos representa el 24.57% de la variación en el rendimiento percibida por un gerente de proyecto.

Con base en el análisis de regresión obtenido, el impacto del uso del software en la percepción del rendimiento se puede calcular utilizando la siguiente ecuación de regresión (1)

$$\text{Percepción de rendimiento} = 0.019 + 0.506 * \text{Uso de SW de PM} + \varepsilon. \quad (1)$$

Para cada unidad de aumento en el uso del software de gestión de proyectos, la percepción del rendimiento de los usuarios aumenta en un 50,6% de la unidad.

En la Tabla 5 podemos encontrar los resultados encontrados para cada una de las hipótesis en ambas investigaciones.

Hipótesis	Resultado del estudio original	Resultado del estudio actual
H1	Fuertemente apoyado	No soportado
H2	Fuertemente apoyado	Soportado
H3	Fuertemente apoyado	Soportado
H4	Moderadamente soportado	Parcialmente soportado
H5	Parcialmente soportado	No soportado
H6	Fuertemente apoyado	Soportado
H7	No soportado	No soportado
H8	Parcialmente soportado	No soportado
H9	Moderadamente soportado	No soportado
H10	Fuertemente apoyado	Soportado

Tabla 5 – Hipótesis contrastadas con la investigación original

Según un estudio de Straub (1997), se argumenta que las predicciones de TAM no necesariamente se mantendrán entre culturas. Esto indica que los países difieren en términos culturales. La investigación de Hofstede (1980) sobre las dimensiones culturales proporciona una base teórica para explorar el impacto de las diferencias culturales en la adopción y difusión de innovaciones basadas en TI, como el correo electrónico.

Hofstede (1980) describe cuatro dimensiones que pueden ser utilizadas para distinguir entre las diferentes culturas: poder-distancia, anulación de la incertidumbre, masculinidad y el individualismo. La Tabla 6 enumera las cuatro dimensiones y una breve descripción de cada una.

Hofstede Dimensión	Abr.	Descripción
<i>Poder - distancia</i>	PDI	Grado de desigualdad entre las personas que la población de una cultura considera normal
<i>Anulación de la incertidumbre</i>	UAI	El grado al cual la gente en una cultura se siente incómoda con la incertidumbre y la ambigüedad
<i>Individualismo</i>	IDV	Grado en que las personas en una cultura prefieren actuar como individuos y no como miembros de grupos
<i>Masculinidad</i>	MAS	Grado en que valores como la asertividad, el rendimiento, el éxito y la competencia prevalecen entre las personas de una cultura sobre valores más suaves como la calidad de vida, el mantenimiento de cálidas relaciones personales, el servicio, el cuidado de los débiles, etc.

Tabla 6 – Las cuatro dimensiones culturales de Hofstede

5. Consideraciones, conclusiones y trabajo futuro

Dado que esta fue una investigación inicial en la aplicación del modelo TAM en el software de gestión de proyectos de empresas medianas en la ciudad de Lima, hubo varias limitaciones. Es muy probable que los resultados y conclusiones de esta investigación se limiten a la muestra, las variables y el marco de tiempo representado por el diseño de la investigación. La cultura del país y la naturaleza de las organizaciones (públicas frente al privado) son variables que podrían ser mejor explotadas (Da Silva Radaiesk, Feier Fróes & Lindstrom Bandeira, 2015). En la investigación de Saadé, Dong y Wan (2015) se menciona que las credenciales de certificación y la capacitación son percibidos por los participantes (análisis descriptivo) como irrelevantes lo cual parece ser una indicación de otros factores influyentes como el político y cultural dentro de una organización.

Los resultados presentan algunas diferencias que nos indican nuevos caminos de investigación. El perfil profesional de los gerentes de proyectos probablemente contribuyó a las diferencias encontradas entre las investigaciones (Da Silva Radaiesk, Feier Fróes & Lindstrom Bandeira, 2015).

Las conclusiones deben considerarse intentos hasta que otras investigaciones adicionales puedan confirmar o rechazar hallazgos similares. Además, estos hallazgos se limitan a los gerentes de proyectos de empresas medianas en Lima Metropolitana.

Las características de la muestra dejan en claro que los encuestados de esta investigación no pueden verse como un reflejo de la población en general. Por lo tanto, los resultados de este estudio no pueden extrapolarse ya que la muestra puede no ser muy representativa de la población nacional de quienes manejan los proyectos.

Para futuros estudios se propone extender el modelo propuesto y buscar otras variables que se consideran importantes, relacionadas con las características del software y con las

características del usuario, tal como en la investigación de Liberatore y Pollack-Johnson (2003) donde todos los participantes eran certificados PMI.

Esta investigación enriquece la comprensión de los factores que afectan el uso de las tecnologías de la información en la gestión de proyectos. En el estudio, el uso de software de gestión de proyectos explica el 24,57% de la variación en el rendimiento percibida por un gerente de proyecto.

Se encontraron diferencias marcadas en los resultados de nuestra investigación con respecto al estudio inicial. De todos estos factores no confirmados, tres son específicos del gerente de proyecto (capacitación en gestión de proyectos, nivel de experiencia y nivel de formación académica). La falta de confirmación de estos tres factores puede deberse a términos culturales, como Straub (1997) lo pone en su estudio titulado “Probando el modelo de aceptación de la tecnología a través de las culturas: un estudio de tres países”, donde se señala que las dimensiones culturales proporcionan una base para explorar el impacto de las diferencias culturales en la adopción y difusión de TI innovaciones basadas (Hofstede, 1980). Por otro lado, de León-Sigg, Vázquez-Reyes y Villa-Cisneros (2017) señalan que una de las razones por la que los empleados tienen buena predisposición hacia las TI, y apoyan las iniciativas de adopción es que se contratan a los usuarios ya capacitados o son personas a las que se les brinda la capacitación específica que necesitan.

Por otro lado, los resultados obtenidos indican que los factores sociológicos no influyen en la aceptación del software de gestión de proyectos. En opinión de los responsables del proyecto, parece claro que la experiencia profesional y la formación no puede ser reemplazada, pero el software puede ayudar a mejorar la capacidad de organización, mejorar la programación de tareas y obtener una mejor comprensión de cómo y por qué nos va bien o mal en los proyectos.

Sin embargo, parece que estas tres hipótesis parecen no tener una relación directa con el uso del software, pero sí con la percepción de gerentes de proyecto con éxito tal como lo menciona Saadé, Dong, y Wan, (2015).

Referencias

- Abreu, A. (2008). Gerenciamento de projetos: entenda alguns dos principais conceitos. *Engenharia de Softwares Magazine*, (6).
- Aguilera C., Villalobos M.T., & Dávila A. (2018) Impact of organizational and user factors on the acceptance and use of project management software in the medium-sized company in Lima. In: Mejía J., Muñoz M., Rocha Á., Quiñonez Y., Calvo-Manzano J. (eds) *Trends and Applications in Software Engineering. CIMPS 2017. Advances in Intelligent Systems and Computing*, vol 688. Berlin: Springer, Cham.
- Ali, A. S. B., Anbari, F. T., & Money, W. H. (2008). Impact of Organizational and Project Factors on Acceptance and Usage of Project Management Software and Perceived Project Success. *Project Management Journal*, 39(2), 5–33.
- Ajzen, I., & Fishbein, M. (1980). *Understanding Attitude and Predicting Social Behavior*. Nueva Jersey: Prentice-Hall, Inc.

- Ajzen, I. (1985). *From intentions to actions: A theory of planned behavior* (pp. 11–39). Berlin: Springer.
- Anantatmula, V. S. (2008). The Role of Technology in the Project Manager Performance Model. *Project Management Journal*, 39(1), 34–48.
- Birk, J. (1990). A corporate project management council. In *The Proceedings of the IEEE International Engineering Management Conference*. Santa Clara, CA, USA: IEEE.
- Capterra. (2015). *Project Management User Research Report*. Retrieved from: <https://www.capterra.com/project-management-software/user-research/>
- Da Silva-Radaiesk, G., Feier-Fróes, M., & Lindstrom-Bandeira, D. (2015). Fatores que influenciam o uso e a seleção de um software de gerenciamento de projetos: aplicando o modelo de liberatore e pollack-johnson em uma empresa pública brasileira de desenvolvimento de software. *Brazilian Journal Of Management/ Revista De Administração Da UFSM*, 8(1), 9–25. doi:10.5902/198346595873.
- Davis, F. D. (1985). *A Technology Acceptance Model for Empiricall Testing New End-user Information Systems: Theory and Results*. Cambridge, MA: MIT Sloan School of Management.
- de León-Sigg, M., Vázquez-Reyes, S., & Villa-Cisneros, J. L.. (2017). Factores que Afectan la Adopción de Tecnologías de Información en Micro y Pequeñas empresas: un Estudio Cualitativo. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (22), 20–36. <https://dx.doi.org/10.17013/risti.22.20-36>.
- Dillon, A., & Morris, M. G. (1996). User Acceptance of information technology: Theories and Models. *Annual Review of Information Science and Technology* (31), 31–32.
- Gido, J., & Clements, J. P. (2007). *Gestão de projetos*. 3.ed. São Paulo: Cengage Learning.
- Hofstede, G. (1980). *Culture's Consequences: International Differences in Work-Related Values*.
- Information technology. (2012). *Merriam-Webster Online Dictionary*. Retrieved from: <http://www.merriam-webster.com/dictionary/information%20technology>.
- Jabbouri, N. I., Siron, R., Zahari, I., & Khalid, M. (2016). Impact of Information Technology Infrastructure on Innovation Performance: An Empirical Study on Private Universities. *Iraq Procedia Economics and Finance*, 39, 861–869.
- Liberatore, M. J., & Pollack-Johnson, B. (2003) Factors influencing the usage and selecton of project management software. *IEEE Transactons on Engineering Management*, 50(2), 164–174.
- Meredith, J. R., & Mantel, S. J. (2006). *Project management: A managerial approach*. Nueva York: Wiley.
- Olivera, T., & Fraga-Martins, M. (2011). Literature Review of Information Technology Adoption Models at Firm Level. *The Electronic Journal Information Systems Evaluation*, 14(1), 110–121.

- Palos-Sánchez, P. R., Arenas-Márquez, F. J., & Aguayo-Camacho, M. (2017). La adopción de la tecnología cloud computing (SaaS): efectos de la complejidad tecnológica vs formación y soporte. *RISTI - Revista Ibérica de Sistemas e Tecnologías de Informação*, (22), 89-105. <https://dx.doi.org/10.17013/risti.22.89-105>.
- PricewaterhouseCoopers. (2004). *Boosting Business Performance through Programme and Project Management*.
- PricewaterhouseCoopers. (2007). *Insights and Trends: Current Programme and Project Management Practices*.
- PricewaterhouseCoopers. (2012). *Insights and Trends: Current Portfolio, Programme, and Project Management Practices*.
- Rogers, E. (2003). *Diffusion of innovations*, 5a ed. Nueva York: Simon and Schuster Inc.
- Saadé, R. G., Dong, H., & Wan, H. (2015). Factors of project managers success. Interdisciplinary. *Journal of Information, Knowledge, and Management*, 10, 63-80.
- Slappendel, C. (1996). Perspectives on Innovation in Organizations. *Organization Studies*, 17(1), 107-129.
- Straub, D. (1997) Testing the Technology Acceptance Model Across Culture: A Three Country Study. *Information & Management*, 33(1), 1-11.
- Straub, E. T. (2009). Understanding Technology Adoption: Theory and Future Directions for Informal Learning. *Review of Educational Research*, 79(2), 625-649.
- The Access Group. (2013). *Inbox Insight Survey 2013*.
- Williams, T. (2005). Assessing and moving on from the dominant project management discourse in the light of project overruns. *IEEE Transactions on Engineering Management*, 52, 497-508.

Identificando caminos para iniciar la implementación de mejora de procesos de software en organizaciones de desarrollo de software

Jezreel Mejía¹, Marcos Gonzalez¹, Jose Calvo-Manzano², Tomas San Feliu²

jmejia@cimat.mx, marcos.gonzalez@cimat.mx, joseantonio.calvomanzano@upm.es, tomas.sanfeliu@upm.es

¹Centro de Investigación en Matemáticas – Unidad Zacatecas, 98160, Zacatecas, Zacatecas, México

²ETS de Ingenieros Informáticos :: Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte (Madrid), España

DOI: 10.17013/risti.26.31-42

Resumen: En la actualidad, uno de los principales obstáculos de las organizaciones de desarrollo de software en la implementación del Modelo CMMI-Dev es como iniciar su implementación y aumentar la probabilidad de éxito en una evaluación SCAMPI. Este método de evaluación permite validar, establecer e identificar las oportunidades de mejora de procesos para obtener un nivel de madurez. Sin embargo, el método de evaluación SCAMPI no garantiza una evaluación exitosa, debido a que el proceso de evaluación es empírico basado en la experiencia del equipo auditor experto en Mejora de Procesos de Software con respecto a este modelo. Este artículo propone la identificación de alternativas de caminos a través de una herramienta para lograr una efectiva implementación en el modelo CMMI-DEV Nivel 2, considerando sus objetivos de negocio de la organización.

Palabras-clave: Herramientas; Software; Procesos; Mejora; CMMI; Evaluación.

Identifying paths to start the software process improvement in software development organizations

Abstract: One of the main obstacles of software development organizations in the implementation of the CMMI-Dev Model is how to start its implementation and increase the probability of success in a SCAMPI evaluation. This evaluation method allows to validate, establish and identify opportunities for process improvement to obtain a level of maturity in this model. However, the SCAMPI evaluation method does not guarantee a successful evaluation, because the evaluation process is empirical based on the experience of the expert audit team in Software Process Improvement with respect to this model. This article proposes the identification of paths alternatives through a tool to achieve an effective implementation in the CMMI-DEV v1.3 Level 2 model, considering its business objectives of the organization.

Keywords: Tools; Software; Process; Improvement; CMMI; Evaluation.

1. Introducción

Hoy en día la implementación de diversas metodologías y modelos como: IDEAL, PDCA, Pmbok, COMPETISOFT, ágil SPI-Process, ISO 27001, ISO 9001, CMMI, ITIL, COBIT, RISK IT, VAL IT y ISO 90003, son aplicados en gran medida para mejorar, madurar, adquirir e institucionalizar mejores prácticas y obtener la base del conocimiento en el desarrollo de sus proyectos (Pardo et al, 2011).

Investigaciones recientes muestran que las organizaciones de desarrollo de software utilizan la Mejora de Procesos de Software (SPI, por sus siglas en inglés) para implementar la administración, control y estructura al desarrollo de software resultando en un cambio significativo ya que se considera que la calidad del software es la llave para la obtención de mejores ventajas competitivas haciendo uso de la implementación de los modelos más populares como ISO/IEC 15504 y CMMI (Rahmani et al, 2016).

Informes generados de estas investigaciones, indican que las organizaciones siguen enfrentando problemas para la SPI y aumentan el riesgo de fracaso en pequeñas y medianas organizaciones (Sulayman et al, 2012; Larrucea et al, 2016), en específico en el modelo CMMI Dev v1.3. Sin embargo, las organizaciones dan un alto valor a la mejora de sus procesos, de tal forma que invierten grandes cantidades de recursos y esfuerzos en programas para la implementación de mejoras de procesos de software no solo en la implementación del modelo CMMI Dev, sino a través de diversos modelos y estándares que les permita mantener una alta calidad en sus productos y servicios (Larrucea et al, 2016).

Por lo tanto, este artículo propone la identificación de caminos para iniciar la implementación de mejora de procesos de software en organizaciones de desarrollo de software a través de una propuesta de herramienta nombrada “HiSPI” para indicar el inicio efectivo de SPI, con base en un sustento matemático que identifique la importancia de las áreas de proceso en el Modelo CMMI Dev (Mejia et al, 2017). Esto permitirá, se establezca el inicio de SPI de acuerdo a los objetivos de negocio de la organización, según sus dependencias explícitas, lo cual, minimizará el esfuerzo y tiempo promedio en la SPI y aumentar la probabilidad de éxito en el nivel 2 tras una evaluación SCAMPI.

Este artículo está estructurado de la siguiente manera: en la segunda sección, se muestran los trabajos relacionados en iniciativas de SPI, en la tercera sección, la herramienta propuesta, finalmente en la cuarta sección, las conclusiones de las secciones anteriores y el trabajo futuro posible a desarrollar.

2. Trabajos relacionados

En el desarrollo de este estudio, se han identificado que existe un déficit en herramientas automatizadas que faciliten el cómo iniciar la implementación de programas de SPI (Khokhar, 2010; Miler & Wesolowska, 2012). Con base a una revisión literaria (Mejia et al, 2017) se detectaron algunas herramientas que permiten la SPI en las organizaciones:

La herramienta CIP-UQIM fue propuesta por los autores Hosein Rahmani, Ashkan Sami y Abdullah Khalili CSE con la finalidad de establecer un modelo para la armonización de las prácticas comunes entre los modelos CMMI-Dev, ISO 9001 y PMBOOK. Esta propuesta fue desarrollada para SMEs (muy pequeñas empresas) para realizar la

integración de las actividades del proceso, el cual, fue dirigido por un marco de trabajo que permitiera reducir las opiniones personales y determinar resultados más confiables (Khokhar, 2010; Miler & Wesolowska, 2012).

Esta herramienta consiste en tomar una base partir de un mapeo entre los modelo mencionados anteriormente, donde se obtenga la reutilización de procesos existentes de cualquier modelo que se tenga definido, de tal manera que se evite la implementación de procesos desde cero, desperdiciando tiempo y reutilización de otros recursos en otros programas, la finalidad de este modelo es aumentar la productividad y la probabilidad de éxito en SPI (Rahmani, 2016).

Marina Pilar, Jocelyn Simmonds y Hernán Astudillo autores de una herramienta Semi automatizada de un marco de trabajo enfocado en establecer una guía o modelo de herramientas para mejorar y mantener el desarrollo de procesos de software. La propuesta de esta herramienta toma como entradas un dominio, definición de procesos y un conjunto de criterios de evaluación de herramientas. Como resultado, se muestra una lista clasificada de herramientas para mejora de procesos.

Esta herramienta está basada en técnicas de Toma de Decisiones Multicriterios, específicamente Proceso analítico jerárquico (AHP) y Teoría de la utilidad basada en atributos múltiple (MAUT) para lograr reducir la cantidad significativa de tiempo que es requerida por cada usuario en la identificación de aspectos clave en cada proceso. Ha sido validada al usarlo para recomendar herramientas en un proceso y proyecto real, que ya se encuentran previamente documentados y de esta manera una evaluación de esta propuesta, la cual, fue realizada de manera empírica por expertos en implementación de mejoras de procesos (Pilar, 2014).

García et al, (2011) autores de la herramienta RIA realizan una aplicación para tomar iniciativas en implementación de mejoras de procesos de software, basados en un diseño colaborativo de modelos, donde se propone una estrategia para la evaluación y modelización de procesos colaborativos. Esta propuesta permite evaluar y modelar de forma colaborativa los procesos de software en un proyecto de SPI, a partir de una infraestructura y una estrategia distribuida y colaborativa que no está enfocada solo en la base del conocimiento acerca del desarrollo de procesos de software, sino que también permite comunicar y organizar el trabajo a realizar.

La herramienta HProcessTool fue desarrollada por los autores Pardo et al, (2011) con la finalidad de gestionar, supervisar y controlar proyectos a través de guías que indican paso a paso como establecer la armonización de múltiples modelos para cualquier tipo de compañía, obteniendo una optimización del tiempo, esfuerzo y recursos empleados en la armonización de un proyecto.

Esta herramienta ha sido utilizada en dos casos de estudio en los cuales uno de ellos está enfocado a la armonización de ISO 27001 y ISO 20000 con el objetivo de identificar las diferencias y similitudes entre estos estándares, de tal forma que se resuelvan sus discrepancias para facilitar la obtención de la certificación de la organización en el estándar ISO 20000 (Pardo et al, 2011).

Como resultado del análisis de los trabajos relacionados identificados en la revisión sistemática y de las herramientas detectadas para la mejora de procesos de software

en evaluación de procesos, se determina que las propuestas para la implementación de mejoras de proceso de software son a través de los siguientes puntos:

- Cuestionarios y entrevistas aplicados por expertos en mejora de procesos para afirmar el correcto seguimiento del modelo CMMI-Dev v1.3, donde los resultados son evaluados a través diversas sumatorias.
- Todas las propuestas de modelos o herramientas están basadas en el modelo IDEAL y a los resultados previos que cada uno proporciona a través de sus diversas técnicas, el cual están enfocadas al logro de los objetivos de las áreas de proceso del modelo CMMI-Dev v1.3.

3. HiSPI: Herramienta para la identificación de caminos para la implementación de mejora de procesos de software

El modelo propuesto para el desarrollo de la herramienta HiSPI está enfocado al modelo CMMI-DEV v1.3, para el nivel de madurez 2, con base en un método matemático que permite identificar como iniciar la implementación de mejoras de proceso de software partiendo de sus objetivos de negocio. El desarrollo de esta propuesta de herramienta, pretende solventar la problemática de las en las organizaciones de desarrollo de software que tienen como objetivo mejorar sus procesos, donde, no identifican cómo iniciar la mejora y no solo obtener la base de conocimiento en técnicas y métodos empíricos, propuestos por expertos en SPI.

El modelo de referencia propuesto en la herramienta HiSPI, permite establecer una guía a las organizaciones para la determinación de las alternativas para el inicio de SPI. Esta propuesta de modelo está formada por 5 fases. La Tabla 1 muestra de manera general las principales funcionalidades que abordan las 5 fases que conforman a la herramienta.

Funcionalidad	Descripción
Conocimiento del entorno organizacional.	Consiste identificar los objetivos de negocio de la organización relacionados SPI.
Trazabilidad con áreas de proceso CMMI Dev v.1.3.	Define una relación de los objetivos de negocio con respecto a las relaciones explícitas que existen entre las áreas de proceso.
Análisis actual de la organización.	Definir el entorno con áreas de proceso CMMI Dev v.1.3 (estado actual de la organización)
Determinar ruta óptima.	Con base a los objetivos de negocio y la trazabilidad identificada las áreas de proceso las rutas sugeridas teniendo en cuenta el modelo matemático descrito
Adaptabilidad de rutas.	Realizar la adaptabilidad de las rutas sugeridas según convenga al usuario, indicando la mejor alternativa según convenga al modelo.
Seguimiento de iniciativa de SPI.	Realizar el seguimiento de la implementación de SPI.

Tabla 1 – Funcionalidad de la herramienta.

La Figura 1, muestra la interfaz de bienvenida de la herramienta HiSPI para iniciar sesión, asimismo para la creación de una cuenta.



Figura 1 – Interfaz de bienvenida.

3.1. Conocimiento del entorno organizacional

En esta primera fase de la herramienta se determinan los objetivos de negocio que la organización tiene enfocados al inicio de SPI. Para ello, se realiza la introducción de los datos de la organización dentro del sistema, como se muestra en la Figura 3, y a continuación se procede al registro de los objetivos de negocio. Para realizar un registro de un nuevo Objetivo de Negocio (ON), se utilizan los elementos estructurales que sugiere la metodología GQM (Goal Question Metric) propuesta por Víctor Basili (Basili et al, 2007), para el proyecto de SPI, como se muestra en la Figura 4. La estructura de los ON debe definirse a partir de 5 elementos: Objetivo, Propósito, Enfoque, Punto de vista, entorno. Una vez obtenidos los objetivos se obtiene la trazabilidad con respecto al modelo CMMI, haciendo uso de un analizador léxico para definir la relación con las definiciones de las áreas de proceso, las cuales están descritas por las hojas de ruta que el modelo CMMI propone como guía para una efectiva adopción del modelo CMMI–Dev (Cannegieter et al, 2008).



Figura 2 – Interfaz de entorno organizacional (1/2).



Figura 2 – Interfaz de entorno organizacional (2/2).

3.2. Trazabilidad con áreas de proceso CMMI Dev V.1.3.

Una vez que se han identificado cada uno de los objetivos de negocio con la estructura de los cinco elementos de GQM, en la siguiente actividad de realiza un mapeo de estos con la definición de las áreas de proceso y el RoadMaps que el modelo CMMI sugiere para la iniciación de SPI.

Esta segunda fase de la herramienta permite visualizar a través de grafos una trazabilidad de las áreas de proceso que conforma el modelo CMMI–DEV para el nivel de madurez 2, en la cual, se indica las dependencias explícitas que existen entre cada una de ellas, por medio de la palabra reservada “Refer to”, identificando los 4 niveles principales (Mejia et al, 2017), donde se localiza la relación explícita de estas áreas de proceso, como se muestra en la Tabla 2.

Dependencia	Definición
Subpráctica a Subpráctica	Esta dependencia es establecida para indicar que para completar una Subpráctica es necesario realizar otra Subpráctica que se hace referencia.
Subpráctica a Práctica Específica	Esta dependencia es establecida para indicar que para completar una Subpráctica se tiene que realizar una práctica específica. Esta práctica específica puede contener a su vez una o más Subprácticas.
Subpráctica a Meta Específica	Esta dependencia es establecida para indicar que para completar una Subpráctica se tiene que realizar una meta específica. Esta meta específica puede contener a su vez una o muchas prácticas y Subprácticas específicas que cumplen con un objetivo específico del área de proceso
Subpráctica a Área de Proceso	Esta dependencia es establecida para indicar que para completar una Subpráctica se tiene que realizar un área de proceso. Esta área de proceso puede contener metas, prácticas y Subprácticas específicas determinadas para el logro del área de proceso.

Tabla 2 – Identificación de dependencias entre áreas de proceso.

El resultado obtenido permite definir una guía para enfocar el inicio de SPI, como se puede apreciar en la Figura 3.



Figura 3 – Trazabilidad con CMMI-Dev v1.3.

3.3. Análisis actual de la organización.

En esta fase de la herramienta se realiza un análisis de todos los procesos que ya están establecidos por la organización. Este análisis es realizado a partir de un cuestionario donde se verifica el cumplimiento de cada una de las prácticas de cada área de proceso. Este cuestionario es conforme al proceso de evaluación SCAMPI, donde se requiere comprobar las prácticas con alguna evidencia generada a lo largo del proceso.

Como resultado de este cuestionario se genera un grafo del estado actual de los procesos con respecto a otras áreas. Además se genera el estado que se encuentra la organización respecto al nivel de madurez que se pretende obtener.

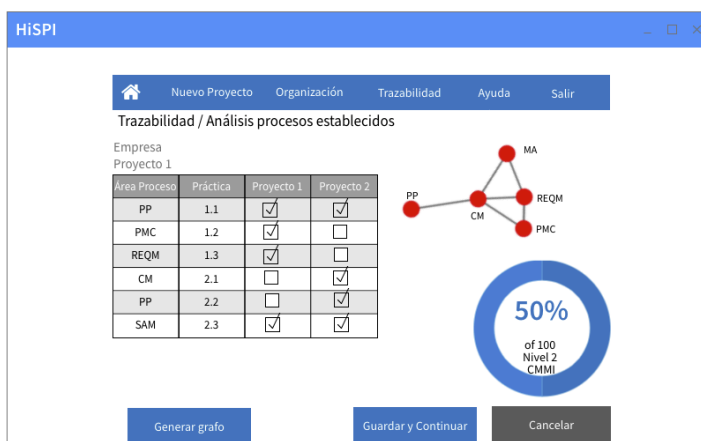


Figura 4 – Resultados de trazabilidad con CMMI-Dev v1.3.

3.4. Determinar ruta óptima.

De acuerdo a los resultados obtenidos en las dependencias identificadas en la Fase 2 y los objetivos de negocio detectados en la Fase 1, se construye un grafo que representa cual es la alternativa efectiva para el inicio de SPI seguido por el modelo CMMI-Dev v1.3.

En este grafo se identifica el peso resultante que es calculado para cada área de proceso a partir de una fórmula matemática. La fórmula identifica el valor de cada dependencia determinada por la trazabilidad entre áreas de proceso, a través de los resultados del análisis, ingeniería inversa y relaciones explícitas de métodos, modelos, herramientas utilizados tanto en implementación como en evaluaciones de programas de mejora de procesos de software (Mejia et al, 2017).

$$PA = \sum_{i=1}^n \frac{nr}{npe} \quad (1)$$

Nomenclatura:

nr = número de relaciones.

npe = número de Subprácticas a evaluar por área de proceso.

n = número de áreas de proceso correspondientes a un nivel.

PA = peso de la dependencia del área de proceso.

Partiendo de la importación del análisis de las dependencias entre áreas de proceso y la trazabilidad de los objetivos de negocio con respecto a CMMI, se realiza una comparación generando la ruta que es más efectiva para dar inicio a la implementación de mejoras de procesos de software como puede visualizarse en la Figura 5.

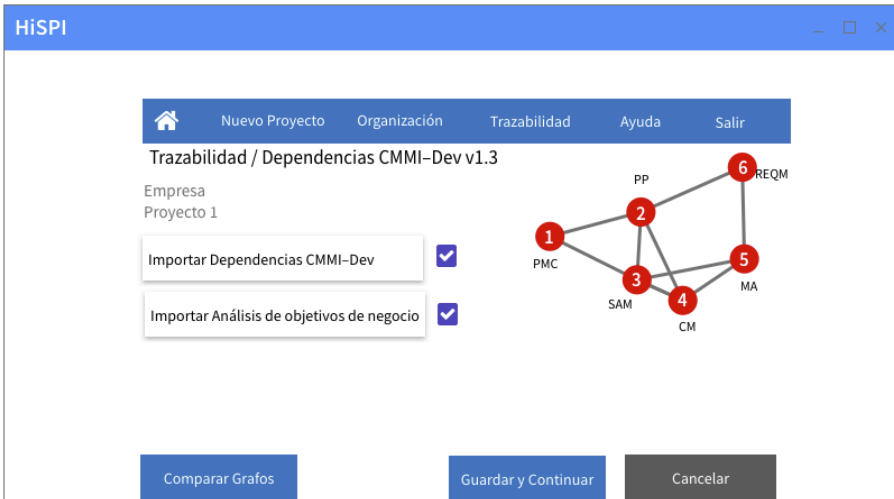


Figura 5 – Ruta óptima.

3.5. Adaptabilidad de rutas.

La funcionalidad de adaptabilidad de rutas, permite que el usuario pueda realizar ajuste a la propuesta que la herramienta ha generado con base a su análisis. Y con ello se realiza una comparación entre la propuesta y la personalizada, dando como resultado el porcentaje de esfuerzo que implica realizar la mejora de procesos con respecto a la propuesta de la herramienta.

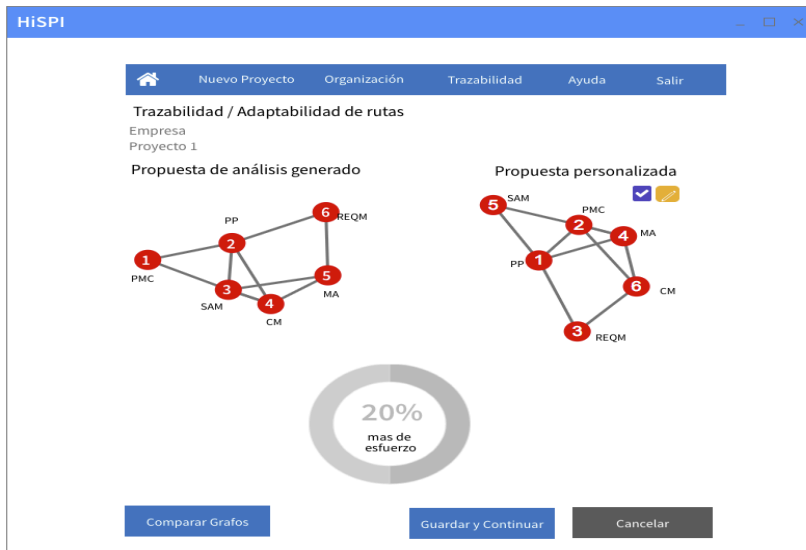


Figura 6 – Adaptabilidad de rutas.

3.6. Seguimiento de iniciativa de SPI.

Esta última fase de la herramienta permite realizar un seguimiento a la implementación de las mejoras de proceso de software, a través de un lista de verificación (checklist) con base en la herramienta Apraisal Asistents y herramientas utilizadas en evaluaciones SCAMPI para la validación y verificación de las prácticas que el modelo CMMI sugiere (Richter & Escorial, 2014).

Este seguimiento se realiza por cada una de las áreas de proceso donde se encuentra involucrado el proyecto de iniciativa de SPI. El proceso registra el logro de las prácticas a través de las evidencias generadas cuando se lleva a cabo un proceso de evaluación formal. Al finalizar el seguimiento de una área de proceso es posible realizar una previa evaluación observando el resultado obtenido en cada momento y guardar ese resultado generado, de tal forma que se mantenga el monitoreo a cada una de las áreas de proceso hasta culminar la totalidad de su implementación.

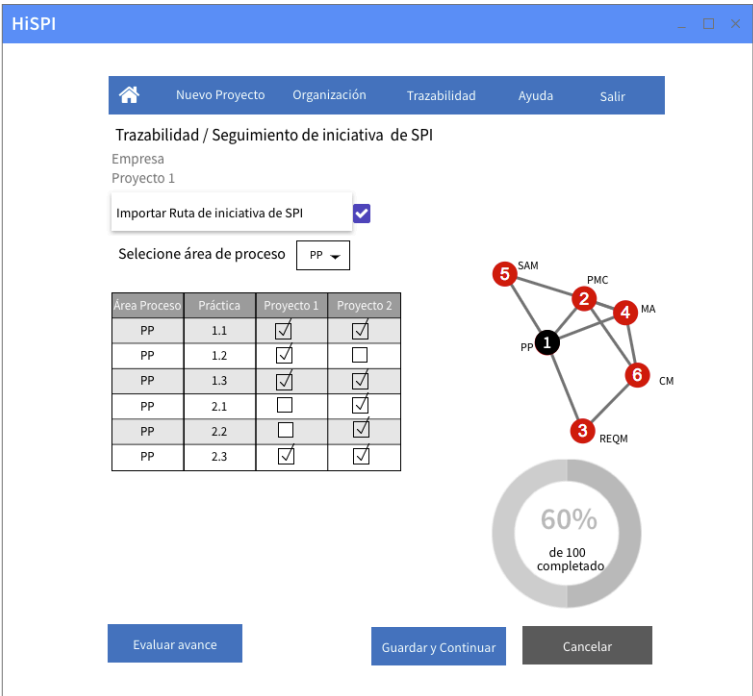


Figura 7 – Seguimiento de iniciativa de SPI.

La herramienta HiSPI forma parte de un caso de estudio que se tiene planificado implementar en una empresa de desarrollo de software, dicha empresa actualmente tiene el modelo de calidad CMMI Dev en el nivel 2 de madurez y pretende que se mantenga la certificación y se pueda preparar para el siguiente nivel de madurez.

Esta herramienta será la base que tome el personal a cargo del área de procesos. La herramienta servirá para tomar la decisión de como comenzar el proyecto de SPI. Además, la experiencia en su uso que se obtenga se usará para realizar mejoras a la herramienta y conseguir una primera versión estable.

4. Conclusiones

HiSPI es un marco de trabajo que sugiere alternativas para lograr identificar como iniciar la implementación de mejora de procesos de software, siguiendo el enfoque de los objetivos de negocio que las organizaciones establecen para la mejora de procesos. HiSPI muestra alternativas que pueden ser modificadas y obtener un análisis del esfuerzo que conlleva el aplicar dicha modificación para una eficaz iniciación de SPI formalizado para el nivel 2 de CMMI-Dev v1.3.

Asimismo, apoyado en la herramienta permite obtener una visualización de los requerimientos necesarios para iniciar los programas de SPI, de tal forma, que se pueda

realizar un análisis de los procesos definidos con respecto a los requisitos por el modelo. Y además, permite realizar un seguimiento al cumplimiento de la implementación de las buenas prácticas que requiere cada una de las áreas de proceso.

Como trabajo futuro dentro de esta investigación se propone realizar la formalización de los niveles 3, 4 y 5 de madurez del modelo CMMI, considerando técnicas o herramientas de predicción de datos, que son requeridas para estos niveles de mayor complejidad. Igualmente, se podría considerar la automatización de las fases de análisis de procesos ya establecidos y el seguimiento de implementación de SPI a través de la recolección de la información y las evidencias que son requeridas al realizar una evaluación formal.

Agradecimientos

El desarrollo de esta publicación es patrocinado por el Centro de Investigación en Matemáticas, A.C.

Referencias

- Basili, V., Heidrich, J., Lindvall, M., Münch, J., Regardie, M., & Trendowicz, A. (2007). GQM+Strategies - Aligning business strategies with software measurement. In *Proceedings of 1st International Symposium on Empirical Software Engineering and Measurement, ESEM 2007*, (pp. 488-490). <https://doi.org/10.1109/ESEM.2007.56>.
- Cannegieter, J. J., Heijstek, A., Linders, B., & Van Solingen, R. (2008). *CMMI roadmaps, (November)*. Retrieved from <http://repository.cmu.edu/sei/295/>.
- Garcia, I., Pacheco, C., & Cruz, D. (2011). Implementation of a RIA Tool for Supporting a Collaborative Initiative of Software Process Improvement. In *Proceedings of International Conference on Cooperative Design, Visualization and Engineering*, (pp. 45–52). Mallorca, Spain: Springer.
- Khokhar, M. N., Rehman, S. U., Mansoor, A., Khokhar, M. N., & Rauf, A. (2010). MECA: Software process improvement for small organizations. In *Proceedings of 2010 International Conference on Information and Emerging Technologies*. <https://doi.org/10.1109/ICIET.2010.5625678>.
- Larrucea, X., O'Connor, R. V., Colomo-Palacios, R., & Laporte, C. Y. (2016). Software process improvement in very small organizations. *IEEE Software*, 33(2), 85–89. <https://doi.org/10.1109/MS.2016.37>
- Mejia, J., Gonzalez, M., & Muñoz, M. (2017). Organization of the process areas of CMMI-Dev v1.3 level 2 through of its dependencies. In *Proceedings of 2017 12th Iberian Conference on Information Systems and Technologies* (pp. 1-7). Lisbon: IEEE. doi:10.23919/CISTI.2017.7975919.
- Miler, J., & Wesołowska, H. (2012). Improvement of Task Management with Process Models in Small and Medium Software Companies. In *Proceedings of European Conference on Software Process Improvement* (pp. 145-156). Ostrava, Czech Republic: Springer. https://doi.org/10.1007/978-3-642-31199-4_13.

- Pardo, C., Pino, F., García, F., Romero, F. R., Piattini, M., & Baldassarre, M. T. (2011). HProcessTOOL: A support tool in the harmonization of multiple reference models. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, (PART 5, pp. 370–382). https://doi.org/10.1007/978-3-642-21934-4_30.
- Pilar, M., Simmonds, J., & Astudillo, H. (2014). Semi-automated tool recommender for software development processes. *Electronic Notes in Theoretical Computer Science*, 302, 95–109. <https://doi.org/10.1016/j.entcs.2014.01.022>
- Rahmani, H., Sami, A., & Khalili, A. (2016). CIP-UQIM: A unified model for quality improvement in software SME's based on CMMI level 2 and 3. *Information and Software Technology*, 71, 27–57. <https://doi.org/10.1016/j.infsof.2015.10.009>
- Sulayman, M., Urquhart, C., Mendes, E., & Seidel, S. (2012). Software process improvement success factors for small and medium Web companies: A qualitative study. *Information and Software Technology*, 54(5), 479–500. <https://doi.org/10.1016/j.infsof.2011.12.007>
- Richter S., Escorial R. D. (2014). Software Process Assessment Validation and Improvement Prioritization. In: Barafort B., O'Connor R.V., Poth A., Messnarz R. (eds). *Systems, Software and Services Process Improvement. EuroSPI 2014. Communications in Computer and Information Science*, (vol 425). Berlin, Heidelberg: Springer.

Implementación de un enfoque DevSecOps + Risk Management en un Centro de Datos de una organización Mexicana

Oswaldo Díaz¹, Mirna Muñoz²

oswaldo.diaz@inegi.org.mx, mirna.munoz@cimat.mx

¹ Grupo de Ingeniería de Sistemas, Instituto Nacional de Estadística y Geografía (INEGI) Av. Héroe de Nacozari no 2301, Aguascalientes, México.

² Centro de Investigación en Matemáticas – Unidad Zacatecas, 98068, Zacatecas, Zacatecas, México

DOI: 10.17013/risti.26.43–53

Resumen: En México la importancia de los Centros de Datos ha incrementado el interés en enfoques DevOps (Developer Operations). Implementar un enfoque DevOps en Centros de Datos permite establecer estrategias para la gestión de operaciones automatizadas, ingeniería de software y aseguramiento de la calidad. Este artículo presenta la implementación de un enfoque evolucionado de DevOps nombrado como DevSecOps+ Risk Management, que además de lo antes mencionado permite establecer estrategias para gestionar la seguridad informática y gestión de riesgos. El artículo incluye tanto la descripción del enfoque, cómo su implementación para el desarrollo de proyectos en un centro de datos de una empresa gubernamental mexicana. Los resultados han demostrado que este enfoque apoya en la finalización exitosa de proyectos críticos de la organización.

Palabras-clave: DevOps, Ingeniería de Software, Control de Calidad, Infraestructura Tecnológica, Centro de Datos, Seguridad informática, Gestión de Riesgos.

Implementation of a DevSecOps + Risk Management in a Data Center of a Mexican Organization

Abstract: In Mexico the importance of Data Centers has increased the interest on DevOps (Developer Operations) approaches. The implementation of a DevOps approach in a Data Center allows establishing strategies for managing automatize operations, software engineering and quality assurance. This paper presents the implementation of an evolve approach of DevOps named as DevSecOps + Risk Management, which in addition to the above mentioned allows establishing strategies for managing information security and Risk Management. The paper includes both, the description of the approach as well as its implementation for developing projects in a Data Center in a large government organization of Mexico. The results have demonstrated that this approach support the successful closure of critical projects within the organization.

Keywords: DevOps, DevSecOps, Software engineering, quality control, Technology infrastructure, Data Centers, Informatics security, Risk Management.

1. Introducción

En los últimos años los Centros de datos han evolucionado significativamente en todo el mundo y México no es la excepción, ya de acuerdo a (Judge, 2016) es un país que cuenta con alrededor del 20% de los Centro de Datos existentes en América Latina. Estos Centros de Datos tienen necesidades específicas al ser los que albergan los proyectos considerados “de misión crítica” para las organizaciones.

En este contexto, surge lo que se conoce como “DevOps”, que es una mezcla de dos palabras, desarrolladores y Operaciones que surge de la necesidad de construir sistemas de software reduciendo el tiempo de entrega del mercado, y presentando una solución al problema de optimización del ciclo de vida de entrega (Virmani, 2015). La importancia de DevOps incrementa cada vez más, ya que se enfoca en reforzar la cooperación entre el profesional de TI y los desarrolladores de software para reducir el tiempo de entrega del proyecto, permitiendo establecer estrategias para la gestión de flujos de trabajo, control de versiones y entrega de productos de software.

Una de las empresas que cuentan con un Centro de Datos en México es Instituto Nacional de Estadística y Geografía (INEGI), que es el instituto encargado de las siguientes actividades: (1) regular y coordinar el sistema de información estadística y geográfica; (2) llevar a cabo censos nacionales; (3) integrar el sistema de cuentas nacionales y elaborar los índices nacionales de precios al productor y (4) ser la agencia cartográfica nacional de México.

El Centro de Datos de INEGI, por lo tanto desarrolla sistemas de información en múltiples tecnologías como son para internet, dispositivos móviles y servicios en la nube, entre otras; con la finalidad de diseminar información estadística y geográfica que permitan al gobierno mexicano la toma de decisiones. Para lograr el desarrollo de estos proyectos con éxito, INEGI decide implementar DevOps como una medida para el logro de los objetivos del centro de datos (Muñoz & Díaz, 2016). Sin embargo, después de su implementación y análisis de sus resultados, decidieron reforzar este enfoque con seguridad informática y gestión de riesgos surgiendo DevSecOps+Risk Management (Díaz & Muñoz, 2017).

El objetivo de este artículo es presentar cómo fue reforzado el enfoque DevOps con seguridad informática y gestión de riesgos, así como el análisis de los resultados de su implementación para el desarrollo de proyectos de misión crítica Centro de Datos de INEGI.

El artículo está organizado de la siguiente forma: en la sección 2 se describen conceptos clave. La Sección 3 presenta una visión global del enfoque. La Sección 4 describe cómo se refuerza el enfoque con seguridad informática y gestión de riesgos. En la sección 5 se presenta el análisis de los resultados de su implementación y, finalmente, la sección 6 presenta las conclusiones y trabajo futuro.

2. Conceptos clave

En esta sección se definen los conceptos considerados clave para comprender el contexto de este artículo.

- *Centro de Datos*: es una instalación dentro de una organización encargada de centralizar las operaciones el equipo de TI enfocando en el almacenamiento,

administración y despliegue de la información de la organización. (Robertazzi M., 2012).

- *DevOps*: término compuesto de dos palabras *desarrollo* y *operaciones*, y surge de la necesidad de construir sistemas de software optimizando el ciclo de desarrollo del producto hasta su entrega (Virmani, 2015).
- *Seguridad de información*: se refiere al uso de estándares, procedimientos, métodos y técnicas enfocadas en lograr sistemas de información seguros y confiables, teniendo en cuenta la integridad, confidencialidad y disponibilidad de la información (Betancourt & Valverde, 2015).
- *Gestión de Riesgos*: comprende la identificación de problemas potenciales antes de que éstos ocurran, tal que se pueden planificar actividades para gestionar los riesgos, las cuales pueden ser invocadas a través de todo el ciclo de desarrollo de un producto o proyecto, tal que puedan ser mitigados los impactos adversos que impidan el logro de los objetivos (CMMI Product Team, 2010).

3. Enfoque DevSecOps + Risk Management

3.1. Antecedentes

El enfoque DevSecOps+ Risk Management surge como una evolución del enfoque DevOps para cubrir las necesidades de proyectos de misión crítica alojados en el centro de datos.

INEGI toma la decisión de evolucionar el enfoque DevOps (Muñoz & Díaz, 2016) implementado para cubrir dos principales necesidades:

1. Reducir las vulnerabilidades que ponen a la información estadística y geográfica en riesgo de robo o secuestro
2. Gestionar los riesgos a través de todo el ciclo de desarrollo del proyecto / producto para reducir eventualidades lógicas o físicas.

Este nuevo enfoque, por lo tanto, se fortalece con seguridad informática y gestión de riesgos como a continuación se describe (Díaz & Muñoz, 2017):

1. *Seguridad Informática*: se ocupa de diseñar normas, procedimientos, métodos y técnicas destinados a conseguir un sistema de información seguro y confiable, tomando en cuenta la integridad, confidencialidad y disponibilidad de la información alojada en un centro de datos de la organización o empresa.
2. *Gestión de Riesgos*: consiste en aumentar la probabilidad y el impacto de los eventos positivos (ejemplo: más usuarios concurrentes), y disminuir la probabilidad y el impacto de los eventos negativos (ejemplo: eventualidades las cuales el usuario experimente interrupción en el servicio) del proyecto orientado a tecnologías de información implementado en la organización o empresa.

3.2. Fases del enfoque

El enfoque DevSecOps+ Risk Management se compone de 4 fases, todas llevadas cabo dentro de un marco de gestión de riesgos, las cuales se muestran en la Figura 1 y se describen a continuación.

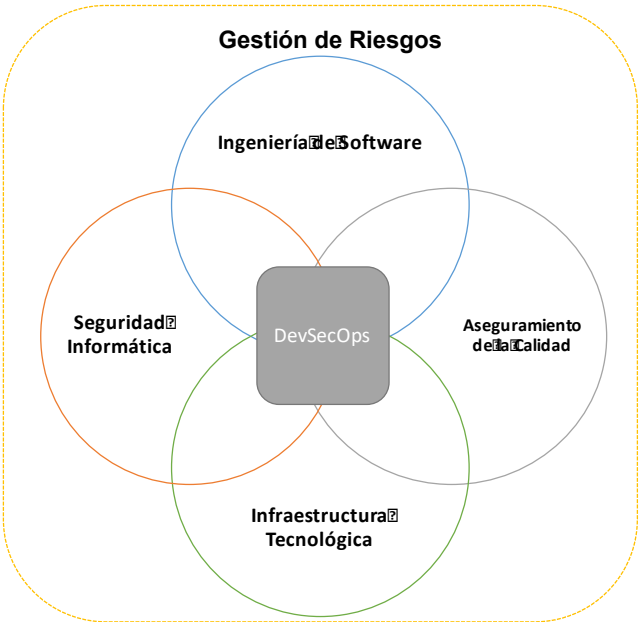


Figura 1 – Enfoque DevSecOps + Risk Management fases

- 1. Ingeniería de Software: esta fase se enfoca en el desarrollo de software. Para lograr el desarrollo de software se usa la metodología Scrum (Gloger, 2014). como una solución para eliminar liberaciones lentas.

El flujo de trabajo de Scrum definido en el enfoque contiene 6 fases y 4 equipos como se muestran en las Tablas 1 y 2.

Fase	Descripción	Equipo Responsable
Diseño	En esta fase se definen las reglas de cumplimiento y un cronograma con fechas de compromisos, de acuerdo con los requisitos del proceso a ser automatizado para publicarlos en un ambiente controlado por TI.	Equipo de desarrollo
Desarrollo	En esta fase se construye el proceso automatizado basado en prácticas de ingeniería de software y teniendo en cuenta el hardware definido como parte del “Diseño”, de manera que un avance significativo llamado “Beta” se entrega para interactuar con el proceso de prueba.	Equipo de desarrollo
Pruebas	En esta fase se realiza un plan estratégico y de pruebas específicas con base en los requisitos del proyecto y teniendo en cuenta la versión “Beta” y la retroalimentación de todas las eventualidades detectadas.	Equipo de control de calidad
Despliegue	En esta fase se publica y controla la versión del proyecto denominada “Release” en un entorno de TI controlado.	Equipo de gestión de implementación

Fase	Descripción	Equipo Responsable
Retrospectiva	En esta fase se realiza una evaluación del análisis de riesgos basado en las eventualidades detectadas en la fase de pruebas y toma las decisiones para mitigar o enfrentar los riesgos	Equipo de desarrollo
Recolección de requisitos y retroalimentación	En esta fase se realiza una reunión interactiva cara a cara, para determinar el desempeño del proyecto, tanto de progreso como de demora, con el fin de acelerar los procesos y procedimientos para que el calendario de entrega se pueda cumplir	Equipo de revisión y control

Tabla 1 – Fases Scrum definidas

Fase	Descripción	Roles involucrados
Equipo de desarrollo	Equipo responsable del desarrollo de la arquitectura de la estrategia y de la arquitectura de la base de datos.	Ingenieros de software y Arquitectos para el modelado de los daos con base al requerimiento del proceso del negocio.
Equipo de revisión y control del sistema	Equipo responsable de la integración continua, la gestión de versiones y el control de repositorio de versiones tomando en cuenta la estrategia de proceso y calendario de actividades.	Revisores de la integridad en la informacion con base a normas y cumplimientos de la organización o empresa
Equipo de control de calidad	Equipo responsable de llevar a cabo las pruebas de integración, seguimiento de soluciones a problemas en eventualidades de seguridad y percepción de servicio orientado al usuario final y la conformación de base de datos de conocimiento en base a incidencias.	Arquitectos en tecnologías de informacion revisores del control de calidad con base a niveles de servicios en la operación.
Equipo de gestión en la implementación	Equipo responsable de administrar y mantener el funcionamiento de la infraestructura de los sistemas operativos; sistemas de almacenamiento, sistemas de procesamiento, sistemas de respaldos y sistemas de comunicaciones; que habilitan la publicación de los proyectos en el ambiente de producción.	Ingenieros de procesos con base en los roles [Webmaster, SysAdmin, DBA entre otros]

Tabla 2 – Equipos definidos

Cabe mencionar que puede formarse un solo equipo de trabajo que mediante roles desempeñe las funciones descritas en los diferentes equipos.

2. *Fase de Control de Calidad:* esta fase cubre temas relacionados con la gobernanza, construcción, verificación e implementación. La fase contiene 4 proceso definidos con base en OWASP Software Assurance Maturity Model (Owasp SAMM) (Carphey & Groves 2015). El objetivo de cada proceso se describe a continuación.
 - *Proceso de gobernanza,* permite que se definan indicadores cuantitativos y cualitativos basados en las políticas y controles de cumplimiento, de manera que se generen tableros de control para la toma de decisiones.

- Proceso de construcción, permite que establezcan los niveles de seguridad teniendo en cuenta las políticas internas y externas basadas en los estándares organizativos, así como las tendencias tecnológicas en hardware y software en los que la organización tenga madurez, de modo que la adopción de estos no implica un retraso en la ejecución de los proyectos de la organización
 - Proceso de verificación, permite que se establezca un plan de pruebas en conjunto con el líder del equipo y tener en cuenta los aspectos del proyecto con respecto a las mejores prácticas del motor de código y el perfil de rendimiento enfocado en el cumplimiento de las líneas base establecidas por la organización Modelo-Vista-controlador (MVC).
 - Proceso de Implementación, permite que se establezcan entornos tolerantes a fallos para la publicación de los proyectos, las vulnerabilidades gestión y excepciones basadas en las métricas de seguridad informática, la realización de endurecimiento 3 capas de nivel para los componentes que intervienen en el proyecto, tener un seguimiento de los procesos, recursos y del personal involucrado en la organización.
3. *Fase de Infraestructura Tecnológica para la Operación:* esta fase se enfoca en el cuidado de la infraestructura para el procesamiento de visualización, almacenamiento de la información, los requisitos específicos y estratégicos de transferencia de información. Su objetivo es difundir los productos o servicios generados de una manera segura y oportuna.

Para lograr una administración y control de riesgos en la infraestructura del proyecto se establecen: (1) entornos controlados para escenarios de desarrollo, preproducción y producción que tienen como objetivo separar las aplicaciones y las bases de datos con el fin de reducir el riesgo y; (2) niveles de servicios mediante la implementación de prácticas ITIL (ITIL/OGC, 2010), que permite establecer indicadores cualitativos y cuantitativos para generar información permitiendo la toma de decisiones basada en las mejores prácticas de tal forma para la gestión de la infraestructura de TI.

Las fases de Seguridad Informática y gestión de riesgos se describen en la siguiente sección.

4. Seguridad informática y Gestión de Riesgos

4.1. Seguridad Informática

La seguridad informática se refiere al uso de modelos, marcos de trabajo, estándares, metodologías, normas, técnicas, herramientas y estructuras organizacionales encaminadas a proteger la información en sus diferentes formas y estados (Muñoz & Rivas, 2015); (Mejía & Ramírez, 2016). En enfoque DevSecOps+Risk Management esta fase se enfoca en el robustecimiento del enfoque para proteger la información

Para lograr lo antes menciona, se define un decálogo que se basa en la adaptación del ISO/IEC 27005 (UNAM CERT, 2016) de acuerdo la experiencia y flujo de trabajo Instituto Nacional de Estadística y Geografía. Por lo que, tomando en cuenta la lógica del

negocio y con base en la misión y visión, gestiona la información que involucra la captura, integración, procesamiento, validación y difusión de la información. El decálogo se lista a continuación:

1. Establecer un programa de gestión de riesgos de la información formal que contenga políticas, procesos y soluciones, en áreas donde no se está realizando, permitiendo automatizar la recolección centralizada de riesgos y presentación de información.
2. Establecer una estrategia para realizar una analítica de datos de las bitácoras correlacionadas con eventos – reportes – de eventualidades reportadas.
3. Establecer procesos y procedimientos para evaluar periódicamente el conocimiento de los usuarios frente a vectores de ataques.
4. Implementar y administrar, un proceso formal de gestión de administración - reparación a eventualidades reportadas, inventario y manejo de software para servidores donde estén alojados los proyectos de información estratégica.
5. Implementar y administrar, políticas y procedimientos integrales para la gestión de vulnerabilidades, de preferencia en un laboratorio para pruebas de penetración externas con herramientas especializadas.
6. Establecer procesos de administración y gestión para la protección contra códigos maliciosos en la transferencia de información.
7. Establecer e Implementar una política, estándares y procesos para el hardening¹ de servidores físicos y virtuales.
8. Definición de un programa que incluya procesos, documentación y preparación del personal para evitar la fuga de datos, además de la implementación de un programa corporativo para la prevención de fuga de la información que incluya políticas, procesos, flujos de trabajo, soluciones y personal capacitado.
9. Implementar herramientas para automatizar políticas de seguridad actualizadas, con base a reportes en líneas de cumplimiento.
10. Implementar un plan de respuesta a incidentes, incluyendo la estructura del equipo, sus roles y responsabilidades, definir responsables y empezar a documentar procesos con base a la estrategia para la continuidad de operaciones y Plan de recuperación en caso de desastres.

4.2. Gestión de Riesgos

Esta fase cubre se establece con base en el estándar ISO/ IEC 27005 (UNAM CERT, 2016) y la Metodología de análisis de riesgos OCTAVE ALLEGRO (UNAM CERT, 2016) como a continuación se describe:

Marco de trabajo con 8 actividades con base ISO/IEC 27005:

- Establecimiento del contexto de los servicios.
- Identificación del riesgo en los servicios.
- Estimación del riesgo con indicadores cuantitativos y cualitativos.
- Valoración del riesgo base a métricas de control.
- Tratamiento del riesgo en colaboración con las áreas involucradas.
- Aceptación del riesgo de forma conjunta con las áreas involucradas.
- Comunicación del riesgo – afectación y recuperación.

- Monitoreo y revisión del riesgo, así como también, su registro en una base de conocimientos para aprender de las eventualidades.

Metodología de análisis de riesgos de 4 fases con base en OCTAVE Allegro:

- *Establecimiento de controles*: establecer criterios de las métricas de riesgo para cada servicio–proceso del proyecto.
- *Perfil de activos*: desarrollar el perfil de los activos de información e identificar los contenedores de los activos de información para su resguardo e integridad.
- *Identificación de amenazas*: identificar amenazas en las áreas de preocupación, tomando en cuenta escenarios de amenaza internas y externas.
- *Identificación y mitigación de riesgos*: analizar eventualidades reportadas en el servicio con base a la selección de metodologías para la mitigación.

5. Análisis de los resultados de su implementación

Esta sección muestra la implementación del enfoque DevSecOps + Risk Management para un Centro de Datos de una organización Mexicana.

Un centro de datos es una instalación clave en una organizaciones porque aloja los sistemas más críticos de la organización, por lo que es de vital importancia para la continuidad de la operación cotidiana (Kant & Mohapatra, 2004), razón por la cual tienen una alta prioridad tanto la fiabilidad como la seguridad (Callou et al., 2014).

De acuerdo a (Callou et al., 2014), aunque los diseños de centros de datos son únicos, pueden clasificarse de acuerdo con su dominio de aplicación en dos tipos: orientados a internet y empresariales o internos.

El centro de datos de esta empresa gubernamental Mexicana es de tipo empresarial o interno, que son aquellos que dan soporte a más aplicaciones que van desde aplicaciones estándares hasta aplicaciones personalizadas, y, tienen pocos usuarios bien conocidos (Callou et al., 2014).

La empresa gubernamental Mexicana es el Instituto Nacional de Estadística y Geografía (INEGI). Como parte de las actividades realizadas en el instituto, fue necesario implementar una estrategia para controlar la operación de los proyectos: Índice Nacional de Precios, México en Cifras, Sistema Nacional de Información Estadística y Geográfica, Banco de Información Económica, Mapa Digital de México, entre otros, considerados de misión crítica, porque tiene la característica de requerir disponibilidad de acceso por internet las 24 horas, los 365 días de año y con tolerancia a fallas, permitiendo que el usuario no experimente interrupciones en el servicio.

El enfoque DevSecOps + Risk Management permitió establecer un entorno de colaboración basado en las normas de ingeniería de software; las mejores prácticas de las TIC's y la infraestructura, implementando un ambiente controlado en la operación, obteniendo beneficios para administrar la operación en centros de datos en puntos geográficos diferentes, con esquemas de seguridad y administración de riesgos.

La estrategia de la infraestructura mediante la implementación del enfoque DevSecOps + Risk Management se muestra en la Figura 2 y se describe a continuación.

ROL \ Actividades	Fase de ingeniería de software	Fase de control de calidad,	Fase Infraestructura tecnológica	Fases de Seguridad informática y de Gestión de Riesgos
<i>Roles</i>	<i>Líderes de proyecto y desarrolladores.</i> <i>Administradores de servidores de aplicaciones.</i> <i>Administradores de sistema operativo.</i> <i>Arquitectos DBA modeladores de base de datos.</i>	<i>Establecer un equipo con el rol de Beta Tester, estas personas son responsables de aplicar las mejores prácticas en seguridad, escalabilidad, percepción y experiencia de usuario, tomando en cuenta entregar el informe de control de calidad.</i>	<i>Establecer la siguiente infraestructura para el desarrollo e implementación de proyecto:</i> <i>servidores de aplicaciones</i> <i>servidor – Clúster Base de datos (nodos) proceso</i> <i>en paralelo y tolerante a fallas.</i> <i>servidor de almacenamiento y copia de seguridad [con tecnología SAN y NAS]</i>	<i>Verificación de los 10 puntos mencionados en el apartado (fase de seguridad informática) tomando en cuenta el mapa de servicios de proyecto con base a procesos, personas y tecnología.</i> <i>Elaboración de matriz de riesgo personalizada del proyecto con puntos de control y actividades para remediar eventualidades.</i>
<i>Puntos de revisión de riesgos</i>	<i>Estrategias para cubrir los roles y no detener productividad.</i> <i>Adaptabilidad y tolerancia a fallas en la operación.</i> <i>Retroalimentación del equipo de trabajo involucrado en esta fase.</i>	<i>Seguimiento y control en base de datos de conocimientos referentes a las eventualidades (excepciones) reportadas en el flujo del proyecto, tomando en cuenta todos los módulos.</i> <i>Retroalimentación del equipo de trabajo involucrado en esta fase.</i>	<i>Seguimiento y control en las bases de datos de conocimientos tomando en cuenta las bitácoras del hardware y software involucrado en la operación, con la finalidad de tener elementos cuantitativos y cualitativos para la toma de decisiones en controles de cambios, incidencias y problemas en el centro de datos.</i> <i>Retroalimentación del equipo de trabajo involucrado en esta fase.</i>	<i>Documentación de la continuidad operativa del proyecto con base al proceso.</i> <i>Retroalimentación del equipo de trabajo involucrado en esta fase.</i>

Tabla 3 – Estrategia y Equipos de trabajo definidos implementando el enfoque DevSecOps + Risk Management

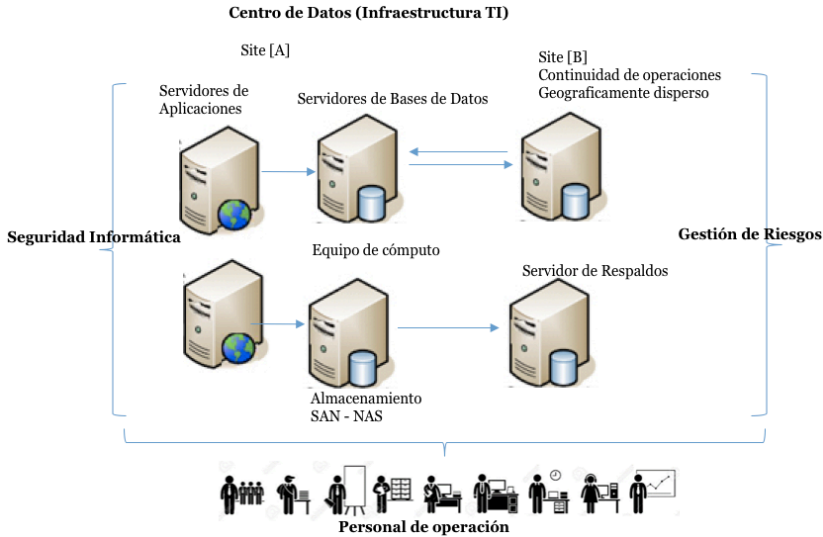


Figura 2 – Ejemplo Infraestructura de un Centro de Datos

Dentro de las fases descritas para la estrategia DevSecOps + Risk Management el equipo de trabajo colaborativo para mantener la operación se muestra en la Tabla 3.

6. Conclusiones y trabajo futuro

Hoy en día la importancia de los Centros de Datos en diferentes organizaciones se está haciendo más evidente, debido a que albergan los sistemas de misión crítica, por lo tanto, contar con estrategias que permitan lograr flujos de implementación y entrega efectivos es una creciente necesidad. Además, es necesario asegurar una alta disponibilidad y continuidad de la operación en estos centros.

En este contexto, el enfoque DevSecOps + Risk Management se presenta como una solución efectiva, permitiendo el control de la operación de las áreas involucradas en el desarrollo de un proyecto. Este enfoque ha permitido en INEGI generar valor agregado en los procesos de los proyectos de misión crítica, contemplando el ciclo de vida con niveles de servicios adecuados, manteniendo la continuidad operativa y permitiendo que el riesgo sea analizado cuantitativa y cualitativamente.

Referencias

- Betancourt, D., & Valverde, D. (2015). Implementación del SGSI (ISO/IEC 27001:2013). In *Talleres y líneas de especialización del Congreso de Seguridad en Cómputo 2015*. Mexico: UNAM.
- Callou, G., Ferreira, J., Maciel, P. Tutsch, D., & Souza, R. (2014). An Integrated Modeling Approach to Evaluate and Optimize Data Center Sustainability, Dependability and Cost. *Energies open access*, 2014 (7), 238–277.

- CMMI Product Team. (2010). *CMMI for development v1.3 Software Engineering Institute Technical Report CMU/SEI-2010-TR-033*.
- Curphey, M., & Groves, D. (2015). *OWASP-SAMM (Open Web Application Security Project - Software Assurance Maturity Model) a Guide to building security into software development Version 1.0*.
- Díaz, O., & Muñoz, M. (2017). Reinforcing DevOps approach with security and Risk Management: an experience of implementing it in a Data Center of a Mexican Organization. In *Proceedings of the 6th International Conference in Software Process Improvement*. Zacatecas, Mexico: IEEE.
- Gloger, B. (2014). Scrum checklist. All meetings.
- ITIL/OGC. (2010). *Scope and Development Plan Copyright TSO 2010*. Retrived from: www.ogc.gov.uk
- Judge, P., (2016). Datacenter Dynamics: Market Focus Latin America. *datacenterdynamics.com*, 2016(14).
- Kant, K., & Mohapatra, P. (2004). Internet Data Centers. *Computer. IEEE Computer Society*, 37(11) , 35–37.
- Mejía, J., & Ramírez, H. (2016). Estableciendo controles y perímetro de seguridad para una página web de un CSIRT. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (17), 1–15.
- Muñoz, M., & Díaz, O. (2017). DevOps: Foundations and Its Utilization in Data Center. In: Marx Gómez J., Mora M., Raisinghani M., Nebel W., O'Connor R. (eds). *Engineering and Management of Data Centers. Service Science: Research and Innovations in the Service Economy*. Berlin: Springer, Cham.
- Muñoz, M. & Rivas, L. (2015). Estado actual de equipos de respuesta a incidentes de seguridad informática. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (E3) 1–15.
- Robertazzi, M.T. (2012). Data Centers. In *Basics of Computer Networking*. In: *SpringerBriefs in Electrical and Computer Engineering* (pp 69-72). Berlin: Springer.
- UNAM CERT. (2016). *SGSI base ISO/IEC 27001:2013, Serie 27000 ,Anexo SL and ISO/IEC 27005, Manual de referencia*.
- Virmani, M. (2015). Understanding DevOps & Bridging the gap form continuous integration to continuous delivery. In *Proceedings of Fifth international conference on Innovative Computing Technology* (pp. 78–82). Galicia, España: IEEE.

Caracterización y Detección Automática de Bad Smells MVC

Perla Velasco-Elizondo¹, Lucero Castañeda-Calvillo², Alejandro García-Fernández³, y Sodel Vazquez-Reyes⁴

pvelasco@uaz.edu.mx, b160629@sagitario.cic.ipn.mx, agarciafdz@cimat.mx, vazquez@uaz.edu.mx

¹ Universidad Autónoma de Zacatecas, Zacatecas, ZAC, 98000, México.

² Centro de Investigación en Computación, Ciudad de México, 07738, México.

³ Centro de Investigación en Matemáticas, Zacatecas, ZAC, 98060, México.

⁴ Universidad Autónoma de Zacatecas, Zacatecas, ZAC, 98000, México.

DOI: 10.17013/risti.26.54-67

Resumen: Los *bad smells* son causa frecuente de acumulación de deuda técnica; término que se refiere al costo en que se incurre por utilizar un enfoque de diseño o desarrollo apresurado y descuidado. Existen trabajos sobre la caracterización de *bad smells*, así como sobre enfoques para detectarlos y corregirlos automáticamente. Sin embargo, pocos de estos trabajos caracterizan, detectan y corrigen *bad smells* arquitectónicos. Este trabajo es un esfuerzo inicial para llenar este vacío y contribuir en: (i) la caracterización de *bad smells* relevantes al estilo de arquitectura MVC, y (ii) la detección automática de estos *bad smells* utilizando técnicas de análisis estático de software. Los resultados obtenidos muestran que la mayoría de los *bad smells* definidos existen en la práctica, y que la estrategia de detección propuesta reduce en un amplio margen el tiempo requerido para detectar *bad smells* mediante una revisión de código de forma manual.

Palabras-clave: Arquitectura de Software; Bad Bad smells; Análisis Estático; MVC; Yii.

Characterization and Automatic Detection of Bad Smells MVC

Abstract: Bad smells are a frequent cause of technical debt, which denotes the cost of adopting a quick and dirty design or development approach. There are works on characterizing bad smells as well as on detecting and fixing them automatically. However, few of these works characterize, detect and fix architectural bad smells. The work presented in this article represents an initial effort to fill this by contributing to: (i) the characterization of bad smells that are relevant to the MVC architecture style, and (ii) the automatic detection of these using static analysis of software techniques. The obtained results show that most of the defined bad smells exist in practice and that the proposed detection method reduces by a wide margin the detection time required by a code review.

Keywords: Software Architecture, Bad Bad smells, Static Analysis, MVC, Yii.

1. Introducción

El éxito de un proyecto de desarrollo de sistemas no sólo depende de que el sistema creado se termine dentro plazo y presupuesto establecido, sino también que genere valor (Espejo Chavarría, Bayona Oré, & Pastor, 2016; Morales Huanca & Bayona Oré, 2017). Así, durante la actividad de diseño de sistemas de software, es muy importante definir un modelo que permita satisfacer los comportamientos y características de calidad especificadas durante la actividad de análisis de requerimientos. Para lograr esto es recomendable que se haga uso de conceptos de diseño, como por ejemplo los patrones. Un patrón describe una solución general a un problema recurrente en el desarrollo de sistemas (Gamma et al. 1995). Sin embargo, el uso de patrones de manera diferente a la definida propicia la generación de *bad smells*.

En Ingeniería de Software el término *bad smell* es utilizado para denotar un síntoma como consecuencia de un diseño pobre o implementación que negativamente impacta las propiedades de un sistema de software (Fowler et al. 1999), por ejemplo, mantenibilidad, reusabilidad. Un *bad smell* es usualmente utilizado para indicar un problema en el software. Aunque no hay un consenso universal en el área, generalmente se acepta que los *bad smells* pueden ocurrir en diferentes niveles de abstracción: desde el código fuente, por ejemplo, una *long parameter list* (Source Makings, Code Smells 2017) hasta la arquitectura, por ejemplo, *connector envy* (García et al. 2009). A estos últimos, esto es a los patrones de arquitectura, también se les llama estilos arquitectónicos.

Los bad smells son una causa frecuente de acumulación de deuda técnica (Brown et al. 2010). La deuda técnica, es un término que hace referencia al costo en el que se incurre por un enfoque de diseño o desarrollo apresurado y descuidado en el que se ignoran principios y conceptos establecidos, en lugar de usar un enfoque sistemático y disciplinado. Por lo tanto, detectar y corregir *bad smells* se vuelve relevante para el diseño y desarrollo de sistemas de software.

En años recientes, la comunidad de investigación ha estado activamente identificando las características de *bad smells*, por ejemplo (Ganesh, Sharma, and Suryanarayana 2013), así como en el desarrollo de enfoques para detectarlos y corregirlos por ejemplo (Fernandes et al. 2016). Sin embargo, la mayoría de estos enfoques se centran en *bad smells* que ocurren en niveles de abstracción más bajos y pocos de ellos se caracterizan, identifican y corrigen a nivel de arquitectura. Adicionalmente, en estos trabajos de presta poca atención a la realización de estas actividades en el contexto de los estilos arquitectónicos, estructuras de diseño comúnmente utilizadas para el desarrollo de software.

El trabajo presentado en este artículo, es un enfoque inicial para llenar ese vacío y contribuir en: (i) la caracterización de *bad smells* que son relevantes para el estilo de arquitectura Modelo-Vista-Controlador (Model-View-Controller o MVC, por sus siglas en inglés) (Best MVC Practices 2017), que ha sido ampliamente adoptado en frameworks para el desarrollo de sistemas Web, y (ii) la detección automática de estos *bad smells* utilizando técnicas de análisis estático de software en sistemas implementados con el Framework Yii (yiiFramework 2017). Los resultados obtenidos muestran que los *bad smells* definidos existen en la práctica y permiten tener una idea inicial sobre cuáles ocurren con mayor frecuencia. Con respecto al método de detección automática, los resultados muestran que el método de detección propuesta reduce en un amplio margen el tiempo requerido para detectar *bad smells* mediante una revisión de código de forma manual

El contenido de este documento está organizado de la siguiente manera. En la sección 2, se explica el estilo arquitectónico MVC y, en función de sus limitaciones generales, se define una caracterización de *bad smells* relacionados. En la sección 3, se explica el método usado para la detección automática de estos *bad smells*. En la Sección 4 se presentan obtenidos al realizar dos experimentos para determinar la relevancia de las contribuciones esperadas. En la Sección 5 se presenta una discusión del trabajo relacionado. Finalmente, en la Sección 6, se presentan las conclusiones de este trabajo, así como algunas líneas de trabajo futuro.

2. Estilo Arquitectónico MVC y *Bad Smells* Relacionados

La arquitectura de un sistema de software es un modelo de alto nivel de un sistema definido en términos componentes, conectores y propiedades de ambos (Bass, Clements, and Kazman 2012). Si bien es posible especificar la arquitectura de un sistema utilizando este vocabulario genérico, en dominios de aplicación específicos es mejor adoptar un vocabulario más especializado. A este vocabulario especializado da lugar a los estilos arquitectónicos (Bass, Clements, and Kazman 2012), como por ejemplo cliente servidor, tres capas o MVC.

El estilo arquitectónico MVC ha sido ampliamente adoptado para el diseño e implementación de sistemas Web. Actualmente, varios frameworks de desarrollo permiten la construcción de sistemas Web utilizando este estilo, como por ejemplo, Spring (Spring 2018), Django (Django 2018), Rails (Rails 2018), Laravel (Laravel 2018), and Yii. Este estilo separa la lógica de negocio de la lógica de presentación, lo que da como resultado un sistema más fácil de probar y mantener. La Figura 1 muestra una vista estática del estilo arquitectónico MVC y la Tabla 1 describe los elementos que esta vista incluye.

El estilo arquitectónico MVC define la siguientes restricciones generales:

- El Modelo no debe tratar directamente con el procesamiento de las solicitud del usuario final. Por ejemplo, su implementación no debe contener variables `$_GET`, `$_POST`.
- El Modelo no debe tratar directamente con la presentación de los datos para la solicitud del usuario final. Por ejemplo, su implementación no debe contener código de presentación HTML.
- La Vista no debe tratar directamente de realizar un acceso explícito a los datos del sistema. Por ejemplo, su implementación no debe contener código para consultas a Bases de Datos.
- La Vista no debe tratar directamente con la solicitud del usuario final. Por ejemplo, su implementación no debe contener variables `$_GET`, `$_POST`.
- El Controlador, no debe tratar directamente de realizar un acceso explícito a los datos del sistema. Por ejemplo, su implementación no debe contener código para consultas a Bases de Datos.

El Controlador no debe tratar directamente con la presentación de los datos para la solicitud del usuario final. Por ejemplo, su implementación no debe contener código de presentación HTML.

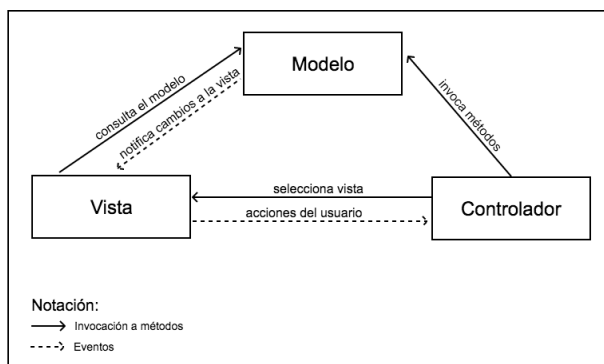


Figura 1 – Representación gráfica del estilo arquitectónico MVC.

Elemento	Descripción
<i>Modelo</i>	Representa los datos del sistema y las reglas de negocio que rigen el acceso a los datos. El Modelo notifica a la Vista si se ha hecho un cambio a los datos.
<i>Vista</i>	Es la representación del Modelo en un formato deseado por el usuario final. La Vista consulta al Modelo para cualquier cambio en los datos.
<i>Controlador</i>	Es un intermediario entre la Vista y el Modelo. El Controlador recibe las acciones del usuario final, las solicitudes de comandos provenientes de la Vista, invoca los métodos requeridos en el Modelo y cambia la presentación de la Vista del Modelo cuando es necesario.

Tabla 1 – Descripción de los elementos del estilo arquitectónico MVC.

2.1. Caracterización de *Bad Smells* MVC

El concepto de *bad smell* arquitectónico fue originalmente usado en (Lippert and Roock 2006) para describir la aparición de un problema de diseño o implementación que ocurre a un alto nivel de abstracción. Esto es, problema de diseño o implementación en la arquitectura. Las causas de los *bad smells* arquitectónicos incluyen, entre otros, aplicar un concepto de diseño arquitectónico en un contexto inapropiado, mezclar responsabilidades entre componentes o aplicar abstracciones de diseño en el nivel a un nivel de granularidad incorrecto (Lippert and Roock 2006).

Aunque casi todos los desarrolladores Web conocen el estilo MVC, su implementación correcta elude a muchos de estos (Best MVC Practices 2017). Frecuentemente, las restricciones generales, que se definieron en la sección anterior, no son respetadas. Esto da como resultado decisiones de diseño deficientes de diseño o implementación, que en este trabajo llamamos *bad smells* MVC. Específicamente, y considerando las restricciones antes mencionadas, la Tabla 2 describe una caracterización de *bad smells* MVC.

Es importante resaltar que esta caracterización de *bad smells* no es propuesta desde cero por nosotros; se deriva de una recopilación de información extraída de varias fuentes (por ejemplo, blogs de desarrolladores, sitios de preguntas y respuestas), que hemos definido en este trabajo de manera más completa y consolidada.

Id	Elemento	Descripción
1.	<i>El Modelo incluye cálculos y/o datos de la Vista</i>	Ocurre cuando el Modelo realiza responsabilidades relacionadas a presentación o recepción de datos en la interfaz del usuario (por ejemplo, código HTML).
2.	El Modelo incluye cálculos y/o datos del Controlador	Ocurre cuando el Modelo accede de forma directa a información sobre las solicitudes del usuario (por ejemplo, acceso directo a las variables \$_GET, \$_POST).
3.	La Vista incluye cálculos y/o datos del Modelo	Ocurre cuando la Vista tiene responsabilidades relacionadas a la consulta y persistencia de datos (por ejemplo, código de acceso a la base de datos).
4.	La Vista incluye cálculos y/o datos del Controlador	Ocurre cuando la Vista accede de forma directa a información sobre las solicitudes del usuario (por ejemplo, acceso directo a las variables \$_GET, \$_POST).
5.	El Controlador incluye cálculos y/o datos de la Vista	Ocurre cuando el Controlador Modelo realiza responsabilidades relacionadas a presentación o recepción de datos en la interfaz del usuario (por ejemplo, código HTML).
6.	El Controlador incluye cálculos y/o datos del Modelo	Ocurre cuando el Controlador tiene responsabilidades relacionadas a la consulta y persistencia de datos (por ejemplo, código de acceso a la base de datos).

Tabla 2 – Caracterización de *bad smells* relevantes en el estilo arquitectónico MVC.

Una vez presentada la caracterización de *bad smells* relevantes en el estilo arquitectónico MVC, que es la primera contribución presentada en este trabajo, en la siguiente sección describimos el método de detección automática de estos *bad smells*.

3. Detección automática de Smells MVC

La herramienta de detección automática de *smells* MVC presentada en este trabajo funciona para sistemas implementados en el framework Yii. Yii es un framework de desarrollo creado en el año 2008. Yii es orientado a objetos, usa en el lenguaje PHP (PHP 2018) y es ampliamente utilizado para el desarrollo de una amplia gama de sistemas Web. Gracias a que no demanda grandes recursos para la ejecución de los sistemas implementados, Yii es adecuado para desarrollar sistemas Web de gran tráfico como portales, foros, sistemas de administración de contenidos, sistemas de comercio electrónico, etc. Como la mayoría de los frameworks PHP, Yii es un framework MVC (modelo-vista-controlador).

En lugar de construir una herramienta desde el principio, se decidió extender las capacidades de la herramienta PHP_CodeSniffer. En las siguientes secciones se proveen los detalles correspondientes.

3.1. PHP_CodeSniffer.

PHP_CodeSniffer es una herramienta de análisis estático de software (PHP_CodeSniffer 2018). El análisis estático de software es un tipo de análisis que se realiza sin ejecutar el código fuente de un sistema –en contraste con el análisis dinámico de software que se realizado sobre un sistema en ejecución. En la mayoría de los casos, el análisis se

realiza en alguna versión del código fuente y en otros casos se realiza en el código objeto. El término se aplica generalmente a los análisis realizados automáticamente usando una herramienta –en contraste con el análisis realizado por un humano, que es denominado comprensión de programas o revisión de código.

PHP_CodeSniffer utiliza archivos denominados *sniffs* para detectar violaciones al conjunto de reglas definidas en un estándar de código (PHP_CodeSniffer 2017). Un estándar de código puede definirse como un conjunto de convenciones que deben utilizarse para escribir archivos de código fuente con el objetivo de lograr estructuras de código que tienen más calidad. Algunos ejemplos de estándares de codificación incluyen: el estándar de código PEAR (PEAR - PHP Extension and Application Repository 2018) o el de GNOME (GNOME Programming Guidelines 2018).

PHP_CodeSniffer funciona sobre la base de separar el contenido de un archivo de código en bloques de construcción, llamados tokens y luego validarlos para verificar una variedad de aspectos especificados en un estándar de codificación. Se pueden usar múltiples estándares de codificación dentro de PHP_CodeSniffer. Después del proceso de análisis estático, PHP_CodeSniffer genera una lista de las violaciones que han sido encontradas, con los correspondientes mensajes de error y números de línea.

Un estándar de código en PHP_CodeSniffer consiste en una colección de archivos de código abierto denominados *sniff*. Cada archivo *sniff* verifica una convención del estándar de código y puede ser codificada en PHP, JavaScript, o CSS. Por lo tanto es posible crear nuevos estándares de codificación, reutilizando, extendiendo y construyendo nuevos archivos *sniff*.

3.2. Estándar de Código para Detectar Smells MVC.

Para detectar *bad smells* Arquitectónicos MVC, construimos una estándar de código llamado Yii MVC. Los archivos *sniff* en este estándar de código son clases PHP, que se utilizan en seis algoritmos de detección para identificar los *bad smells* MVC definidos en la Tabla 2.

La Figura 2 muestra un extracto de código de un archivo *sniff*. Este archivo *sniff*, en comparación con otros, es utilizado para detectar el *bad smell* número 6, que ocurre cuando el Controlador incluye cálculos y/o datos del Modelo. Como se explicó anteriormente, este *bad smell* está relacionado con el problema que aparece cuando un Controlador realiza las responsabilidades de un Modelo. Por lo tanto en términos de código, este *bad smell* resulta incluir un modelo de código para leer escribir o actualizar los datos e incluso almacenarlos, en una base de datos.

Como se muestra en la Figura 2, una clase de tipo Sniff debe implementar la interfaz **PHP_CodeSniffer_Sniff**. Esta interfaz declara dos funciones que son necesarias para el análisis estático del código: la función **register** y la función **process**. La función **register** permite a un *sniff* recuperar los tipos de tokens que van a analizar, en este caso el análisis es sobre cadenas de texto. Una vez que las cadenas están listas para ser analizadas, se llama a la función **process** con una referencia al archivo de código que se está analizando y otra referencia a una pila en donde se encuentran las cadenas de texto: la primer referencia es el parámetro denominado **PHP_CodeSniffer_File** y la segunda referencia es el parámetro **\$stackPtr**, respectivamente.

Se puede encontrar información sobre una cadena de texto con el método **getTokens** en el archivo que contiene el código que se está analizando (línea 13). Este método regresa un arreglo de cadenas, que se indexa de acuerdo a la posición de la cadena en la pila de cadenas. Las cadenas tienen un índice denominado **content** para acceder a su contenido tal y como aparece en el código.

El análisis implementado detecta la declaración **delete** (línea 14) en el código de un archivo que pertenece al Controlador. Esta declaración permite la eliminación de registros en la base de datos. Si la declaración **delete** se encuentra en el código se activa el mensaje de detección del *bad smell* (líneas 17-20). Un *sniff* indica que ha ocurrido un error mediante el método **addError**, el cual es el encargado de generar el mensaje de error como primer argumento, y la posición en la pila donde fue detectado el error como segundo argumento, incluido el código para identificar de forma única el tipo de error dentro del *sniff* y un arreglo de datos utilizado dentro del mensaje de error.

```

...
5: class Yii_Sniffs_BadSmellsArchitectureMVC_SA6_DeleteSniff implements
6:   PHP_CodeSniffer_Sniff {
7:     public function register() {
8:       return array(T_STRING);
9:     }
10:
11:     public function process(PHP_CodeSniffer_File $phpcsFile, $stackPtr){
12:       $name = $phpcsFile->getFilename();
13:       $tokens = $phpcsFile->getTokens();
14:       if ($tokens[$stackPtr]['content'] === 'delete') {
15:         ...
16:         if (strpos($name, "controllers")) {
17:           $error = '6. Controller includes Model's computations and/or
18:             data --> Executes a SQL statement;
19:             Delete found';
20:           $data = array(trim($tokens[$stackPtr]['content']));
21:           $phpcsFile->addError($error, $stackPtr, 'Found', $data);
22:         }
23:       }
24:     }
...

```

Figura 2 – Extracto de un archivo *sniff* para el *bad smell* número 6: El Controlador incluye cálculos y/o datos del Modelo.

Cuando PHP_CodeSniffer se ejecuta en un proyecto Yii usando el estándar de código Yii MVC, existen dos tipos de reportes que se pueden obtener: detallado o resumido. La Figura 3 muestra un ejemplo de un reporte detallado que produce una lista de *bad smells* encontrados, incluyendo el mensaje de error correspondiente y el número de línea. En la Figura 3 se muestra el reporte producido en el análisis estático del archivo de código: **ConfigurationController.php**. La Figura 4 muestra un ejemplo de

reporte resumido, que contiene una lista de todos los *bad smells* detectados en el análisis estático los cuales están ordenados por tipo.

Una vez que se ha presentado la herramienta de detección automática de *bad smells* MVC, que es la segunda contribución presentada en este trabajo, en la siguiente sección se describe la evaluación de las contribuciones presentadas.

4. Evaluación

En esta sección se presentan los resultados obtenidos al realizar dos experimentos para determinar la relevancia de las contribuciones esperadas: (i) la caracterización de *bad smells* relevantes para el estilo MVC y (ii) la detección automática de estos *bad smells* utilizando técnicas de análisis estático.

4.1. Caracterización Propuesta y Detección Automática.

Idealmente para evaluar si la caracterización de los *bad smells* era relevante, se deberían comparar los *bad smells* detectados con los contenidos en un *gold standard* como en los Sistemas de Recuperación de Información (Manning, Raghavan and Schütze 2017). Hasta donde sabemos, no existe un *gold standard* para determinar el número de *bad smells* introducidos en un sistema MVC. Por esta razón realizamos una serie de experimentos básicos en algunos sistemas existentes. Específicamente, se analizaron cinco sistemas creados por terceros e implementados con el Framework Yii para detectar los *bad smells* MVC utilizando PHP_CodeSniffer con el estándar de código Yii MVC. Los sistemas son públicos y de código abierto, y pueden ser descargados desde GitHub (GitHub 2018).

```
FILE ... protected/Controllers/ConfigurationController.php
-----
FOUND 6 ERRORS AFFECTING 6 LINES
-----
84 | ERROR | 6. Controller includes Model's computations and/or data --> Executes a
   | | SQL statement.;
   | Delete found
154| ERROR | 6. Controller includes Model's computations and/or data --> Executes a
   | | SQL statement.;
   | Count found
222| ERROR | 6. Controller includes Model's computations and/or data --> Executes a
   | | SQL statement.;
   | Delete found
244| ERROR | 6. Controller includes Model's computations and/or data --> Executes a
   | | SQL statement.;
   | Update found
264| ERROR | 6. Controller includes Model's computations and/or data --> Executes a
   | | SQL statement.;
   | Insert found
291| ERROR | 6. Controller includes Model's computations and/or data --> Executes a
   | | SQL statement.;
   | Delete found
```

Figura 3 – Un ejemplo de un reporte de errores detallado.

La Tabla 3 muestra los resultados obtenidos al ejecutar los experimentos. Se comprobó que todos los sistemas analizados tenían *bad smells MVC*. El *bad smell MVC* detectado con mayor frecuencia fue el *bad smell* número 6: El Controlador incluye cálculo y/o datos del Modelo. El único *bad smell MVC* sin ocurrencias en los análisis realizados a todos los sistemas fue el número 4: La Vista incluye cálculos y/o datos del Controlador. Aunque los tiempos de detección automática son un tanto heterogéneos, es claro que estos aumentan entre más líneas de código tenga el sistema.

CODE BAD SMELLS SUMMARY		
STANDARD	BAD SMELL ID	COUNT
Yii MCV	6	91
Yii MCV	3	50
Yii MCV	3	12
Yii MCV	2	9
Yii MCV	5	6
Yii MCV	1	5
Yii MCV	2	2
Yii MCV	2	1
A TOTAL OF 176 BAD SMELLS FOUND		
TIME: 69 MIN, 49 SEC		

Figura 4 – Un ejemplo de un reporte de errores resumido.

Sistema	Líneas de código	Bad smells detectados	Bad smells con mayor ocurrencia	Bad smells sin ocurrencias	Tiempo
1. Linkbooks	66, 954	176	6	4	69 min, 49 seg.
2. yii play ground	17, 341	20	6	4	10 min, 54 seg.
3. Blog Bootstrap	6, 393	15	6	4	5 min, 5 seg.
4. yii2-shop	5, 324	14	6	4	4 min, 37 seg.
5. yii-jenkis	836	1	6	1, 2, 3, 4 y 5	1.57 seg.

Tabla 3 – Resultados obtenidos utilizando PHP_CodeSniffer con el estándar de código Yii MVC.

4.2. Caracterización Propuesta y Detección Manual.

Para evaluar el grado de disminución de tiempo y esfuerzo en la detección de *bad smells MVC* usando el PHP_CodeSniffer se realizó otro experimento. El experimento consistió

en la detección manual de los *bad smells* MVC en el sistema en el cual el PHP_CodeSniffer consumió mas tiempo: Linkbooks. La detección la realizaron a un grupo de 15 desarrolladores de nivel experimentado, con conocimientos sólidos en ingeniería de software, arquitectura de software, el estilo arquitectónico MVC y el framework Yii.

Los sistemas fueron proporcionados los participantes y se les pidió que realizaran una revisión de código durante 69 minutos y 49 segundos. Esto es durante el mismo tiempo que tomó la detección automática de *bad smells* MVC en el sistema Linkbooks. Se explicó y proporcionó la caracterización propuesta de *bad smells* MVC a los desarrolladores.

Diez desarrolladores detectaron de 2 a 5 *bad smells*, tres desarrolladores encontraron de 6 a 10 *bad smells* y dos desarrolladores encontraron de 10 a 20 *bad smells*. Así, con la revisión de código realizada por 15 desarrolladores durante 69 minutos y 49 segundos se detectaron un total de 45 *bad smells* MVC. Esto en contraste con los 176 detectados automáticamente.

De este total de 44, el bad smell MVC detectado con mayor frecuencia fue el número 6: El Controlador incluye cálculo y/o datos del Modelo, con 33 ocurrencias. Los *bad smells* MVC que no fueron detectados son el número 4: La Vista incluye cálculos y/o datos del Controlador y el número 5: El Controlador incluye cálculos y/o datos de la Vista.

Habiendo discutido los resultados de estos análisis básicos, en la siguiente sección abordaremos trabajos relacionados.

5. Trabajos Relacionados

Hay dos tipos de categorías principales para trabajos relacionados: (i) catálogos de *bad smells* y (ii) las herramientas para la detección de *bad smells*. En esta sección, relacionamos nuestro trabajo con el encontrado en la literatura sobre estas dos categorías.

5.1. Catálogos de *bad smells*.

En la literatura se distinguen varios tipos de *bad smells*, que al ser organizados ha propiciado la definición de catálogos que caracterizan los *bad smells* en los siguientes tipos: (i) de código, (ii) de diseño y (iii) de arquitectura. A continuación describimos cada uno de estos.

Los *bad smells* de código describen problemas originados por malas prácticas de programación. Existen catálogos que definen bad smells de código; entre los más relevantes encontramos (SourceMaking: Code Smells 2018).

Los *bad smells* de diseño se producen generalmente por violaciones a principios o conceptos del diseño orientado a objetos. Existen catálogos que definen bad smells de diseño como por ejemplo (Ganesh, Sharma, and Suryanarayana 2013) y (Lippert and Roock 2006).

La arquitectura de un sistema es un término que se utiliza para referirse a un modelo de alto nivel que describe su estructuración general en términos de componentes y relaciones (también llamados conectores) entre ellos y las propiedades de ambos (Bass,

Clements, and Kazman 2012). Los *bad smells* de arquitectura se refieren a soluciones no apropiadas al contexto o uso inapropiado de abstracciones de diseño arquitectónico.

A pesar de la gran variedad de trabajos que analizan el impacto de los *bad smells* en arquitecturas de software, se han propuesto muy pocos catálogos de *bad smells* arquitectónicos. Uno de ellos se propone en (García et al. 2009) e incluye cuatro *bad smells* arquitectónicos mencionados a continuación: *connector envy*, *scattered parasitic functionality*, *ambiguous interfaces*, y *extraneous adjacent connector*. En (Vale et al. 2014) los autores realizaron una revisión sistemática para caracterizar los *bad smells* arquitectónicos. Los autores reportaron un conjunto de 14 *bad smells* arquitectónicos, que además de los 4 *bad smells* arquitectónicos reportados en (García et al. 2009) y en versiones específicas de SLP (por ejemplo, *connector envy SPL*), se encontraron otros como: *component concern overload*, *cyclic dependency*, *overused interface*, *redundant interface*, *unwanted dependencies* y *feature concentration*. Además de los pocos catálogos que consideran los *bad smell* a nivel arquitectónico, ninguno de ellos considera los *bad smells* dentro del contexto de un estilo arquitectónico, en contraste con nuestro trabajo.

5.2. Herramientas para la detección de bad smells.

Existen herramientas que permiten la detección automática de algunos de los *bad smells* mencionados anteriormente y otras herramientas que proporcionan una vista de la estructura del sistema para encontrarlos manualmente.

En (Fernandes et al. 2016) los autores presentan los hallazgos de una revisión sistemática en la literatura con un total de 84 herramientas para la detección de *bad smells*. Entre otras observaciones los autores reportan que las herramientas existentes para analizar el código, se concentran en 3 lenguajes de programación, como Java, C, y C++. Encontraron 4 herramientas para el análisis de sistemas de código desarrollados en PHP. Con respecto a las estrategias de detección, los autores descubrieron que la mayoría de la herramientas están basadas en métricas. Finalmente, los autores encontraron 61 *bad smells* diferentes detectables por herramientas, incluyendo Fowler's [1] así como otras encontradas en (Bavota et al. 2015), (Khomh et al. 2011), (Maiga et al. 2012) y (Vidal, Marcos, and Díaz-Pace 2014). A diferencia de la mayoría de estas herramientas descritas, la herramienta presentada en este trabajo puede detectar *bad smells* en sistemas implementados en PHP. Además, los *bad smells* detectados por nuestra herramienta son arquitectónicos y relevantes para el estilo arquitectónico MVC.

6. Conclusiones y Trabajo Futuro

En este trabajo, presentamos nuestro progreso en la caracterización (i) de un conjunto de *bad smells* que son relevantes para el estilo arquitectónico MVC, así como (ii) la detección automática de estos *bad smells* utilizando técnicas de análisis estático en código de sistemas implementados con el Framework Yii.

La realización de este trabajo nos permite afirmar que las herramientas de detección de *bad smells* arquitectónicos son ciertamente útiles para evaluar qué partes del código

deben mejorarse para respetar las restricciones impuestas por el estilo arquitectónico utilizado. Es posible concluir que se pueden desarrollar herramientas de detección de *bad smells* arquitectónicos usando análisis estático y estándares de código definidos para este propósito.

Los resultados obtenidos en nuestros experimentos en la detección automática muestran que la mayoría de los *bad smells* MVC definidos existen en la práctica. Además, aunque los tiempos de detección automática son un tanto heterogéneos, son mejores en comparación con los requeridos en una revisión de código realizada de forma manual. Para el experimento realizado con una revisión de código, involucrando a 15 desarrolladores, se logró detectar solo el 25.5 % de los *bad smells* MVC que detectaron usando la herramienta PHP_CodeSniffer. A nivel individual el desarrollador que rindió mejor detectó 20 *bad smells* MVC, lo que representa solo el 11.3 % de los que detectaron de forma automática.

Aunque este trabajo continúa en progreso, consideramos que los resultados obtenidos hasta el momento son importantes no solo en el área de investigación sino también para el área desarrollo, donde podría ser de gran utilidad el uso de los artefactos generados.

Como posibles líneas de trabajo a futuro, tenemos planeado investigar aspectos relacionados, incluyendo la identificación y caracterización de otros *bad smells* que pueden ocurrir en arquitecturas MVC, la detección de *bad smells* en otros lenguajes de programación, y la mejora continua de la herramienta. En los siguientes párrafos abundamos sobre estas líneas.

Consideramos que es posible agregar a nuestro conjunto inicial de *bad smells* arquitectónicos MVC otros *bad smells* que no se están documentados en la literatura. Nosotros visualizamos la clasificación de *bad smells* MVC basada considerando dimensiones, como por ejemplo el comportamiento y la estructura (Ganesh, Sharma, and Suryanarayana 2013). Todos los *bad smells* detectados por usando el enfoque descrito en este trabajo consideran solo la una dimensión de comportamiento.

Como se describe en (Fernandes et al. 2016) la mayoría de las herramientas que permiten detectar *bad smells*, están restringidas para solo detectar *bad smells* en un lenguaje de programación específico, lo cual se convierte en una limitación significativa para cada una de las herramientas que permiten detectar *bad smells*. Por otro lado, existen técnicas de *machine learning* que son métodos computacionales que usan la “experiencia” para obtener una mejor precisión en la predicción. Consideramos que la aplicación de técnicas de *machine learning* para la detección de *bad smells* arquitectónicos puede favorecer el desempeño y la precisión, así como soportar más de un lenguaje de programación, siempre y cuando se tengan conjuntos de ejemplos para el entrenamiento.

Finalmente, con respecto a la mejora del soporte de herramientas, tenemos la intención de explorar la identificación de posibles refactorizaciones del código para corregir los *bad smells* identificados. Además, también sería útil que los *bad smells* detectados se pudieran mostrar de manera visual. Consideramos que esto contribuiría a que programadores novatos pudieran comprender de forma más fácil por qué no se respetan las restricciones del estilo.

Referencias

- Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice*. Boston, MA, USA: Addison-Wesley Professional.
- Bavota, G. (2015). An Experimental Investigation on the Innate Relationship between Quality and Refactoring. *Journal of Systems and Software*, 107, 1–14.
- SourceMaking. (2018). *SourceMaking: Code Smells 2018*. Retrieved from: <https://sourcemaking.com/refactoring/badsmellsincode>
- Yiiframework. (2017). Best MVC Practices. Retrieved from: <http://www.yiiframework.com/doc/guide/1.1/en/basics.best-practices>.
- Brown, N. (2010). Managing Technical Debt in Software Reliant Systems. In *Proceedings of Workshop on Future of Software Engineering Research, at the 18th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ACM, (pp. 47-52). Santa Fe, NM, USA : ACM.
- Django Software Foundation. (2018). *Django*. Retrieved from: <https://www.djangoproject.com>.
- Espejo Chavarría, A., Bayona Oré, S., & Pastor, C. (2016). Aseguramiento de la calidad en el proceso de desarrollo de software utilizando CMMI, TSP y PSP. *RISTI - Revista Ibérica de Sistemas y Tecnologías de Información*, 20(12), 62–77.
- Fernandes, E. (2016). A Review-Based Comparative Study of Bad Smell Detection Tools. In *Proceedings of 20th International Conference on Evaluation and Assessment in Software Engineering (pp. 109-20)*. New York: ACM.
- Fowler, M. (1999). *Refactoring: Improving the Design of Existing Code*. Boston, MA, USA: Addison-Wesley.
- Gamma, E. (1995). *Design patterns: Elements of reusable object-oriented software*. Reading, MA, USA: Addison-Wesley.
- Ganesh, S. G., Sharma, T., & Suryanarayana, G. (2013). Towards a Principle-Based Classification of Structural Design Smells. *Journal of Object Technology* 12(2): 1–29.
- García, J., Popescu, D., Edwards, G., & Medvidovic, N. (2009). Toward a Catalogue of Architectural Bad Smells. In *Proceedings of 5th International Conference on the Quality of Software Architectures (pp. 146–62)*. Berlin, Heidelberg: Springer-Verlag.
- GitHub. (2018). *GitHub*. Retrieved from: <https://github.com/>
- GNOME developer. (2018). GNOME Programming Guidelines. Retrieved from: <https://developer.gnome.org/programming-guidelines/stable/>
- Khomh, F., Vaucher, S., Guéhéneuc, Y., & Sahraoui, H. (2011). BDTEX: A GQM-Based Bayesian Approach for the Detection of Antipatterns. *Journal of Systems and Software* 84: 559–72.
- Laravel. (2018). *Laravel*. Retrieved from: <https://laravel.com>.

- Lippert, M., & Roock, S. (2006). *Refactoring in Large Software Projects: Performing Complex Restructurings Successfully*. Wiley.
- Maiga, A. (2012). Support Vector Machines for Anti-Pattern Detection. In *Proceedings of 27th International Conference on Automated Software Engineering*, (pp. 278-81). New York: IEEE Press.
- Manning, C. D., Raghavan, P., & Schütze, H. (2017). *Introduction to information retrieval*. Delhi: Cambridge University Press.
- Morales Huanca, L., & Bayona Oré, S. (2017). Factores que afectan la precisión de la estimación del esfuerzo en proyectos de software usando puntos de caso de uso. *RISTI - Revista Ibérica de Sistemas y Tecnologías de Información*, 21(03), 18–32.
- PEAR. (2018). *PEAR - PHP Extension and Application Repository*. Retrieved from: <http://pear.php.net/>
- PEAR. (2018). *PHP_CodeSniffer*. Retrieved from: https://pear.php.net/package/PHP_CodeSniffer.
- PHP. (2018). *PHP: Hypertext Processor*. Retrieved from: <https://secure.php.net/>
- Rails. (2018). *Rails*. Retrieved from: <http://rubyonrails.org>
- Source Makings. (2017). *Source Makings, Code Smells*. Retrieved from: <https://sourcemaking.com/refactoring/smells>.
- Spring. (2018). *Spring*. Retrieved from: <https://spring.io>.
- Vale, G., Figueiredo, E., Abílio, R., & Costa, H. (2014). Bad Smells in Software Product Lines: A Systematic Review. In *Proceedings of Eighth Brazilian Symposium on Software Components, Architectures and Reuse* (pp. 84–94). Washington, DC, USA: IEEE Computer Society.
- Vidal, S., Marcos, C., & Díaz-Pace, J. (2014). An Approach to Prioritize Code Smells for Refactoring. *Automated Software Engineering* 23: 501–32.
- YiiFramework. (2017). *YiiFramework*. Retrieved from: <http://www.yiiframework.com/>.

Proceso de evaluación de usabilidad en aplicaciones colaborativas con interfaces de usuario multimodal

Gabriela Sánchez Morales¹, Carmen Mezura-Godoy¹, Itzel Alessandra Reyes Flores¹,
Edgard Benítez-Guerrero¹

gabriesanchez@uv.mx, cmezura@uv.mx, itreyes@uv.mx, edbenitez@uv.mx

¹ Facultad de Estadística e Informática, Av. Xalapa, S/N, 91020, Xalapa, Veracruz. México.

DOI: 10.17013/risti.26.68–82

Resumen: Las aplicaciones colaborativas (AC), pueden ser evaluadas desde varias dimensiones: contexto grupal, usabilidad, colaboración e impacto cultural. La usabilidad en AC se mide en términos de eficiencia, eficacia y satisfacción, en la que un grupo de usuarios pueden alcanzar objetivos y metas en común. Para la evaluación de usabilidad se han propuesto métodos y heurísticas de evaluación que pueden aplicarse en las diferentes fases del desarrollo de una AC. En este trabajo, presentamos una propuesta de evaluación de usabilidad para aplicaciones colaborativas con interfaces de usuario multimodal, aplicada a los cuatro elementos de una AC: comunicación, colaboración, coordinación y regulación. La propuesta contempla la evaluación en tres etapas: planeación, ejecución y análisis y reporte. La propuesta de evaluación se aplicó a una AC con interfaz multimodal que permite la elaboración de diagramas de flujo.

Palabras-clave: usabilidad; evaluación; multimodal; aplicación colaborativa.

Usability evaluation of collaborative applications with multimodal user interface

Abstract: Collaborative applications (CA) can be evaluated from several dimensions: group context, usability, collaboration and cultural impact. Usability in CA is measured in terms of efficiency, effectiveness and satisfaction, in which a group of users can achieve common objectives and goals. In order to evaluate groupware usability, several methods and evaluation heuristics have been proposed, they can be applied in the different phases of the development of an AC. In this paper, we present a proposal for usability evaluation for collaborative applications with multimodal user interfaces applied to the four elements of a CA: communication, collaboration, coordination and regulation. The proposal contemplates the evaluation in three stages: planning, execution and analysis and report. The evaluation proposal was applied to an AC with multimodal interface that allows the elaboration of flow diagrams

Keywords: Usability, evaluation, multimodal, collaborative applications

1. Introducción

El CSCW¹ Lab ha propuesto cuatro dimensiones a considerar al evaluar sistemas colaborativos: contexto grupal, usabilidad, colaboración e impacto cultural (Araujo, 2002). El trabajo realizado por Pinelle y Gutwin (2002) propone que la usabilidad de groupware² se define como “la medida en que un sistema de software colaborativo permite que el trabajo en equipo ocurra de manera eficaz, eficiente y satisfactoria para un grupo de usuarios y una actividad en particular”.

Para evaluar la efectividad, eficiencia y satisfacción del software se han desarrollado métodos de evaluación que comprenden un conjunto de actividades bien definidas (Fernandez, 2011). Estos métodos evalúan características específicas dependiendo de las aplicaciones, tales como: Web (Fernandez, 2011; Saltiveri, 2013), móviles (Molano, 2015) y colaborativas (Herskovic, 2007; Quiñonez, 2016). Por ejemplo, en la evaluación de las aplicaciones Web se considera: mantener actualizada la información, direccionar cada liga al elemento correspondiente, entre otros aspectos. En el caso de las aplicaciones móviles se considera: el tamaño de la pantalla, navegación por voz, etc. Para las Aplicaciones Colaborativas (AC o groupware) se han considerados variables de contexto (aspectos personales, situacionales, estructura del grupo, actividades), proceso del grupo (interacción entre los miembros del grupo, comunicación) y las salidas del sistema (relacionadas con las tareas del grupo, como con el grupo mismo) (Pinsonneault, 1989).

Las AC constituyen un software que apoya a los usuarios a alcanzar una meta u objetivo en común. Para ello, las AC proveen los medios para que los usuarios se comuniquen, coordinen sus actividades, colaboren y establezcan reglas para el trabajo en grupo, es decir regulen su actividad (Mezura-Godoy, 2008). Para hacer que estas aplicaciones sean más usables, se han dotado de Interfaces de Usuario Multimodales (IUM), es decir que cuenten con más de una modalidad en la interacción del usuario con el sistema.

La modalidad corresponde a entradas o salidas del sistema por medio de: tacto, voz, gestos, entre otros. Una IUM utiliza diferentes dispositivos de entrada y salida de la computadora, haciendo uso de diferentes canales sensoriales (Moller, 2009). La modalidad puede ser entonces representada por la tupla < *lenguaje de interacción*, *dispositivo físico* >. En donde el lenguaje de interacción es un conjunto de expresiones bien formadas que transmiten un significado, mientras que el dispositivo físico se refiere a los periféricos de la computadora que dan la entrada o la salida de información (Vanderdonckt, 2008). Las IUM han demostrado tener muchas ventajas comparadas con interfaces monomodales, en cuanto a que facilitan el acceso a la información y a la funcionalidad del sistema a través de una gama de diferentes canales de entrada y salida (Dix, 2009), proveen robustez a la interfaz, ayudan al usuario a corregir errores o recuperarse más fácilmente, y añaden métodos de comunicación alternativos a diferentes situaciones y ambientes (Jaimes, 2007). Ejemplos de AC con IUM pueden ser: 1) DISCIPLE (Dorohonceanu, 2000), esta aplicación apoya a los usuarios en la toma de decisiones en un contexto militar, las modalidades de entrada corresponden a reconocimiento de voz, manipulación directa mediante un guante táctil y rastreador

¹ Computer Support Collaborative Work

² Término en inglés para denominar a las aplicaciones o software colaborativo.

de mirada, mientras que las de salida corresponde a voz. 2) TowerWorld, diseñado para compartir archivos entre los usuarios con una interfaz con modalidades de reconocimiento de voz y manipulación directa con mouse y teclado, y 3) AudioSpace Invaders, que es un juego colaborativo para personas ciegas que posee una interfaz de audio que permite a los usuarios especificar contextos que les interesa.

En el proceso de desarrollo de software se hace necesario incluir tareas tendientes a la evaluación no sólo funcional, no funcional o de interfaz, sino también en el caso del software colaborativo es necesario evaluar aspectos sociales tales como: conformación del grupo, la interacción entre sus miembros, la comunicación, la colaboración, las reglas del trabajo en grupo, la presencia del grupo, etc. Estos trabajos nos han llevado a incluir en el componente de interfaces diversas modalidades. En trabajos previos hemos propuesto modelos y arquitecturas que contemplen la definición y desarrollo de AC (Mezura-Godoy, 2008; Montané-Jimenez, 2012; Decouchant, 2013) y de AC con interfaces multimodales (Reyes, 2016). Particularmente, en este trabajo nos interesamos en la evaluación de las interfaces de usuario multimodales de las AC. Existen trabajos que proponen la evaluación de la usabilidad de las AC en la fase de diseño de las interacciones (Herskovic, 2007), otros sugieren evaluar no sólo la usabilidad sino otros aspectos como es el impacto cultural (De Araujo, 2004). Aun cuando estos trabajos consideran evaluar varias características de las AC, no se contemplan a las AC con IUM. Por ello, en este trabajo hacemos una propuesta de evaluación de usabilidad de las AC con IUM.

Este artículo se encuentra organizado de la siguiente manera: en la sección 2 se presenta el estado de arte acerca de los métodos de evaluación de las AC. En la sección 3, se describe la propuesta de este trabajo para evaluar la usabilidad de las AC con IUM mediante pruebas de usabilidad con usuarios y basada en heurísticas; en la sección 4, se presenta la aplicación de las pruebas de evaluación que se propone, en una AC que apoya el diseño colaborativo de diagramas de flujo. Finalmente, la sección 5 se presenta las conclusiones y trabajo futuro.

2. Métodos de evaluación de AC

Las aplicaciones colaborativas (AC) son “sistemas basados en computadoras que soportan grupos de personas que participan en una tarea común y que proveen una interfaz para un ambiente compartido” (Ellis, 1991). Las AC apoyan a los grupos de personas en: la comunicación, la coordinación, la colaboración (3C's) y la regulación (Mezura-Godoy, 2008). La arquitectura de este tipo de sistemas comprende cinco componentes: i) funcional (funcionalidades propias de la aplicación), ii) social (reglas de colaboración, interacciones sociales, roles, objetos a manipular, etc.), iii) individual (características, habilidades, preferencias, etc.), iv) no funcional (seguridad, conectividad, manejo de versiones, etc.) y v) de interfaz de usuario y de grupo. Particularmente, la interfaz en este tipo de sistemas es muy importante, ya que tiene que dar cabida a aspectos particulares del usuario (gustos, preferencias, habilidades, etc.), así como aspectos del trabajo en grupo (¿quién está haciendo qué?, ¿cómo lo está haciendo?, ¿qué objetos manipula?, ¿cuándo los usuarios participaron?), entre otros.

Las AC pueden ser evaluadas bajo un enfoque de usabilidad teniendo en cuenta la efectividad, eficiencia y satisfacción. La efectividad puede ser medida en términos de

la precisión y completitud con la que el usuario alcanza sus objetivos, la eficiencia se refiere al tiempo y esfuerzo para alcanzar los objetivos, y la satisfacción es medible en términos de la ausencia de incomodidad y una actitud positiva hacia el uso de la aplicación (Gutwin, 2000).

Se ha propuesto incorporar varios métodos para evaluar las AC a lo largo de las diferentes etapas del desarrollo (análisis, diseño e implementación, previo a la liberación del producto) como se observa en la Tabla 1. En el análisis se requiere medir la viabilidad del sistema mediante los métodos (Herskovic, 2007): Heurísticas de Evaluación de Colaboración (HEC), y Valor Percibido (Herskovic, 2007). En el diseño, los métodos HEC, Walkthrough Colaborativo (WC), Análisis de Usabilidad de la Colaboración (UAC), Pruebas de Usuario Colaborando (OPUC) y Etnografía rápida y sucia (ERS) ayudan a determinar la correcta implementación de los mecanismos de las AC (Herskovic, 2007). Se evalúan las AC liberadas o en etapas previas a la liberación, para conocer la relación de los usuarios con sus necesidades, así como la efectividad y eficiencia; algunos de los métodos sugeridos corresponden a HEC, WC, AUC y OPUC. Para Herskovic (2007), no es recomendable una combinación específica de las tres etapas indicadas, en su caso dejan al evaluador que determine un método dado el tiempo y el costo de cada una de ellas (Ver Tabla 1).

Método	Etapas	Objetivo
H-E y Valor Percibido	Análisis	Medir impacto organizacional y las expectativas de los diseñadores y usuarios
HEC, WC, UAC, OPUC y ERS	Diseño y desarrollo	Verificar correcta implementación de mecanismos de las AC y encontrar defectos
HEC, WC, AUC y OPUC	Liberado o previo a la liberación	Medir la eficiencia y la efectividad y la relación entre funcionalidad, usuarios y las necesidades de la organización

Tabla 1 – Métodos de evaluación por etapas de desarrollo software

Para evaluar las AC, trabajos previos han modificado los métodos usados para evaluar aplicaciones mono-usuarios (heurísticas de evaluación, walkthrough cognitivo, observación de las pruebas de usuario, análisis de tarea, etc.). Algunos de los métodos que permiten evaluar las AC son los siguientes (Herskovic, 2007):

- HEC evalúa la comunicación de los usuarios distribuidos y la colaboración con artefactos a través de un espacio visual compartido. HEC se basa en 8 heurísticas relacionadas con los medios para la comunicación (verbal intencional, gestual intencional, adecuada, acciones corporales de los usuarios, artefactos compartidos, entre otras), la protección al trabajo y la administración de las transiciones de la colaboración (estrecha y ligeramente acoplada); finalmente parte de estas 8 heurísticas brindan apoyo a las personas para coordinar sus acciones, buscar colaboradores y contactarlos (Baker, 2002).
- En WC se observa a usuarios para identificar episodios de colaboración y construir escenarios como una descripción de la actividad o conjunto de tareas. Se incluye al usuario, resultados planeados y circunstancias alrededor. Un

evaluador puede tomar uno o todos los roles y seguir una tarea recordando cada problema encontrado.

- Pruebas de Usuario Colaborando (OPUC): Algunos métodos existentes parten de observar la colaboración entre los usuarios, ya sea para analizar algún problema en especial (Herskovic, 2007).
- ERS se basa en la etnografía para buscar problemas de aceptabilidad y usabilidad en sistemas actuales que puedan ser corregidos en sistemas futuros o dar simplemente una idea general a los diseñadores de cómo se lleva a cabo el trabajo colaborativo (Herskovic, 2007);
- Análisis de Usabilidad de la Colaboración (AUC): método que se basa en un análisis jerárquico de tareas individuales y grupales, permite ejecutar una tarea usando rutas alternativas y tareas opcionales; las tareas grupales deben relacionarse con mecanismos colaborativos que puedan verse reflejados en la interfaz de usuario (Pinelle, 2003).
- Humano-Ejecución (H-E): divide la interfaz en espacios compartidos y definen varios escenarios críticos de las actividades colaborativas, con la finalidad de comparar la ejecución de estos (Herskovic, 2007).

Es importante señalar que para llevar a cabo una evaluación se debe seguir un plan de trabajo bien definido. Para ello, De Araujo (2004), Yu (2013), y Rogers (2011) coinciden en establecer los siguientes puntos para realizar un plan de evaluación, que consiste en: establecer un objetivo, seleccionar el método de evaluación, determinar el escenario de interacción, seleccionar y reclutar el grupo de participantes, programar las tareas y fechas, recolectar y analizar resultados.

3. Evaluación de AC-IUM mediante pruebas de usabilidad y heurísticas

La propuesta de evaluación de AC-IUM se basa en el proceso general de la evaluación de usabilidad en tres etapas (Sánchez, 2018): 1) planeación para definir los objetivos de la prueba, el método de evaluación, la selección de los usuarios, el laboratorio, entre otros aspectos; 2) ejecución es llevar a cabo todo lo planeado y 3) análisis y reporte de los datos obtenidos. En este trabajo estas tres etapas son específicas para evaluar Aplicaciones Colaborativas con Interfaces de Usuario Multimodal (AC-IUM), mediante pruebas de usabilidad y basadas en heurísticas. En esta propuesta se considera que las pruebas de usabilidad son las que se llevan a cabo con usuarios, mientras que las basadas en heurísticas son ejecutadas por evaluadores expertos en usabilidad.

Algunas actividades en el proceso de evaluación pueden usarse tanto en las pruebas de usabilidad como en las basadas en heurísticas, pero otras no; por tal razón, en esta propuesta, se especificará cuales actividades pueden ser aplicadas para ambos casos o solo para uno. Además de lo anterior, en este trabajo se ofrecerán sugerencias con la finalidad de ayudar a personas con poca experiencia en la realización de dichas pruebas. A continuación se explican con detalle cada una de las tres etapas mencionadas para la evaluación de usabilidad de las AC-IUM.

3.1. Etapa de planeación

En la planeación de la evaluación se consideran 4 aspectos: 1) objetivos y preguntas de apoyo, 2) tareas, 3) recursos y 4) plan de ejecución. Se recomienda realizar formatos desde las tareas, con la finalidad de tener una mejor organización.

1) *Objetivos y preguntas de apoyo*

Definir los objetivos de la evaluación, los cuales frecuentemente responden a qué tan usable es la aplicación considerando cada una de las modalidades empleadas; mientras que las preguntas de apoyo ayudan a resolver dichos objetivos; por ejemplo, ¿podrán los usuarios enterarse del ingreso de un usuario mediante la modalidad de audio? o ¿podrán los usuarios colaborar para el lograr el objetivo deseado mediante la modalidad textual?

2) *Tareas*

Consiste en determinar las tareas individuales y grupales considerando la (las) modalidad(es) de interacción; dichas tareas frecuentemente se relacionan con las preguntas de apoyo o con los mecanismos de colaboración (comunicación, coordinación, colaboración y regulación). Algunas tareas individuales pueden ser configurar el entorno personal o realizar una búsqueda mediante navegación por voz o textual. Mientras que las tareas grupales pueden relacionarse con los mecanismos de colaboración (comunicación, colaboración, coordinación y regulación según la clasificación de (Sánchez, 2017)) y sus diferentes modalidades.

En las *pruebas de usabilidad*, una actividad relacionada con las tareas es establecer el tiempo de cada una; con la finalidad de poner tiempo límite para realizar estas y su vez reportar cuales fueron realizadas en el tiempo estimado. Sugerencias: 1) considerar tiempo para que los usuarios exploren la aplicación; 2) realizar un formato para los usuarios, en donde se incluyan las tareas a realizar, un escenario por cada tarea y preguntas relacionadas con la satisfacción, y 3) otro formato para los observadores con las tareas seleccionadas y los eventos que se sugieren observar.

3) *Recursos*

En los recursos se consideran los **humanos (RH)**, los **físicos** y los de **evaluación**.

Los **RH** incluyen el equipo de pruebas y la selección de usuarios o evaluadores expertos. El equipo de las pruebas se refiere a las personas encargadas de realizar las pruebas, cuyos roles corresponden a: Moderador, Observador y Técnico. El Moderador conducirá toda la prueba que abarca desde dar la bienvenida a los participantes hasta finalizar las pruebas. El Observador es empleado en las *pruebas de usabilidad* para registrar: tiempo empleado en cada tarea, dificultades encontradas y el estado de ánimo de los usuarios al interactuar con el sistema en cuestión, etc. El Técnico verificará las instalaciones físicas requeridas y de software, la conexión a internet, etc.; en el caso de las *pruebas de usabilidad*, una actividad más es preparar lo requerido al momento de interactuar con diferentes modalidades (micrófono y audífonos para interactuar por

voz, etc.). *Sugerencia en las pruebas de usabilidad:* 1) establecer el perfil deseado en la selección de usuarios, por ejemplo el nivel de experiencia, rango de edades, género, preferencias de las modalidades de interacción, etc. 2) Formar subgrupos de usuarios heterogéneos respecto a sus preferencias de modalidades.

Los **recursos físicos** consideran el hardware, el software y el lugar para realizar las pruebas. El hardware corresponde a los dispositivos en donde se ejecutará la aplicación a evaluar; en el caso de las *pruebas de usabilidad* se incluyen lo necesario para grabar a los usuarios, así como equipo auxiliar que permitan la interacción con diferentes modalidades. Mientras que el software se refiere a la aplicación a evaluar; para las *pruebas de usabilidad* se deben considerar las herramientas de grabación (interacciones y el rostro del usuario). El recurso físico también incluye el lugar en donde se realizaran las pruebas, como son laboratorios de usabilidad o habitaciones acondicionadas. El recurso físico en el caso de *pruebas de usabilidad* es aconsejable designar a cada usuario en cuartos diferentes, en el caso que la modalidad de interacción sea por voz.

El **recurso de evaluación** se refiere a seleccionar el método de evaluación (Sánchez, 2017) para determinar la usabilidad en términos de efectividad, eficiencia y satisfacción, considerando las heurísticas ligadas a los aspectos de las AC (comunicación, coordinación, colaboración y regulación), mismos que se observan en la Tabla 2. Dichos métodos deben considerar cada una de las modalidades, por ejemplo, entrada de voz, o salida de audio.

En el aspecto de comunicación se considera 3 heurísticas: la heurísticas 1 que provee los medios para la comunicación verbal intencional y apropiada, la heurística 2 proporciona los medios para la comunicación gestual intencional y adecuada, mientras que la heurística 3 provee comunicación originada por las acciones corporales de los usuarios. En general, las heurísticas de este aspecto pueden considerar modalidades como entrada de voz, salida de audio, entrada de gráfico, etc. Por tal razón, es importante evaluar la usabilidad de las heurísticas considerando sus diferentes modalidades.

Mecanismos de las AC	Heurística	Método		
		Efectividad	Eficiencia	Satisfacción
Comunicación	1,2,3	OPUC, WC, UAC	HEC, OPUC, WC	Cuestionarios de satisfacción considerando las modalidades
Coordinación	4,5			
Colaboración	6,8			
Regulación	7, 9			

Tabla 2 –Métodos de evaluación a partir de los mecanismos de las AC y las heurísticas

El aspecto de coordinación considera las heurísticas 4 y 5 (Ver Tabla 2), la heurística 4 proporciona comunicación consecuente de artefactos compartidos, a su vez la heurística 5 provee protección al trabajo. Es importante mencionar que dadas las diferentes modalidades que se pueden emplear en estas heurísticas, es necesario medir la usabilidad que ofrecen. Por ejemplo, no es lo mismo comunicar por audio los cambios en un artefacto compartido que comunicarlo de manera textual.

Mientras que la colaboración está compuesta por las heurística 6 que se encarga de administrar las transiciones entre la colaboración estrechamente y ligeramente acoplada; adicionalmente, la heurística 8 facilita la búsqueda de colaboradores y la manera de establecer contacto.

En el aspecto de regulación de una AC se consideran la heurística 7 que se relaciona con la asignación de tareas a los participantes de la actividad, y cómo éstas se tienen que llevar a cabo para alcanzar la tarea final. La heurística 9 que provee a los usuarios los medios para la definición de la regulación social [5].

Tanto en el aspecto de colaboración como de regulación se debe considerar las modalidades que permiten cada uno de ellos.

A partir de los mecanismos de las AC y las heurísticas de evaluación propuestas se propone evaluar la usabilidad considerando la: efectividad, eficiencia y satisfacción.

- Efectividad: se refiere a la completitud de las tareas individuales y grupales, considerando cada una de las modalidades establecidas, por ejemplo: colaborar en la redacción de un documento mediante la modalidad de entrada de voz. La completitud puede definirse mediante WC, AUC u OPUC; ya que la primera está relacionada con el recorrido de las tareas, la segunda con el desglose de las tareas; mientras que la última se basa en observar la colaboración.
- Eficiencia: es reportar las tareas individuales o grupales terminadas dentro o fuera del tiempo esperado, considerando la complejidad que puede estar asociada a una modalidad, por ejemplo: la modalidad de audio asociada al *awarness*. La eficiencia puede calcularse mediante HEC, OPUC o WC; las dos primeras pueden servir para determinar el tiempo; también el segundo y el tercer modelo pueden ser empleados para saber el esfuerzo requerido por los usuarios.
- Satisfacción: se obtiene al hacer cuestionarios o entrevistas que involucren las modalidades utilizadas en cada una de las heurísticas consideradas en cada aspecto de las AC.

4) Plan de ejecución

Se considera 2 actividades: 1) Redactar un guión de ejecución con los siguientes puntos: agradecimiento, explicación de la aplicación a evaluar, objetivo de la evaluación, actividades a realizar; indicar si serán observados al momento de interactuar, y 2) Realizar el programa de actividades de acuerdo al guión. En esta parte se sugiere: 1. Hacer un formato para el guión y establecer el tiempo por cada actividad; si las realizaciones de las tareas duran más de 25 minutos se debe tener un descanso. 2. Hacer un formato tipo *checklist* que considere actividades a realizar antes, durante y después de la prueba (Sánchez, 2018).

3.2. Etapa de ejecución

El día de la prueba se debe tener a la mano todos los formatos en el orden que serán usados, llegar con anticipación al lugar de la evaluación para preparar lo necesario. Durante la prueba se debe seguir el guion o el *checklist* correspondiente.

En esta etapa se sugiere que para *las pruebas de usabilidad*: 1) El moderador debe emplear tiempo antes de la prueba para platicar con el usuario con la finalidad de relajarlos y hacerlos sentir en confianza, 2) Evitar horarios usuales de comida, ya que los usuarios no estarán concentrados en la actividad, 3) El equipo de evaluación debe ser amable con los usuarios, no sorprenderse o emitir sonidos cuando cometan errores los usuarios, evitar distracciones (teléfonos, visitas personales, platicar, etc.), 4) Al finalizar las pruebas invierte tiempo con los usuarios, puesto que ellos en ocasiones proveen más información, 5) Reúnete con tu equipo de evaluación para hablar de mejoras en las pruebas, y 6) Al finalizar cada sesión o todas las pruebas platica con los observadores para recopilar sus observaciones, detectar problemas graves sobre la aplicación.

3.3. Etapa de análisis y reporte

Al finalizar las pruebas se debe hacer un reporte que incluya una: explicación breve acerca de la aplicación a evaluar, una descripción por un lado de la planeación siguiendo el punto 3.1, y por otro del proceso que se llevó cabo en la ejecución de la prueba. Finalmente, el reporte incluye el análisis del resultado.

El análisis de los resultados obtenidos incluye el total de los usuarios o evaluadores expertos, cuántos son novatos, intermedios y expertos. Los resultados relacionados con la efectividad, eficiencia y satisfacción se representa usando tablas, gráficos, los valores mínimos y máximos, porcentaje o el número de éxitos del total de usuarios. En donde la:

- La afectividad debe ser reportada al describir la completitud de las tareas individuales y grupales, considerando cada una de las modalidades establecidas.
- La eficiencia: es reportar el tiempo consumido en las tareas individuales o grupales terminadas. En el caso de las pruebas de usuario se reportan las tareas terminada en el tiempo esperado o fuera de tiempo.
- La satisfacción es reportado al mencionar la preferencia de la modalidad de cada tarea reportada por los usuarios finales, dichos resultados son obtenidos de los cuestionarios o entrevistas.

4. Evaluando Co-flows usando la metodología propuesta

En esta sección se describe la aplicación de la propuesta de evaluación en una AC denominada “Co-flows”. Esta AC permite a un grupo de usuarios, ubicados en diferentes espacios físicos, colaborar al mismo tiempo en la creación de diagramas de flujo. La aplicación en su componente funcional ofrece medios para la colaboración, comunicación, coordinación y regulación.

Para la colaboración se ofrecen funcionalidades que permitan la creación, modificación y guardado de diagramas de flujo; inserción, modificación y eliminación de objetos del diagrama tales como: círculos, rombos, rectángulos, entre otros y acceso a un área de dibujo común.

En la comunicación se provee de un chat que permite enviar y recibir mensajes entre los usuarios; dichos mensajes pueden enviarse y recibirse mediante la modalidad gráfica, textual y audio. Los mensajes pueden ser enviados de una modalidad y recibirse en otra

dadas las preferencias y características del ambiente de cada usuario. La comunicación, también ofrece un servicio de notificación para informar de las acciones realizadas en el espacio común y *awareness*; este servicio habilita la modalidad de audio y texto dada las preferencias del usuario y del ambiente de él.

Para la coordinación y regulación, se ha considerado la asignación de roles y actividades a los usuarios, con la finalidad de permitir a los usuarios una mejor coordinación.

Co-flows se ha evaluado en diferentes etapas del desarrollo del software (Sánchez, 2017). En este artículo se muestra el proceso de evaluación de usabilidad de Co-flows en una versión completa.

4.1. Planeación

El *objetivo* de esta prueba es medir qué tan usable es Co-flows al ofrecer diversas modalidades para la interacción.

En esta evaluación se consideran las *tareas grupales* como se observa en la Tabla 3, en donde se describe cada tarea, el tiempo considerado y las modalidades que serán empleadas durante la ejecución de la tarea solicitada.

Respecto a los participantes (*Recursos humanos*) se consideran 5 equipos formados entre 3 y 4 estudiantes de semestres similares, con experiencia en el uso de aplicaciones colaborativas, además de identificarse por su canal sensorial preferente, obtenido de un pre-test.

No.	Tarea	Descripción de la tarea	Modalidades a evaluar	Tiempo estimado
1	Explorar la aplicación	Los usuarios pueden explorar la aplicación colaborativa	ninguna	5 min.
2	Elegir un coordinador	Los usuarios deben coordinarse usando el chat		5 min
3	Realizar un diagrama de flujo	El equipo de manera grupal realiza un diagrama de flujo, que representa el envío de un correo electrónico desde un navegador Web.	Audio, texto	16 min

Tabla 3 – Tareas grupales solicitadas a los usuarios

El *equipo de pruebas* será formado por 2 personas para desarrollar el rol de técnico antes de la prueba; durante la prueba uno será observador y otro moderador. El moderador debe tener experiencia en la realización de pruebas.

El *hardware* requerido: cuatro computadoras personales (Laptop) con webcam para las grabaciones y audífonos. El *software* requerido es un programa para grabar a los usuarios y Co-flows instalado en una computadora. Los *lugares* para realizar las pruebas serán laboratorios de pruebas, cuartos de observación y laboratorio de cómputo de dos universidades públicas, una en Puebla y otra en Veracruz. Se elaboró un formato de listado de actividades relacionadas con la preparación de hardware y software para el

día de la evaluación y un formato de consentimiento por parte de los usuarios para ser video grabados.

En el *recurso de evaluación* se aplicó el método OPUC que midió la efectividad en todos los mecanismos de las AC mientras que la satisfacción se midió usando un cuestionario basado en las 6 heurísticas implementadas. Dicho cuestionario permitirá conocer cuál de todas las modalidades ofrecidas en la comunicación, colaboración, coordinación y regulación es más satisfactoria.

Es importante mencionar que la eficiencia es medible respecto al tiempo y esfuerzo; por lo tanto, en esta evaluación, únicamente se considera en el mecanismo de colaboración; debido a que es muy notorio identificar el tiempo empleado para realizar el diagrama de flujo; mientras que para el resto de los mecanismos no es evidente.

En la planeación se consideraron dos formatos a llenarse: 1) Guion de ejecución y 2) Programa de actividades antes, durante y después de las pruebas

Mecanismos de las AC	Heurística	Método		
		Efectividad	Eficiencia	Satisfacción
Comunicación	1,2,3	OPUC	-	Cuestionarios de satisfacción considerando las modalidades
Coordinación	4		-	
Colaboración	8		OPUC	
Regulación	7		-	

Tabla 4 – Métodos de evaluación considerados en Co-flows

4.2.Ejecución

Las pruebas se llevaron a cabo durante cinco días, un equipo cada día, en cada prueba se siguió el formato “Programa de actividades” y los demás formatos generados.

Tres pruebas se llevaron a cabo en el laboratorio de usabilidad de la universidad de Puebla y en un cuarto de observación; debido a que dos usuarios se ubicaban en el laboratorio y el resto (dos o un participante) en el área de observación. Dos pruebas más se realizaron en un laboratorio de cómputo, en la universidad de Veracruz. Durante la ejecución de las 5 pruebas se colocó a los usuarios en lugares separados cada uno en diferente computadora, con el mayor silencio posible y evitando interrupciones externas.

Debido a que Co-flows permite usar diferentes modalidades, se decidió cambiar estas aleatoriamente, para que los usuarios pudiesen evaluar las diferentes modalidades que ofrece Co-flows. Durante la ejecución de las tareas solicitadas se grabaron las interacciones en pantalla y los rostros de cada usuario, para obtener datos que pudiesen ayudar en determinar la efectividad y la eficiencia.

4.3. Reporte y análisis

Participaron 16 estudiantes de carreras afines al área de computación, el 62% eran de 8° semestre, los restantes eran de 10° semestre; todos con experiencia en el manejo de aplicaciones colaborativas. El 56% de los participantes tenían preferencia por el canal sensorial visual, el 25% kinestésico y 19% auditivo.

La efectividad relacionada al mecanismo de comunicación, se puede decir que Co-flows si permitió a los usuarios comunicarse; sin embargo se presentaron algunos errores relacionados a la modalidad de audio, ya que el traductor usado para cambiar de audio a texto no funcionaba bien; otro error fue que el audio se cicló repitiendo el mismo mensaje todo el tiempo o se atrasaban los audios.

Analizando la efectividad de Co-flows en el mecanismo de coordinación; esta aplicación si permitió coordinar a los usuarios sin problemas; ya que retroalimentaba a los usuarios cuando los objetos o herramientas estaban siendo ocupados por alguien más, e informaba de lo que están haciendo los usuarios.

Modalidad	%Satisfacción- Comunicación			%Satisfacción- Coordinación		
	Completa	Media	Nula	Completa	Media	Nula
Audio	64.1	31.3	4.7	68.7	25	6.3
Texto	87.5	12.5	0	81.2	18.8	0
Gráfico	59.4	37.5	3.1	68.7	31.3	0

Tabla 5 – Niveles de satisfacción de la comunicación y coordinación

En la Tabla 5 se muestra la satisfacción en los mecanismos de comunicación y coordinación. Respecto a la comunicación y la coordinación, la modalidad textual proporcionó mayor satisfacción en ambos casos.

El mecanismo de colaboración fue evaluado por la efectividad, eficiencia y satisfacción. Los datos correspondientes a la efectividad mostraron que la colaboración tuvo un poco de problemas, ya que un equipo logró terminar el diagrama, pero los demás quedaron de la siguiente forma: 65%, 56%, 45% y 31% de completitud de la tarea solicitada. Respecto a los problemas encontrados al momento de colaborar, uno de ellos ocurrió al momento de agregar una etiqueta de texto en el área de dibujo, la aplicación agregaba dos, otro problema se relacionó con el equipo que mostró menos avance, ya que no trabajaron en equipo.

La efectividad respecto a la regulación indica que los usuarios pudieron regular sus actividades poco, ya que Co-flows informaba quiénes están conectados pero no era evidente saber si estaban disponibles.

Respecto a la eficiencia en la colaboración, el tiempo para hacer un diagrama variaba entre 15:40 y 11:02 minutos.

La Tabla 6 muestra la satisfacción de la colaboración y la regulación en donde se observa que la modalidad que les dio mayor satisfacción fue la textual.

Al finalizar la prueba, algunos comentarios de los usuarios coincidían en que les parecía interesante utilizar todas las modalidades para interactuar con la aplicación, pero que les gustaría que fuera una combinación de todas, y que ellos pudieran elegir cuándo utilizar una u otra.

Modalidad	%Satisfacción- colaboración			%Satisfacción- regulación		
	Completa	Media	Nula	Completa	Media	Nula
Audio	75	18.8	6.2	62.5	31.2	6.3
Texto	81.2	18.8	0	75	25	0
Gráfico	75	18.7	6.3	68.7	25	6.3

Tabla 6 – Niveles de satisfacción en la colaboración y regulación

5. Conclusiones

La evaluación de la usabilidad de un AC con interfaz de usuario multimodal puede resultar una tarea muy complicada para los evaluadores ya sea novatos o expertos. Ya que además de considerar que la AC provea de los mecanismos de comunicación, colaboración, coordinación y regulación, es necesario identificar el apoyo que recibe el usuario al incluir varias modalidades en la interacción con el sistema y los otros usuarios. Por ello, en este trabajo se propone orientar respecto al proceso de evaluación de usabilidad de las AC-IUM; así como sugerir algunos métodos de evaluación. En esta propuesta la evaluación de AC considera la satisfacción, eficiencia y efectividad, mediante la agrupación de heurísticas de acuerdo a los mecanismos de las AC (colaboración, comunicación, coordinación y regulación).

La propuesta de evaluación de AC-IUM se valida aplicándola a Co-flows, un software de construcción de diagramas de flujo. Al evaluar la usabilidad de la aplicación Co-flows, siguiendo el proceso propuesto en este artículo, se observó una mejor organización de la evaluación, incluso se facilitó seleccionar el método de evaluación adecuado ya que se describe de manera general como evaluar la usabilidad mediante estos métodos; por otro lado fue más fácil seleccionar las tareas ya que estas están relacionadas con los mecanismos de las AC (comunicación, colaboración, coordinación y regulación) y a su vez con las heurísticas. Además la propuesta de evaluación, ayudó a determinar cómo calcular la eficiencia, efectividad y satisfacción de las AC-IUM.

Como trabajo futuro se propone proveer multimodalidad a otros componentes de la interfaz de usuario, con objeto de evaluar otros aspectos de la AC a más profundidad. También es deseable, reclutar usuarios con diferentes canales sensoriales para realizar otras pruebas y comparar resultados. Finalmente, se desea validar las heurísticas y los métodos propuestos con otras aplicaciones que están en desarrollo actualmente.

Referencias

- Araujo, R., Santoro, F., & Borges, M. (2002). The CSCW lab for groupware evaluation. In *Proceedings of CRIWG'02*, (Vol 2440, pp. 222-231). Lecture Notes in Computer Science.
- Baker, K., Greenberg, S., & Gutwin, C. (2002). Empirical development of a heuristic evaluation methodology for shared workspace groupware. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work* (pp. 96-105). ACM.
- Barnum, C. M. (2010). Usability testing essentials: ready, set... test! (pp. 11-13). Elsevier.
- De Araujo, R. M., Santoro, F. M., & Borges, M. R. S. (2004). The CSCW lab ontology for groupware evaluation. In *Proceedings of Computer Supported Cooperative Work in Design*, 2004. The 8th International Conference on (Vol. 2, pp. 148-153). IEEE.
- Dix, A. (2009). Human-computer interaction. In: *Encyclopedia of database systems* (pp. 1327-1331). Springer US.
- Decouchant, D., Mendoza, S., Sánchez, G., & Rodríguez, J. (2013). Adapting groupware systems to changes in the collaborator's context of use, *Expert System with Applications*, 40 (11), 4446–4462.
- Dorohonceanu, B., Sletterink, B., & Marsic, I. (2000). A novel user interface for group collaboration. In *Proceedings of the 33rd Annual Hawaii International Conference on* (pp. 10-pp). IEEE.
- Ellis, C. A., Gibbs, S. J., & Rein, G. (1991). Groupware: some issues and experiences. *Communications of the ACM*, 34(1), 39–58.
- Fernandez, A., Insfran, E., & Abrahão, S. (2011). Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, 53(8), 789–817.
- Gutwin, C., & Greenberg, S. (2000). The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000. (WET ICE 2000). Proceedings. IEEE 9th International Workshops on* (pp. 98-103). IEEE.
- Herskovic, V., Pino, J. A., Ochoa, S. F., & Antunes, P. (2007). Evaluation methods for groupware systems. In *Proceedings of International Conference on Collaboration and Technology* (pp. 328-336). Berlin, Heidelberg: Springer.
- Jaimes, A., & Sebe, N. (2007). Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, 108(1-2), 116–134.
- Montané-Jiménez, L., G., Benítez-Guerrero, E., & Mezura-Godoy, C. (2014). Towards a Context-Aware Framework for Improving Collaboration of Users in Groupware Systems. *EAI Endorsed Transactions on Context-aware Systems and Applications*, 14(1), e4.
- Mezura-Godoy & C. Montané-Jiménez, L. (2008) Modelling Regulated Social Spaces for Groupware Applications, *Research in Computing Science. Advances in Computer Science and Artificial Intelligence*, 39, 47–60.

- Molano, J. I. R., Yara, E. S., & García, L. K. J. (2015). Model for measuring usability of survey mobile apps, by analysis of usability evaluation methods and attributes. In *Proceedings of Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on* (pp. 1-6). IEEE.
- Moller, S., Engelbrecht, K. P., Kuhnel, C., Wechsung, I., & Weiss, B. (2009). A taxonomy of quality of service and quality of experience of multimodal human-machine interaction. In *Proceedings of Quality of Multimedia Experience, 2009. QoMEX 2009. International Workshop on* (pp. 7-12). IEEE.
- Pinelle, D., Gutwin, C., & Greenberg, S. (2003). Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 10(4), 281–311.
- Pinelle, D., & Gutwin, C. (2002). Groupware walkthrough: Adding context to groupware usability evaluation. In *Proceedings of the 2002 SIGCHI Conference on Human Factors in Computing Systems* (pp. 455–462). ACM Press.
- Pinsonneault, A., & Kraemer, K. (1989). The impact of technological support on groups: An assessment of the empirical research. *Decision Support Systems*, 5(3), 197–216.
- Quiñonez, Y., Luzardo, G. & Granda, R. (2016). Implementación de un sistema multi-táctil en ambientes educativos para promover y facilitar la evaluación del trabajo colaborativo en el aula. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, 17, 57–65.
- Reyes, I. A., Mezura-Godoy, C., & Sánchez, G. (2016). Hacia un modelo de interfaces multimodales adaptables a los canales de aprendizaje en aplicaciones colaborativas como apoyo a la educación. *Research in Computing Science*, 111, 57–67.
- Rogers, Y., Sharp, H., & Preece, J. (2011). *Interaction design: beyond human-computer interaction*. Hoboken, NJ, USA: John Wiley & Sons.
- Saltiveri, T. G., Fransi, E. C., & Salla, Y. M.. (2013). Análisis de usabilidad de cooperativas del sector de la fruta y aceite en el área de Lleida. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, 11, 45–59.
- Sánchez, G., Mezura-Godoy, C., Reyes, I. A., & Benítez-Guerrero (2017, October). Evaluación de usabilidad de aplicaciones colaborativas con interfaces de usuario multimodal, Applications in Software Engineering. In *Proceedings of the 6th International Conference on Software Process Improvement (CIMPS)* (pp. 119,127), IEEE.
- Sánchez, G., Mezura-Godoy, C., & Benítez-Guerrero (2018). Proceso de Pruebas de Usabilidad de Software. *Tecnología Educativa Revista CONAIC*, 5(1).
- Vanderdonckt, J., Calvary, G., Coutaz, J., & Stanciulescu, A. (2008). Multimodality for plastic user interfaces: models, methods, and principles. In *Multimodal User Interfaces* (pp. 61-84). Berlin, Heidelberg: Springer.
- Yu, Z., Guo, W., & Zheng, X. (2013, October). A brief review: Groupware evaluation by performance of specific tasks. In *TENCON 2013-2013 IEEE Region 10 Conference* (31194) (pp. 1-4). IEEE.

Una Metodología para Procesos Data WareHousing Basada en la Experiencia

Wilson Castillo-Rojas ¹, Fernando Medina Quispe ², Francisco Fariña Molina ²

wilson.castillo@uda.cl, femedina@unap.cl, franciscofarina@unap.cl

¹ Universidad de Atacama, Facultad de Ingeniería / DIICC, 1530000, Copiapó, Chile.

² Universidad Arturo Prat, Facultad de Ingeniería y Arquitectura, 1100000, Iquique, Chile.

DOI: 10.17013/risti.26.83–103

Resumen: El artículo presenta una nueva metodología para procesos data warehousing, que integra diversos enfoques, técnicas y metodologías, tales como: especificación de requisitos de información, modelamiento relacional, modelo de desarrollo combinado a partir de las propuestas de Kimball y Hefesto, un proceso aumentado de extracción-transformación y carga que incorpora explícitamente una fase de validación de indicadores, y finalmente visualizaciones integradas e interactivas para el análisis multidimensional de los indicadores obtenidos. La metodología propuesta no sólo se basa en los aspectos teóricos descritos en el artículo, sino que además en la experiencia lograda por parte del equipo investigador, en el desarrollo de diversos proyectos de data warehousing, principalmente orientados a la generación de indicadores de productividad académica y de gestión de una universidad, lo que corresponde al caso de estudio de aplicación de la metodología que se describe. Los resultados de éxito en diferentes proyectos donde ha sido utilizada esta metodología avalan su eficacia.

Palabras-clave: Inteligencia de negocios, almacén de datos, data warehousing, datamart, ETL, OLAP.

A Methodology for Data WareHousing Processes Based on Experience

Abstract: The article presents a new methodology for data warehousing processes, which integrates different approaches, techniques and methodologies, such as: specification of information requirements, relational modeling, combined development model based on the Kimball and Hefesto proposals, an increased process of extraction-transformation and load that explicitly incorporates a validation phase of indicators, and finally integrated and interactive visualizations for the multidimensional analysis of the obtained indicators. The proposed methodology is not only based on the theoretical aspects described in the article, but also on the experience gained by the research team in the development of various data warehousing projects, mainly oriented to the generation of indicators of academic productivity and management of a university, which corresponds to the case study application of the methodology described. The results of success in different projects where this methodology has been used guarantee its effectiveness.

Keywords: Business intelligence, data warehouse, data warehousing, datamart, ETL, OLAP.

1. Introducción

Actualmente las organizaciones utilizan la información y el conocimiento para apoyar la toma de sus decisiones estratégicas, y de este modo lograr sus metas y mejorar sus procesos. Lo anterior, en un contexto actual de alta competitividad y requerimiento de certificación de procesos, globalización de mercados, etc. En el ámbito de educación superior, las universidades en Chile se encuentran en procesos de alta exigencia para asegurar la calidad en su gestión administrativa y académica, a través de procesos de acreditación institucional donde estas instituciones deben rendir cuentas de toda su actividad académica.

Uno de los desafíos que enfrentan hoy las organizaciones, es el aumento de datos, lo que ha generado dos grandes problemas; el primero, identificar los datos relevantes para dar seguimiento a su estrategia organizacional, y lograr que se cumplan los planes con las metas establecidas. Y el segundo problema, la capacidad para administrar esta gran cantidad de datos.

Para lo anterior, las organizaciones requieren no solamente sistemas de información para dar soporte tecnológico a sus procesos operacionales, sino que requieren de soluciones tecnológicas que les proporcionen, por un lado, indicadores que les permitan medir el desempeño de sus procesos de gestión en general y, por otro lado, patrones o reglas que les permitan describir o predecir comportamientos en sus movimientos o transacciones de sus procesos de negocios. Todo esto, para apoyar la toma de decisión estratégica de una manera mejor preparada (Illescas, Sanchez & Canziani, 2015), (Prieto & Piattini, (2015).

Lo descrito se puede lograr con el uso de tecnologías y herramientas de análisis de datos masivos, que permiten la administración y generación de conocimiento utilizando data histórica de la organización. Entre estas tecnologías se encuentran; herramientas ETL, almacenes de datos o Data Warehouse (DW), herramientas OLAP, minería de datos y otras. Y que son parte de lo que se conoce como Inteligencia de Negocios (IN o BI; acrónimo del inglés Business Intelligence).

Una definición formal de IN, es que corresponde al conjunto de estrategias y herramientas tecnológicas enfocadas a la administración y creación de conocimiento, a través del análisis de datos existentes en una organización (Back, 2002) (Golfarelli, Rizzi & Cella, 2004). Si bien un DW corresponde al repositorio de datos central de un sistema de IN, su diseño, desarrollo e implementación son parte de lo que se denomina Proceso de Data Warehousing (PDW).

Estas tecnologías son muy demandadas actualmente por las organizaciones, que teniendo satisfechas las prestaciones técnicas de sus sistemas informáticos para apoyar los procesos operacionales, requieren de nuevas prestaciones que les permitan mantenerse competitivas en un entorno cambiante y globalizado, siendo los DW una de las tecnologías IN con más desarrollo de proyectos en Chile (Palocsay, Markham & Markham, 2010).

En las instituciones de educación superior, también ha ido en aumento la cantidad de universidades, que utilizan la tecnología de IN para obtener indicadores de su gestión, y con esto enfrentar de mejor manera un proceso de acreditación. Para esto, requieren contar con: unidades administrativas de análisis institucional, recurso humano preparado en tecnología de IN, herramientas de IN, y proyectos de desarrollo de sistemas orientados hacia la toma de decisiones estratégicas.

Bajo este contexto, el artículo presenta y describe una metodología para el desarrollo de PDW que se basa principalmente en la experiencia de desarrollo de varios proyectos llevados a cabo en la Universidad Arturo Prat de Chile (UNAP). Proyectos orientados a obtener indicadores de productividad académica y de gestión de la universidad. No obstante, para elaborar y formalizar esta nueva metodología, se toman como base conceptual dos modelos de desarrollo de PDW, por un lado, el modelo clásico de desarrollo para un DW de (Kimball & Ross, 2002), combinado con la actual metodología Hefesto de (Bernabeu, 2010), de ambas se extraen las fases comunes y se refunden.

Adicionalmente, la metodología propuesta integra diversos enfoques, técnicas y metodologías, tales como: especificación de requisitos de información utilizada en ingeniería de software, proceso aumentado de ETL con una fase de validación de indicadores (ETL+V; Extraction, Transformation, Loading, and Validation), y visualizaciones integradas e interactivas para el análisis multidimensional de los indicadores, basado en el concepto de tablero de control (dashboard en inglés).

La estructura del artículo es la siguiente: la sección 2 provee un marco teórico sobre conceptos y tecnologías asociadas al trabajo. La sección 3 describe la metodología propuesta. La sección 4 presenta la aplicación de esta metodología en el desarrollo de un PDW para la generación de indicadores de productividad académica de una universidad. En la sección 5 se presentan las conclusiones del trabajo. Finalmente, en la sección 6 se listan las referencias bibliográficas.

2. Marco Teórico

En esta era conocida como la sociedad del conocimiento, una de las características tecnológicas es la utilización y consolidación de las bases de datos, para el registro de transacciones a través de Internet, lo cual representa actualmente, la herramienta principal que proporciona información en una organización. Sin embargo, con el paso de los años esta información tiende a crecer y generar grandes volúmenes de datos, por lo que surge la necesidad de cómo manejarla y qué hacer con ella. Es aquí donde aparece el término IN, que surge como solución para analizar y explotar áreas específicas de información, generando nuevas perspectivas y conocimiento con el fin de apoyar la toma de decisiones.

Una de las primeras definiciones de IN se conoce a partir de 1958 con Hans Petter Luhn, investigador de IBM que lo define como: *“La habilidad de aprehender las relaciones de hechos presentados de forma que guíen las acciones hacia una meta deseada”* (Conesa & Curto, 2010), para luego en la década de los 60 dar origen a los Sistemas de Soporte de Decisiones (SSD). Luego evolucionan en Sistemas de Información Ejecutivos (SIE), que son sistemas más robustos capaces de generar reportes consolidados. Si bien estos

sistemas cumplen su labor en la administración de la información, aún son insuficientes, ya que estas herramientas no cuentan adecuadamente, con visualización y explotación de la información.

El término IN se acuña formalmente en el año 1989 por Howard Dresden, analista de Gartner que lo define como: *“Conceptos y métodos para mejorar las decisiones de negocio mediante el uso de sistemas de soporte basado en hechos”* (Venter, 2005). IN actualmente se encuentra presente en las organizaciones más prestigiosas del mundo, ya que sirve como respaldo y soporte a la toma de decisiones de la parte estratégica de una organización, mediante el análisis de gran cantidad de datos en forma rápida y sencilla, que son procesados según reglas y criterios del negocio. IN es un concepto que se aplica de forma transversal a cualquier tipo de negocio, además de integrar información de diferentes procesos, en distintos periodos, generando un análisis completo de la situación actual de la organización, con el fin de progresar en el rendimiento interno de sus procesos.

Desde un punto de vista pragmático, IN es el conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento a través del análisis de datos existentes en una organización. Esto permite a la organización, contar con información privilegiada para responder a problemas internos del negocio como: optimización de costos, rentabilidad tanto de clientes, obtener perfiles de clientes, mejorar procesos de producción entre otras, todo lo anterior se puede responder de forma rápida y eficiente, con un buen proceso e implementación de la IN, lo cual brinda a la organización una ventaja competitiva y estratégica en el mercado (Mazón, Trujillo & Lechtemborger, 2007).

Una de las tecnologías centrales de IN son los DW y DataMart (DM). Según la clásica y más referenciada definición de Inmon, un DW: *“es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza”* (Inmon, 2005).

Un DW corresponde al almacén de datos corporativo que abarca todas las áreas o procesos de la organización. El origen de esta información proviene de diferentes fuentes de datos, ya sean de sistemas operacionales, plantillas de cálculos o archivos de textos planos. Estas fuentes de datos se integran para darle un formato homogéneo y consistente a los datos. Además, un DW se compone de bases de datos multidimensionales denominadas DM, que se enfocan al análisis de un área de negocio en particular, y tiene como objeto proporcionar indicadores o KPI.

Existen 2 enfoques respecto a los conceptos DW y DM (Kimball & Ross, 2002):

- Bill Inmon propone la idea que los DM se sirven de un DW para extraer información que esta almacenada de manera estructurada en un modelo relacional, a esto se le conoce como enfoque top-down.
- Ralph Kimball, plantea la idea de un enfoque dimensional para el diseño de un DW, y afirma que la unión de todos los DM de una organización constituye el DW corporativo, a lo cual se le conoce como el enfoque bottom-up.

El esquema convencional de un sistema de IN lo componen las siguientes etapas (Vaisman & Zimányi, 2014):

- i. Especificación de requisitos estratégicos; que corresponde a la definición del proyecto IN, la naturaleza del negocio y sus propósitos, así como la especificación de indicadores claves de desempeño (Key Performance Indicator ó KPI), que se requieren medir en los procesos de la organización.
- ii. El proceso ETL; tiene como función la integración de los datos provenientes desde las distintas fuentes heterogéneas, ya sean de sistemas transaccionales, archivo de textos, planillas de cálculos, etc. La integración consiste en la extracción, transformación, cálculos preliminares de KPI, limpieza y homogenización de datos, y carga de datos en el DW. Para esto, se pueden utilizar herramientas ETL, lenguajes de programación y/o lenguaje de consultas de base de datos relacional (Structured Query Language ó SQL).
- iii. El DW; corresponde al repositorio de datos, cuyo diseño conceptual está basado en un Modelo Multidimensional (MM). Esto es, considera a la base de datos como un conjunto de hechos u objeto de análisis, y dimensiones que son los puntos de vistas desde los que se pueden analizar estos hechos. Las informaciones relevantes sobre los hechos se representan por un conjunto de indicadores o medidas (valores numéricos) y que corresponde a los KPI a calcular. La información descriptiva de cada dimensión se presenta por un conjunto de atributos alfanuméricos. La tecnología utilizada por lo general son sistemas gestores de base de datos relacionales.
- iv. Explotación del DW; a través de herramientas de análisis de datos tales como: OLAP (On-Line Analytical Processing), data mining, generadores de reportes y gráficos estadísticos, dashboard, y en general todo tipo de SSD.

Como se señala en la sección anterior, un PDW corresponde al proceso que permite el diseño, implementación y explotación de un DW. Por lo general, un PDW incluye las estrategias y tecnologías descritas en i), ii), iii), y respecto a la explotación del DW, las que tienen relación al procesamiento analítico en línea (OLAP).

3. Metodología Propuesta

La metodología elaborada y propuesta en este trabajo, recoge la experiencia obtenida por el desarrollo de diferentes proyectos de PDW, orientados a generar indicadores o KPI que miden el desempeño y productividad, tanto académica como de gestión, de una universidad chilena. Desde el punto de vista conceptual, esta metodología integra diversos enfoques, técnicas y metodologías.

La elaboración de esta metodología, se basa en el modelo clásico de desarrollo para un DW de (Kimball & Ross, 2002) combinado con la actual metodología Hefesto de (Bernabeu, 2010), de ambas se extraen las fases comunes y se refunden. La metodología de Kimball se basa en lo que denomina Ciclo de Vida Dimensional del Negocio (Business Dimensional Lifecycle). Se consideran de esta metodología las siguientes etapas: definición de requerimientos del negocio, modelado dimensional, diseño físico, y diseño e implementación del subsistema ETL. Por otro lado, la metodología Hefesto consta de 4 etapas: análisis de requerimientos, análisis de los OLTP, modelo lógico del DW, e

integración de datos. Estas etapas son consideradas, pero refundiéndolas y dando un orden de desarrollo distinto.

Como se puede observar en la Figura 1, se funden las primeras etapas de ambas metodologías, en una sola etapa inicial llamada Análisis en la nueva metodología, agregando a ésta la segunda etapa de Hefesto (análisis de los OLTP). Luego, se combinan las etapas modelado dimensional y diseño físico de Kimball con la tercera etapa de Hefesto (modelo lógico del DW), lo que da a lugar la segunda etapa denominada Diseño MM. En la tercera etapa denominada Proceso ETL+V, se integra la etapa diseño e implementación del subsistema ETL de Kimball, con la etapa integración de datos de Hefesto. La cuarta etapa Procesamiento Analítico, corresponde a una fusión entre lo que propone Kimball en su ciclo de vida de un DW y las de varios autores, que consideran a OLAP, como una opción de herramienta, entre otras, para la explotación del almacén y no como parte de su método de desarrollo.

Cada una de estas etapas tienen definidas sus entradas y salidas, representadas a través de flechas con línea segmentada. A su vez, estas etapas tienen un sentido bidireccional de funcionamiento, que permite retroceder a la anterior o avanzar a la siguiente, sin perder el sentido del PDW. También, se establece la relación y participación de los usuarios, con flecha de línea continua y en sentido bidireccional, tanto en la entrada al proceso, como en la salida.

Se destacan en esta metodología, elementos claves señalados de la Figura 1 con un círculo de color rojo: plantilla ad-hoc para la especificación de los KPI, repositorio temporal como área de staging, el código SQL de validación de los KPI, y las visualizaciones gráficas integradas a través de tableros de control o dashboard. Lo anterior, según las conclusiones del equipo de desarrollo del DW y los usuarios finales, favorecieron la efectividad y éxito del desarrollo de los proyectos en que se ha utilizado esta metodología. Se explica a continuación, cada una de las etapas de la metodología propuesta en este trabajo.

3.1. Etapa Análisis.

Esta etapa inicial se vincula fuertemente con los usuarios estratégicos. Consta de dos actividades; análisis de los requerimientos de KPI, y análisis de los sistemas operacionales. Para la primera actividad, la entrada corresponde a una plantilla ad-hoc diseñada para la especificación de los requisitos de información estratégica o KPI. La estructura de esta plantilla, es considerada como uno de los elementos claves en el PDW, y se diseña tomando como referencia las plantillas y patrones lingüísticos de (Duran, 2000), que están enfocados para el proceso de educación de requisitos de información en un proyecto de software. Esta plantilla, permite definir y entender claramente, por parte de todos los usuarios, los indicadores obtenidos, tanto como interfaz o herramienta, en la educación inicial de los requisitos de KPI, su análisis, así como en la validación de cada uno. La salida de esta actividad, es el conjunto de KPI especificados a través de la plantilla diseñada.

La segunda actividad de esta etapa, corresponde al análisis de los sistemas operacionales, cuyas entradas la componen las fuentes de datos de la organización, ya sean internas o externas, más toda la información acerca de los datos que se dispongan, tales como

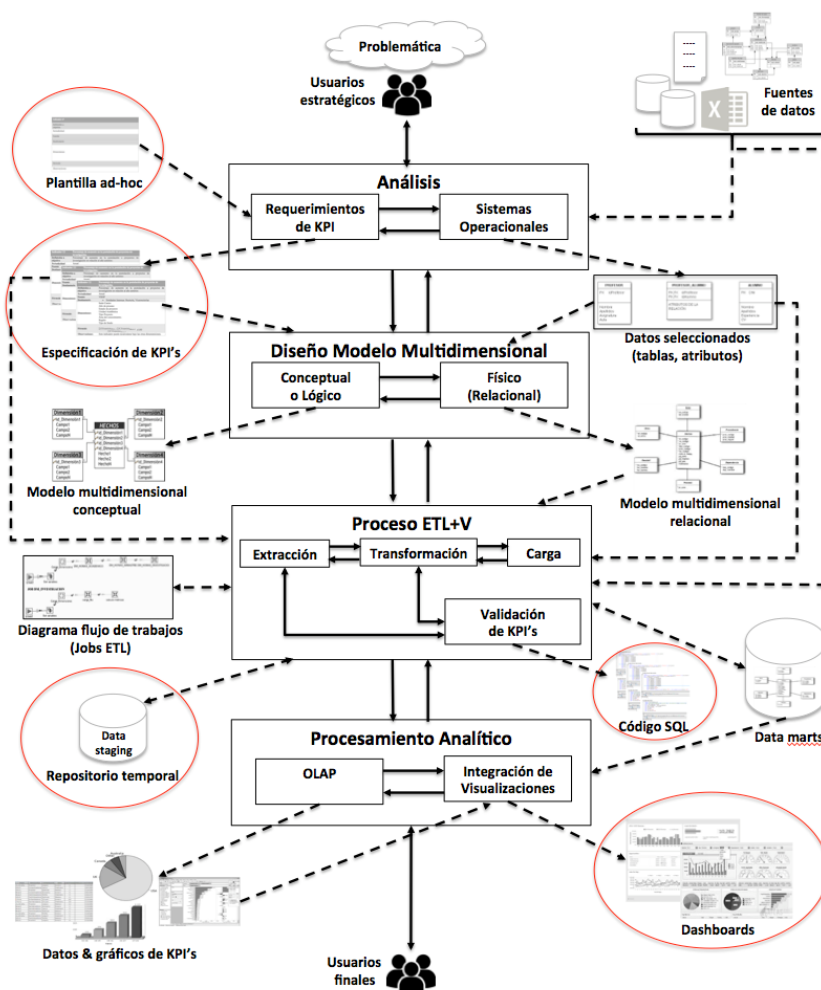


Figura 1 – Metodología propuesta para PDW (elaboración propia)

modelo dato-lógico, meta-datos, etc. Estas fuentes de datos, deben ser analizadas para contrastar los KPI especificados, y entender con qué datos poder realizar sus cálculos, o en caso de no contar con ellos poder obtenerlos. Tiene como salida esta actividad, el meta-datos con la selección de tablas, atributos y datos, que son necesarios para lograr calcular los KPI requeridos.

3.2. Etapa Diseño del MM.

Las salidas de la etapa de análisis, son las entradas en esta etapa. Y en ella, se encuentran dos actividades; el diseño conceptual y el diseño físico (relacional) del MM, cuyas salidas, corresponden a los modelos bajo el mismo nombre, conceptual y físico respectivamente. Para esto, es necesario analizar los KPI especificados junto al meta-datos de las fuentes

de origen, para poder realizar el diseño conceptual. La parte tecnológica de esta etapa la constituye las bases de datos.

Se recomienda utilizar una herramienta de la tecnología CASE (Computer Aided Software Engineering), donde se puede utilizar una de las notaciones gráficas existentes para diagramar el MM, ya sea entidad-relación, UML (Unified Modeling Language), u otra. Para generar el diseño físico del MM, en este caso relacional ya que se utiliza el enfoque de Kimball, basta con utilizar la opción en la herramienta CASE para generarlo, indicando algunos parámetros como: el motor de base de datos, nombre del servidor, y otros datos de conexión.

3.3. Etapa Proceso ETL+V.

La tercera etapa atañe al clásico proceso ETL de extracción, transformación y carga, aumentado con una cuarta actividad incorporada explícitamente, y denominada validación de KPI. Se trata de la etapa más técnica y que mayor tiempo demanda, la de mayor complejidad, y la que más entradas y salidas tiene. Las entradas a esta etapa son; la especificación de los KPI, el MM relacional, la descripción (meta-datos) de los datos seleccionados, y las fuentes de datos originales. Sus salidas corresponden a; diagrama de flujo de trabajo del proceso ETL (generada a través de una herramienta o software para esto, tal como Kettle de Pentaho u otras, o simplemente con el uso de lenguaje SQL), el repositorio temporal o área de staging, los DM cargados con los KPI, y el código SQL de validación.

La etapa ETL+V, cuya variante es la más destacada en este trabajo, tiene como función realizar toda la extracción y cálculo preliminar de los datos desde las fuentes de datos originales, y que necesarios para el cálculo de los KPI, para luego ser depositados en el repositorio temporal relacional. En este repositorio, se realizan todas las transformaciones, cálculos, agregaciones y validaciones de calidad de los datos. También se debe realizar la actividad de carga de los datos hacia los DM con los KPI ya calculados.

Sin embargo, antes de la carga es necesario verificar la validez de los KPI. Para esto, la actividad de validación, permite realizar la revisión de estos en forma eficiente, utilizando el lenguaje de consulta para base de datos relacionales (SQL), y así poder realizar consultas, cálculos y validaciones necesarias. Se trata de verificar que los KPI obtenidos sean consistentes, con los datos integrados en el repositorio temporal, las fuentes de datos originales, y el DM.

3.4. Etapa Procesamiento Analítico.

La última etapa corresponde a la explotación y análisis de la información contenida en los DM, y para esto se requiere una herramienta de la tecnología OLAP, con la cual generar el análisis multidimensional de los datos, y las visualizaciones gráficas de los KPI. Adicionalmente, se considera como segunda actividad la integración de visualizaciones de los distintos objetos de análisis y KPI, utilizando el concepto de tablero de control o dashboard (Marcus, 2006), y que permiten proporcionar a los usuarios finales, una mirada integral de los indicadores provistos por los DM, a través de interfaces gráficas y con reportabilidad interactiva e intuitiva.

4. Caso de Uso: PDW de Productividad Académica

Se describe a continuación, la aplicación de la metodología propuesta para un caso de uso en particular, y que corresponde a un PDW cuyo objetivo es generar indicadores de productividad académica de una universidad chilena.

4.1. Descripción de la Problemática.

En su plan estratégico institucional, la UNAP declara como eje estratégico la “*Gestión Moderna y competitiva*”, que apunta a administrar de manera eficiente los procesos de gestión institucional tanto en recursos humanos, infraestructura y sistemas de información. Para esto, se requiere el fortalecimiento de los sistemas de información a nivel de reportes de indicadores, para la toma de decisiones internas en vicerrectorías, sedes, facultades y centros docentes de la institución.

La Unidad de Análisis Institucional (UAI) de la UNAP, es la encargada de generar y administrar los DM de la institución, con el fin de poder entregar informes tanto cualitativos como cuantitativos del estado actual de la gestión de las diferentes áreas de la institución. Para esta ocasión, la UAI plantea la necesidad del desarrollo de un PDW cuyo objetivo es obtener indicadores sobre la productividad académica. Esto, con el propósito de dar respuesta a requerimientos de información tanto de entidades internas y externas a la Universidad. Además, se busca definir y establecer criterios para la obtención de estos indicadores.

Por lo anterior, se realiza un análisis de los procesos de internos involucrados en la productividad académica de todos los funcionarios del estamento académico de la institución. Se plantean tres aspectos fundamentales dentro de sus actividades que son: docencia, investigación y vinculación con el medio. Para este trabajo, se llega a acuerdo en abordar en las dos primeras labores, ya que son las más requeridas en el trabajo diario de la universidad.

En cuanto a la actividad de docencia, los indicadores buscan conocer como es la distribución del personal académico en las distintas institutos, sedes, facultades y carreras de la universidad, como también la cantidad de horas semanales que dedican a sus labores docentes. También es relevante conocer la cantidad de jornadas completas equivalentes que los docentes representan en la institución, ya que es un indicador que se compara con la cantidad de alumnos matriculados, con el fin de establecer la relación que existen entre ellos.

En lo que concierne a la actividad de investigación, los indicadores deben permitir establecer la producción científica de la universidad, a través de medidas tales como la cantidad de publicaciones y proyectos de investigación realizadas por los académicos de las distintas institutos, sedes, facultades y carreras. Además, se buscan determinar porcentajes y tasas de aumento entre los periodos estudiados. Otro aspecto importante es, conocer el porcentaje de docentes con grado de doctor que realizan las labores de investigación, ya que con este indicador se pueden tomar medidas preventivas o de incentivo, para aumentar la productividad científica, y de ser necesario, contratar nuevos académicos con grado de doctor.

4.2.Etapa Análisis: Requerimientos de KPI.

La toma de requerimientos se lleva a cabo a través de técnicas de educción que son utilizadas en un proceso de software, con la salvedad que el tipo de usuario corresponde al nivel estratégico de la institución. Por lo anterior, se realizan un conjunto de reuniones ejecutivas con las dos vicerrectorías (académica y de investigación) en conjunto con la UAI, que están directamente ligadas a la problemática, con el fin de tomar las inquietudes y necesidades sobre la productividad académica de estas entidades tanto en sus actividades docentes y de investigación. Y de este modo, lograr definir y especificar los KPI requeridos.

También, se consideran relevante en esta etapa, los requerimientos externos provistos por instituciones de gobierno y otras, tales como: Consejo Nacional de Acreditación (CNA), Sistema de Información de Educación Superior (SIES), Consejo de Rectores de Universidad Chilenas (CRUCH), y el Consejo Nacional de Educación (CNED). Por esto, se realiza también un análisis de todos los informes entregados por estas instituciones, y de sus instructivos de solicitud de información. Con ambas visiones, interna y externa, se consolidan los requerimientos de productividad académica, de docencia y de investigación, utilizando la plantilla de requerimientos ad-hoc diseñada para esta actividad. Como muestra se presenta en la Figura 2, la especificación de un KPI de investigación, y uno de docencia en la Figura 3, de un total de catorce especificados (7 de cada uno).

Indicador 11	Tasa anual de Crecimiento en la adjudicación de proyectos de investigación
Definición u objetivo	Representa el valor de aumento o disminución de la relación que existe entre los proyectos adjudicado de un periodo a otro.
Periodicidad	Anual.
Fuente	- Entidades Externas: - - Entidades Internas: Rectoría, Vicerrectorías
Destinatario	VRIP
Dimensiones	Sede-Centro Año de proceso Estado de proyecto Unidad Académica Tipo Proyecto Área del conocimiento. Región Tipo de fondo
Fórmula	$\frac{\sum N^{\circ}Proyectos_{Año\ x} - \sum N^{\circ}Proyectos_{Año\ x-1}}{\sum N^{\circ}Proyectos_{Año\ x-1}}$
Observaciones	Si el valor es positivo significa que aumento de uno año a otro, pero si es negativo implica una disminución en el indicador.

Figura 2 – Indicador: Tasa anual de crecimiento en proyectos de investigación

Se puede observar en ambas figuras, la detallada especificación del indicador que incluye, no sólo su descripción y objetivo, sino que además su periodicidad, las dimensiones involucradas, las fórmulas sobre la cual se calculan los KPI. Toda esta información es acordada y validad por los usuarios estratégicos, y la plantilla sirve como interfaz de comunicación en todo el proceso de educción de requerimientos.

Indicador 5	Nº Jornada Completa Equivalente
Definición u objetivo	Mide la equivalencia en jornada completa de las horas realizadas por el total de académicos contratados.
Periodicidad	Anual, Semestral, Mensual
Fuente	Unidad de Programación y Registro Académico, RR.HH
Destinatario	- Entidades Externas: SIES, CNA, CRUCH, Agencias Acreditadoras, CNED, proyectos MECESUP. - Entidades Internas: Rectoría, Vicerrectorías, DFT, Direcciones, Facultades, Carreras.
Dimensiones	Sede Año de proceso – Semestre - Mes Grado Académico Nacionalidad Sexo Unidad Académica Carreras Sede Nivel Formación Tipo Contrato Tipo Jornada
Fórmula	$JCE = \frac{\sum TotalJC + TotalJM + TotalJH}{44}$ Donde: $TotalJC = \left(\sum_i^n DocentesJC_i \right) \cdot 44$ $TotalJM = \left(\sum_i^n DocentesJM_i \right) \cdot 22$ $TotalJH = \sum Cant. Horas_i \cdot NumDocentesJH_i$ JC=Jornada Completa JM=Jornada Media JH=Jornada Hora
Observaciones	Para el caso del TotalJH el índice i representa la cantidad de horas a multiplicar por el número de docentes que realizan dicha cantidad de horas.

Figura 3 – Indicador: Número de jornadas completas equivalentes

4.3. Etapa Análisis: Sistemas Operacionales.

Una vez definidos los indicadores, se prosigue con la identificación de las fuentes de datos que son necesarias para su cálculo. Para esta actividad, es necesario reunirse con la Unidad de Informática y Comunicaciones (UNICO) de la institución, en conjunto con la UAI, con el objeto de identificar los diferentes modelos de datos involucrados y seleccionar las fuentes de datos. Luego de varias revisiones y análisis a los sistemas de información que maneja la UNAP, se filtran los que se van a utilizar, dando como resultado los que se listan en la Tabla 1.

Sistema	Descripción de la fuente de datos
ICON	Sistema contable y financiero.
SIPER	Sistema de Personal.
SICDO	Sistema Curricular y Docente.
GEDO	Sistema de Gestión de Documentos Online.
VRIP	Sistema que almacena proyectos de investigación.

Tabla 1 – Selección de fuentes de datos a utilizar

Se obtiene de esta actividad; el listado de los sistemas que contienen los datos a utilizar, el meta-datos de estos, su modelo dato-lógico, y se seleccionan las tablas y atributos específicos que se requieren para el cálculo de los KPI.

4.4.Etapa Diseño: MM Conceptual.

Debido a que la UNAP cuenta con algunos DM previamente desarrollados, es necesario conocer las dimensiones que se encuentran presentes en estos, con el objeto de reutilizar la información en la elaboración de los nuevos DM, y así evitar la redundancia de datos junto con la duplicidad de dimensiones a nivel institucional. Luego de analizar los requerimientos y modelos existentes, se elaboran dos MM, para cumplir con los indicadores a obtener.

Uno de estos modelos enfocado a los indicadores de docencia, y el otro de investigación. El primer modelo es del tipo esquema “Copo de Nieve” (Snowflake), donde se hace un trabajo de jerarquización de algunas dimensiones, y reutilización de otras. Por otro lado, el DM de investigación es un esquema estrella, y la gran mayoría de sus dimensiones son creadas en este trabajo. Las dimensiones utilizadas en este PDW, se puede ver en la Tabla 2.

En cuanto a los indicadores de ambos MM, se presentan en las Figuras 4 a) y b), en sus respectivas tablas hechos con los KPI especificados para el esquema copo de nieve de docencia, y estrella de investigación.

DIM_HORAS_ACADEMICO	
HORA_SEMANAL_ACT	Number
HORA_MENSUAL_ACT	Number
HORA_SEMANAL_CONT	Number
HORA_SEM_PF	Number
HORA_SEM_PP	Number
JCE	Number

a)

DM_INVESTIGACION	
NRO_PUBLICACIONES_PP	Number
NRO_PUBLICACIONES	Number
NRO_PROYECTOS	Number
NRO_PROYECTOS_PP	Number

b)

Figura 4 – Tablas de hechos: a) docencia, b) investigación

Dimensión	Descripción
Tiempo	Meses, semestres y años.
Académico	Nombre, fecha nac., fecha ingreso, sexo y nacionalidad.
Jerarquía	identifica la jerarquía docente de cada académico, estas son: “Profesor Asociado”, “Profesor Titular”, “Profesor Asistente”, “Instructor”, “Sin Jerarquía”.
Forma Contractual	Tipos de contratos: Jornada Completa, Media Jornada o Horas.
Grado Académico	Doctorado, Magister, Profesional, Licenciado, Técnico y Sin título.
Tipo de Actividad	“Docencia” o “Investigación”.
Carrera	Actividad del docente asociado a una carrera, cuenta con 3 jerarquías distintas: unidad académica, sede, y nivel de formación.
Tipo de Formación	Tipo de asignatura: Formación General o Formación Profesional.
Fuente de Financiamiento	Internos (UNAP), o externos (públicos o privados).
Área del Conocimiento	Clasificación de un programa académico: Agropecuaria y Ciencias del Mar, Administración y Comercio, Arte y Arquitectura, Ciencias Naturales y Matemáticas, Ciencias Sociales, Derecho, Humanidades, Educación, Tecnología y Salud.
Región	Nombre de regiones
Tipo de Proyecto	El tipo de proyecto categoriza en investigación o prestación de servicios a las postulaciones que realiza un docente a un proyecto en particular.
Estado del Proyecto	Estado del proyecto: adjudicado, en postulación, decretado, o finalizado.
Formato de Publicación	Artículo científico, capítulo de libro, revisión de artículo (review).
Tipo de Indexación	ISI, SCOPUS y SCIELO.
Posición de la Publicación	Que tiene la institución en un artículo científico.

Tabla 2 – Dimensiones a utilizar en el MM

En la Tabla 3 se describen cada uno de los indicadores considerados en el diseño conceptual del MM, para estas dos tablas de hechos.

Indicador	Descripción (hecho docencia)
<i>HORA_SEMANAL_ACT</i>	Cantidad de horas semanales que imparte un docente.
<i>HORA_MENSUAL_ACT</i>	Total de horas mensuales de un docente en una actividad.
<i>HORA_SEM_PF</i>	Cantidad de horas semanales que un docente realiza.
<i>HORA_SEM_PP</i>	Total de horas semanales que imparte un docente.
<i>JCE</i>	Cantidad de Jornada Completas Equivalentes.
Indicador	Descripción (hecho Investigación)
<i>NRO_PUBLICACIONES_PP</i>	Cantidad de publicaciones proporcionales.
<i>NRO_PUBLICACIONES</i>	Cantidad total de publicaciones que realiza el docente.
<i>NRO_PROYECTOS</i>	Cantidad total de proyectos en el que participa un docente.
<i>NRO_PROYECTOS_PP</i>	Número de proyectos proporcionales que realiza el docente.

Tabla 3 – Descripción de indicadores

Una vez definidas las componentes (hechos y dimensiones), se diagraman los diseños conceptuales de ambos MM en una herramienta CASE, conectándolos a través de las dimensiones comunes, y el nivel de las jerarquías que sea necesario. En este trabajo se utiliza la herramienta CASE PowerDesigner.

4.5. Etapa Diseño: Modelo Relacional del MM.

Una vez terminado el diseño conceptual, se genera el modelo relacional a través de la herramienta CASE, generando el código script en lenguaje de definición de datos (o DDL: Data Definition Language), indicando el motor de base de datos a utilizar en la implementación. Se presenta como anexo al final del artículo, el MM en la Figura 12, esto debido al tamaño de la imagen.

4.6. Etapa Proceso ETL+V: Extracción.

Para este proceso, se utiliza la herramienta Pentaho's Data Integration (PDI), que permite modelar el proceso ETL con diagramas de flujos de transformaciones, que consisten en un conjunto de pasos fijos encapsulados en flujos integrados denominados trabajos (job), que pueden contener sentencias o script de ejecución.

En la actividad de extracción, se utiliza el meta-datos, donde está registrada la información de las fuentes de datos seleccionadas, en conjunto con las especificaciones de los KPI, y se proceden a extraer los datos y se van dejando de manera agregada, en caso de ser necesario, directamente en el repositorio temporal. Se genera un flujo de trabajo que realiza las transformaciones necesarias para esta parte del proceso, y a modo de ejemplo se presenta en la Figura 5 un job que extrae información de los académicos.

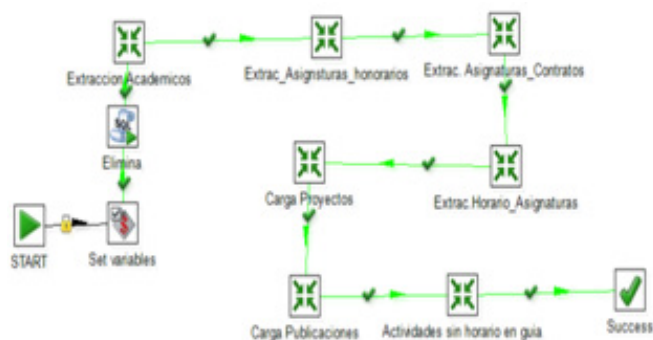


Figura 5 – Flujo de trabajo para la extracción de datos seleccionados

4.7. Etapa Proceso ETL+V: Transformación.

Al diseñar dos DM, se opta por elaborar un flujo de trabajos para cada uno, lo cual permite una depuración más sencilla. En la Figura 6 a) se presenta el flujo de trabajo para el DM que calcula los indicadores de docencia. En la Figura 6 b), se presenta el flujo de trabajo para calcular los indicadores asociados a la actividad de investigación. Cabe señalar, que sólo se presentan los flujos de trabajos resumidos para ambos DM, y

el detalle de cada transformación se deja fuera por limitación de números de páginas para el artículo



Figura 6 – Flujo de trabajo para el cálculo de indicadores de: a) docencia b) investigación

4.8. Etapa Proceso ETL+V: Carga.

La carga de las tablas hechos es la parte más compleja de esta etapa, puesto que se debe realizar la comprobación de los datos cargados o cambio de algún nodo de la transformación. Antes de la carga, se deben verificar si los cálculos obtenidos son válidos en relación a los datos en los sistemas operacionales. Para el DM de docencia, la transformación que presenta la Figura 7, calcula las métricas a nivel mensual, y luego las carga a la tabla hechos de docencia.

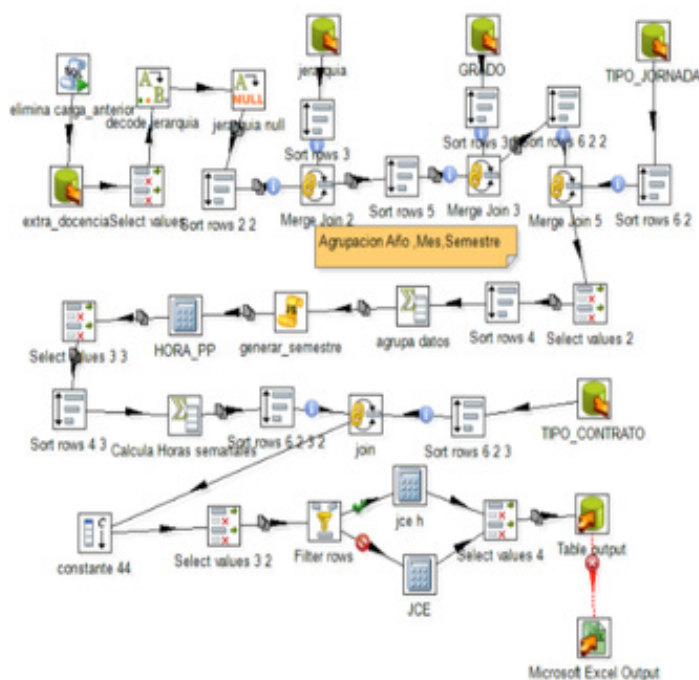


Figura 7 – Flujo de trabajo para la carga de tabla hechos de docencia

Primero se realiza una limpieza de la carga anterior de los datos si es que es necesario, para que en el nodo “extra_docencia” de la Figura 7, ejecute la consulta que extrae los datos de las asignaturas impartidas a una carrera y la cantidad de horas involucradas. En los pasos siguientes se hacen cruces de información para extraer los identificadores únicos de las dimensiones que correspondan.

El nodo agrupa las dimensiones para realizar el cálculo total de horas semanales de la actividad a una carrera en particular. Luego en el siguiente nodo, se suman todas las horas semanales calculadas con el fin de evitar duplicación de información, y no romper las reglas de integridad al momento de cargar los datos. Posterior a este paso, se genera un filtro para separar los docentes según su tipo de jornada, con el fin de poder calcular cuál es su aporte a las JCE de su actividad. Finalmente, se procede a cargar la información a la tabla de hechos de docencia. Algo similar se realiza para el DM de investigación.

4.9.Etapa Proceso ETL+V: Validación de KPI.

Finalmente, se validan que los indicadores obtenidos sean iguales o semejantes a los que se puedan obtener desde los sistemas operacionales. Para esto, se utiliza como método de prueba, calcular cada indicador mediante una consulta SQL a la base de datos desde los sistemas transaccionales, comparándolo con la consulta que se genera con las tablas del MM. En la Figura 8, se muestra la validación de un indicador de un total de 14. En particular, para este ejemplo se puede corroborar que el KPI está bien calculado, ya que su valor corresponde a los datos de origen.

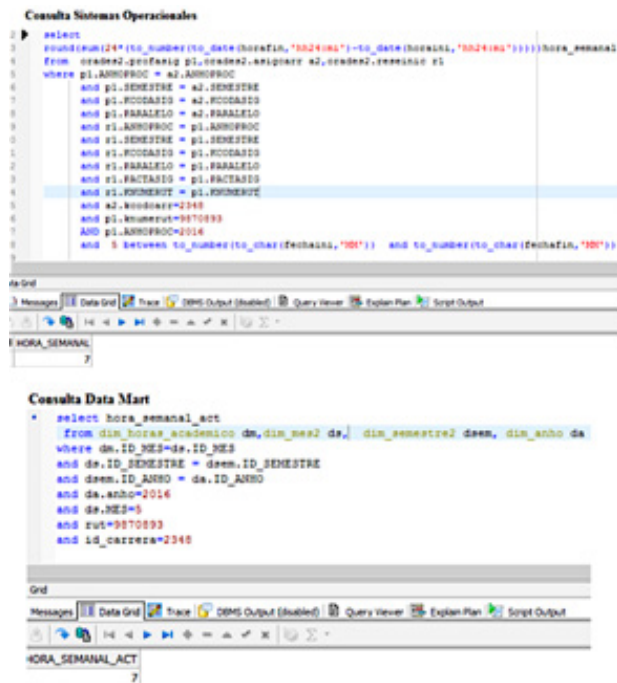


Figura 8 – Código SQL para validación de un KPI

4.10. Etapa Procesamiento Analítico: OLAP.

Se presentan los resultados del DM mediante la explotación y visualización de los indicadores a través de la herramienta OLAP Qlikview. De esta forma se da respuesta a los requerimientos definidos y planteados en un comienzo del proyecto. Estos resultados son expuestos con gráficas, las que permiten visualizar la información y poder realizar el análisis que corresponda. A modo de muestra, se presenta en la Figura 9, una gráfica de un indicador de docencia. En total son 14 gráficas, una por cada KPI.

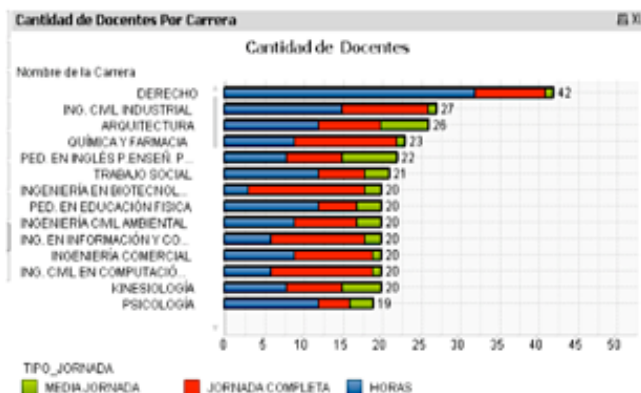


Figura 9 – Gráfica: cantidad de profesores por carrera

4.11. Etapa Procesamiento Analítico: Dashboard.

Una vez generadas las gráficas de los indicadores, se pueden integrar en un formato tipo tablero de control (dashboard), que permite a los usuarios finales, realizar un análisis multidimensional con distintos controles y mecanismos de interacción. El dashboard que muestra la Figura 10, presenta una consolidación de los indicadores relacionados a docencia en la cual, en los extremos tanto izquierdo como derecho, se encuentran las dimensiones manejadas por el modelo, y que pueden ser seleccionadas para cambiar los gráficos presentados al interior del panel. Los datos resaltados en color verde significan las selecciones actuales, y los de color gris que esos valores no están relacionados a la selección actual.

En la Figura 11, se muestran indicadores de proyectos de investigación en un dashboard, que se pueden visualizar por diferentes dimensiones. Los gráficos se encuentran agrupados en un elemento contenedor en la cual se muestra la pestaña sobre la dimensión en la que se muestra el indicador con algún gráfico asociado.

5. Conclusiones

Para el desarrollo de un PDW, toma gran relevancia la comunicación efectiva con las unidades con las que se realizan los trabajos en general. En particular, en el trabajo de educación fue clave la plantilla de especificación de requisitos de los KPI, ya que actuó como interfaz entre los usuarios estratégicos y el equipo del DW.



Figura 10 – Tablero de control de indicadores de docencia



Figura 11 – Tablero de control indicadores de proyectos de investigación

Durante el proceso de desarrollo de un DM, muchas veces se tiene que volver a un paso anterior para corregir o ajustar detalle del proceso en sí, es decir, a prueba y error lo que se ve reflejado mayormente en el proceso ETL+V.

Un factor importante es el análisis y comprensión, que se debe realizar sobre las estructuras de datos y modelos de los sistemas operacionales.

El desarrollo de un repositorio temporal, ayuda a dar respuesta eficiente a requerimientos operacionales durante la elaboración de un DM.

La validación de los KPI fue otro elemento clave en el proceso, ya que permitió generar confianza en el equipo, y las unidades de la institución sobre los resultados obtenidos a través del PDW.

Las visualizaciones gráficas integradas a través de los dashboard fueron muy bien valoradas por parte de los usuarios estratégicos.

Los resultados generaron un gran impacto en los usuarios, ya que en algunos casos los indicadores no se encontraban sobre la media de la institución, por el contrario, existían casos que los resultados presentados eran bastante buenos y aceptables, sin embargo, esta presentación de los resultados, implica que se tome cierta atención a ello, con el fin de apoyar la toma de decisiones desde la perspectiva estratégica y así poder mantener y mejorar los indicadores en general.

Finalmente, la metodología elaborada y utilizada en este PDW, fue muy valorada en general por la institución, así como en el equipo de trabajo.

En cuanto a trabajo futuro, si bien la metodología tiene definidas sus etapas y funciones en cada una de estas, así como tiene determinadas sus entradas y salidas en cada una de sus etapas, le falta establecer una formalización para que pueda responder a un proceso repetible. En esta línea, el trabajo a seguir es documentar detalladamente todos los aspectos y métodos de esta nueva metodología. Así como también aplicarlas en nuevos PDW, y realizar un análisis comparativo de los resultados con y sin la metodología propuesta.

Referencias

- Rocha, Á. (2012). Framework for a Global Quality Evaluation of a Website. *Online Information Review*, 36(3), 374-382. doi:10.1108/14684521211241404
- Antunes, A. A. (2004). Sistemas XYZ. In Sousa A. J. (Ed.), *Tecnologias Internet*. Lisboa: Editora Xxxpto.
- Back, T. (2002). Adaptive business intelligence based on evolution strategies: some application examples of self-adaptive software. *Information Sciences*, 148(1-4), 131-121.
- Golfarelli, M., Rizzi, S., & Cella, I. (2004). Beyond Data Warehousing: What's Next in Business Intelligence. In: *Proceedings of the 7th ACM international workshop on data warehousing and OLAP*, (pp. 1-6). Washington DC, USA: ACM.
- Palocsay, S., Markham, I., & Markham, S. (2010). Utilizing and teaching data tools in Excel for exploratory analysis. *Journal of Business Research*, 63 (2), 191-206.
- Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit - The Complete Guide to Dimensional Modeling* (2nd edition). Hoboken, NJ, USA: Jhon Wiley and Sons, Inc.
- Bernabeu, R. (2010). *HEFESTO: Metodología para la Construcción de un Data Warehouse*. Córdoba, Argentina.

- Conesa, J., & Curto, J. (2010). *Introducción al Bussiness Intelligence* [en línea]. Barcelona: Editorial UOC, ISBN 978-84-9788-886-8.
- Venter, M. (2005). Business Intelligence Initiatives: Failures Versus Success. *Revista Interdisciplinary Journal* [en línea], 4 (1).
- Mazón, J., Trujillo, J., & Lechtenborger, J. (2007). Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. *Data & Knowledge Engineering*, 63(3) 725–751.
- Inmon, W. (2005). *Building the Data Warehouse* (3rd edition). Hoboken, NJ, USA: Jhon Wiley and Sons, Inc.
- Vaisman, A., & Zimányi, E. (2014). Data Warehouse Systems Design and Implementation. *Springer Series: Data-Centric Systems and Applications*, (XXVI), 603.
- Duran, A. (2000). *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*, Tesis doctoral Universidad de Sevilla.
- Marcus, A. (2006). Dashboards in Your Future. *The Art of Prototyping*, 13 (1), 48–60.
- Illescas, G., Sanchez, M., & Canziani, G. (2015). Métodos de Pronóstico por Indicadores dentro de la Gestión del Conocimiento Organizacional. *RISTI - Revista ibérica de Sistemas y Tecnologías de Información*, (E3), 29 – 41. DOI: 10.17013/risti.e3.29-41.
- Prieto, A., & Piattini, M. (2015). Propuesta de marco de mejora continua de gobierno TI en entidades financieras. *RISTI - Revista ibérica de Sistemas y Tecnologías de Información* (15), 51 – 67. DOI: 10.17013/risti.15.51-67.

Anexo

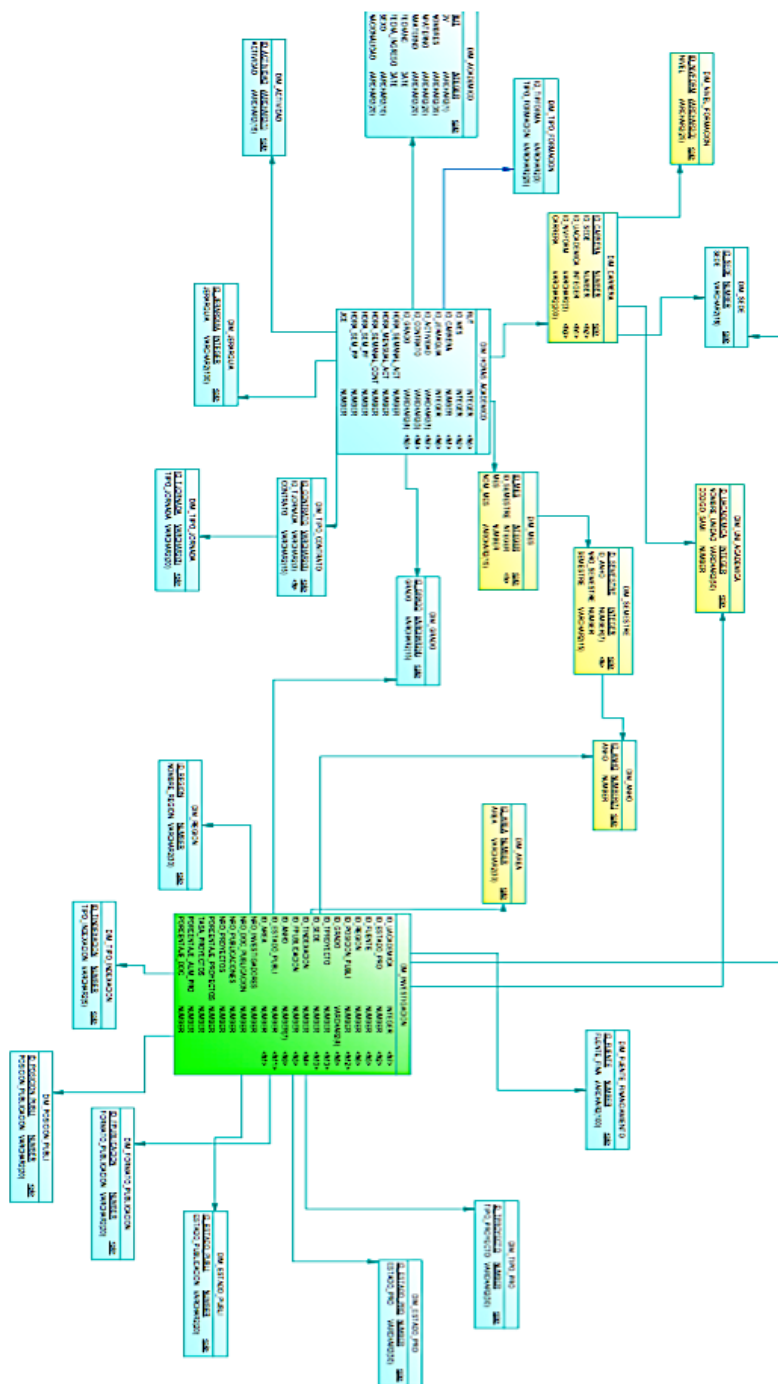


Figure 12 – MM relacional de los DM

Hacia la Interoperabilidad Semántica para el Manejo Inteligente y Sostenible de Territorios de Alta Biodiversidad usando SmartLand-LD

Nelson Piedra¹, Juan Pablo Suárez¹

{nopiedra, jpsuarez}@utpl.edu.ec

¹ Universidad Técnica Particular de Loja, Ecuador

DOI: 10.17013/risti.26.104-121

Resumen: Un reto presente en los actuales esfuerzos de desarrollo sostenible es potenciar la interoperabilidad entre variables e indicadores almacenados en sistemas de información y proveedores de servicios de datos autónomos, distribuidos y heterogéneos. El acceso restringido a fuentes de datos y servicios limita las capacidades de un territorio inteligente y sostenible para mejorar los aspectos sociales, ambientales y económicos. Este trabajo presenta un enfoque de Web Semántica que apoya la extracción, la transformación de datos vinculados, la integración y la explotación de datos masivos y heterogéneos, con aplicación en la toma de decisiones relacionadas con el desarrollo inteligente y sostenible de los territorios. Motivado por el carácter no escalable de los métodos de gestión de datos sostenibles convencionales, así como por los desafíos de interoperabilidad presentes en los silos de datos contemporáneos, el marco descrito, acuñado como SmartLand-LD, aprovecha el potencial combinado de diversas fuentes de datos. Estos incluyen transmisiones de sensores, datos geográficos, bases de datos científicas, recursos relacionados disponibles en repositorios de conocimiento y otras fuentes de datos abiertos estructurados y no estructurados. A través de la apertura, la construcción de gráficos de conocimiento, la interoperabilidad semántica y las técnicas colaborativas de ciencia abierta, el marco propuesto permite el mapeo e integración de datos sobre indicadores de desarrollo sostenible en territorios de alta biodiversidad..

Palabras-clave: Web Semántica, Datos Enlazados, Datos Abiertos, Interoperabilidad, Grafos de Conocimiento.

Towards Semantic Interoperability for Smart and Sustainable Management of High Biodiversity Territories using SmartLand-LD

Abstract: A present challenge in today's sustainable development efforts is to empower interoperability between variables and indicators across autonomous, distributed and heterogeneous information systems, and data service providers. Restricted access to data sources and services limits the capabilities of a smart and sustainable territory to improve social, environmental and economic aspects. This work presents a Semantic Web approach that supports the extraction, linked data transformation, integration, and exploitation of large-scale and

heterogeneous territories data, with application to decision-making related with smart and sustainable development of territories. Motivated by the non-scalable character of conventional sustainable data management methods, as well as by the interoperability challenges present in contemporary data silos, the described framework— coined as SmartLand-LD – leverages the combined potential of diverse data sources. These include sensor streams, geo-data, scientific databases, related resources available from knowledge repositories, and other structured and non-structured open data sources. Through openness, knowledge graphs building, semantic interoperability, and open science collaborative techniques, the framework proposed enables the mapping and integration of data about sustainable development indicators in high biodiversity territories.

Keywords: Semantic Web, Linked Data, Open Data, Interoperability, Knowledge Graph.

1. Introducción

La convergencia de las tecnologías de la información y la comunicación están produciendo cambios en la gestión de los entornos urbanos y rurales. Estos territorios son diferentes de cualquier otra realidad que se haya experimentado hasta el día de hoy. Durante gran parte del siglo XX, la idea de que un territorio podría considerarse inteligente era simplemente una cuestión de ciencia ficción. Sin embargo, la expectativa de que una ciudad o territorio se gestione de forma inteligente y en armonía con el ambiente, convierte esta idea en una nueva realidad impulsada desde diferentes visiones y perspectivas.

Las diferentes visiones de desarrollo y sostenibilidad se pueden consensuar a través de acuerdos y colaboración que permitan avanzar hacia la creación y consolidación de ciudades y territorios, inteligentes y sostenibles. En cuanto al monitoreo y evaluación de los esfuerzos de desarrollo sostenible, se requiere una gran cantidad de indicadores de calidad que midan la sostenibilidad ambiental, así como el impacto en aspectos sociales y económicos (Cutter, Osborn, Romano, & Ullah, 2015; Hák, Janoušková, & Moldan, 2016) world leaders embark on the notion of universe global Sustainable Development Goals (SDGs. Por lo general, estos datos e indicadores están depositados en diferentes sistemas de información. En este sentido, un reto presente en las acciones de desarrollo sostenible está en mejorar la medición de los avances o retrocesos de la sostenibilidad y alcanzar la interoperabilidad con la ayuda de tecnologías que conecten los diferentes silos de información.

Crear un ecosistema para la integración e interoperabilidad de los datos es una tarea compleja. De esta forma,

En este trabajo, los autores presentan un enfoque para el manejo inteligente y sostenible de territorios de alta biodiversidad: la Iniciativa SmartLand. SmartLand es sobre la gestión inteligente de las interacciones entre personas y territorios. SmartLand dispone de un marco de trabajo, SmartLand-LD (descrito como trabajo en progreso en (Nelson Piedra & Suárez, 2017) we investigate the core challenges faced when consuming multiple data sources for smart management of territories of high biodiversity using the Linked Data approach. We design a framework to achieve better data interoperability and integration by republishing real-world data into linked data: the SmartLand-LD framework. SmartLand-LD will work as a flexible and distributed ecosystem of autonomous and

heterogeneous data sources (in content, technology, storage and structure), basado en el enfoque de Web Semántica (Berners-Lee, Hendler, & Lassila, 2001) y las tecnologías de Datos Vinculados (en inglés: *Linked Data*, LD) que habilitan la capacidad de construir un entorno global de información (Heath & Bizer, 2011) alineado con el desarrollo sostenible. Este enfoque reduce la duplicación de esfuerzos al describir los recursos de información y sus atributos, y fomenta la creación de un grafo de conocimiento (en inglés: *Knowledge Graph*, KG) global con capacidad de abarcar de forma masiva la información necesaria para realizar consultas y acciones complejas (W3C, 2011). La primera sección describe de forma general la gestión inteligente de ciudades y territorios; a continuación se presenta la organización de la iniciativa SmartLand. Luego se discute el poder de la Web Semántica (SW) y los Datos Enlazados (LD) como una forma de alcanzar la interoperabilidad semántica de datos y desarrollar la Web de Conocimiento. A continuación se describe el marco propuesto para la publicación de datos enlazados de algunos conjuntos de datos existentes y una aproximación sobre cómo definir y reutilizar recursos ontológicos para representar KG. Finalmente, la última sección cubre las conclusiones respectivas y áreas futuras de investigación.

2. Gestión inteligente de territorios - SmartLand

2.1. La gestión inteligente de las zonas urbanas

En las últimas dos décadas, el concepto de ciudad inteligente (Albino, Berardi, & Dangelico, 2015) se ha vuelto cada vez más popular en la literatura científica y las políticas internacionales. La razón principal parece estar relacionada con el papel principal de las ciudades y los territorios en los aspectos sociales y económicos de las personas en todo el mundo, y en el gran impacto sobre la sostenibilidad ambiental (Mori & Christodoulou, 2012).

Las ciudades son el núcleo del consumo de recursos globales. El número y la proporción de la población urbana crecerán en el futuro. Se estima que las ciudades globales ocupan aproximadamente el 2% de la superficie terrestre del planeta, consumen entre 60-80% de la energía y son responsables del 75% de las emisiones de carbono. Además, muchas de las decisiones llevadas a cabo por los habitantes de las ciudades afectan directamente la biodiversidad y los ecosistemas. A la luz de esto, la tendencia actual es crear ciudades inteligentes (o *Smart Cities*).

La idea de ciudades inteligentes se refiere a la promoción de ciudades más inteligentes, sostenibles e inclusivas a través de la promoción de innovaciones tecnológicas apropiadas (Hollands, 2008). Las ciudades y los territorios se consideran el elemento clave en los planes estratégicos de los países (Nam & Pardo, 2011) (Deakin & Waer, 2011) (Dirks, Gurdgiev, & Keeling, 2010). El ciudadano debe estar en el centro de atención. Sin embargo, la ciencia y la ingeniería desempeñan un papel crucial en la innovación y la creación de nuevos servicios públicos, así como en el desarrollo del concepto de gestión inteligente y sostenible (Ahvenniemi, Huovila, Pinto-Seppä, & Airaksinen, 2017).

2.2. La gestión inteligente de los territorios

Los territorios de alta biodiversidad no existen como una esfera separada de las acciones, ambiciones y necesidades humanas. Territorio y desarrollo son inseparables (Kates,

Parris, & Leiserowitz, 2005), y su gestión plantea un desafío contemporáneo de alcance global (Acosta, Piedra, & Suárez, 2017). El Convenio sobre la Diversidad Biológica establece tres vastos y ambiciosos objetivos: la diversidad biológica de conservación, el uso sostenible de sus componentes, y la creación de propuestas que busquen la distribución justa y equitativa de los beneficios derivado del uso de los recursos del territorio (Balmford et al., 2005). Por otro lado, desde el 2016, arrancó la Agenda 2030 y sus Objetivos de Desarrollo Sostenible, ODS, asumidos por 193 naciones comprometidas con resolver los problemas sociales, económicos y ambientales que aquejan al mundo (United Nations General Assembly, 2015), marcarán la agenda de desarrollo mundial durante los próximos.

Los ODS son un conjunto de 17 objetivos y 169 metas que reflejan un balance entre las dimensiones económicas, sociales y ambientales (Cutter et al., 2015) world leaders embark on the notion of universe global Sustainable Development Goals (SDGs, destinadas a fortalecer la paz, erradicar la pobreza en todas sus formas, combatir las desigualdades, construir sociedades pacíficas, justas e inclusivas, proteger los derechos humanos, promover la igualdad entre los géneros y el empoderamiento de las mujeres y las niñas, así como garantizar una protección duradera del planeta y sus recursos naturales ((UNSG), 2016).

3. SmartLand, una visión para el desarrollo inteligente y sostenible de territorios megadiversos

3.1. Contexto

Sobre un territorio de alta biodiversidad hay muchas partes interesadas, cada una con diferentes visiones, intereses, y opiniones variadas sobre objetivos, políticas, prioridades de inversión, y esfuerzos por el desarrollo y la sostenibilidad. Las partes interesadas son instituciones educativas, organizaciones gubernamentales, inversores, agencias reguladoras y grupos de la sociedad civil que llevan a cabo diferentes acciones que afectan el desarrollo sostenible. En particular, esta maleabilidad inherente del desarrollo sostenible, plantea el reto de mejorar las estrategias de articulación de mediciones sobre el desarrollo sostenible.

A la luz de los desafíos por lograr un mejor equilibrio entre las dimensiones económica, social y ambiental, y como parte de su responsabilidad de distribución social del Conocimiento (Acosta et al., 2017), la Universidad Técnica Particular de Loja (UTPL) promueve la iniciativa institucional SmartLand para la gestión sostenible e inteligente de los territorios de alta biodiversidad (ver <http://smartland.utpl.edu.ec>). SmartLand promueve la visión de un uso más sostenible e “inteligente” de la biodiversidad, y la distribución justa y equitativa de los beneficios derivados del uso de recursos del territorio. SmartLand impulsa la construcción de una visión consensuada sobre territorio y desarrollo; promueve alianzas para el desarrollo sostenible, la participación activa, el uso de conocimiento científico, e innovaciones tecnológicas para una gestión inteligente de la interacción entre los seres humanos y los territorios biodiversos. De acuerdo a la visión de SmartLand, los esfuerzos por el desarrollo inteligente de los territorios deben abarcar indicadores de impacto que midan la sostenibilidad ambiental, económica y social.

3.2. Apertura, Colaboración, Datos y Desarrollo sostenible

En la misión de SmartLand está medir el progreso relativo o la falta de progreso hacia el desarrollo sostenible. En este sentido, un tema crucial en el desarrollo de ciudades y territorios inteligentes y sustentables es el uso de estándares y enfoques para garantizar la interoperabilidad, de modo que los sistemas producidos por diferentes proveedores trabajen juntos sin interrupciones, y que brinden la capacidad reducir costos a través de economías de escala.

La apertura y colaboración son fundamentales para conectar recursos de información autónomos relacionados con la iniciativa SmartLand. Sin capacidad de interoperabilidad, los esfuerzos de monitoreo de indicadores a menudo toma la forma de largas y heterogéneas listas de indicadores, muchos de los cuales se enfocan en el presente o en el muy corto plazo. Además, las listas de indicadores resultantes generan silos de información (el concepto de silo está explicado en (Ensor, 1988; Piedra Pullaguari, 2016)), que reflejan las diversas aspiraciones del conjunto diverso de partes interesadas y demandan la necesidad de integrar los esfuerzos aislados de medición.

La perspectiva de medición de SmartLand es la de monitoreo permanente y de interoperar silos de información. SmartLand trabaja en la creación de indicadores que faciliten la planificación a través de múltiples escalas de tiempo, aumentando la posibilidad de hacer que los territorios sean más inteligentes a largo plazo, mediante el monitoreo de variables que describan el territorio. Con el fin de mejorar la comprensión de los indicadores, la iniciativa SmartLand promueve estrategias y herramientas tecnológicas que son necesarias para la recopilación, interoperabilidad, integración, análisis de datos, visualización y preservación de grandes colecciones de datos e información. Se busca alcanzar la interoperabilidad y el reuso de variables e indicadores, así como el compromiso de monitoreo a largo plazo, necesario para reflejar el horizonte temporal del ahora y del futuro en relación al desarrollo sostenible.

3.3. Organización de la Iniciativa SmartLand

SmartLand se ejecuta a través de programas/proyectos de investigación multidisciplinarios, organizados en 12 paquetes de trabajo, 4 objetivos comunes, y una plataforma tecnológica para describir e intercambiar datos. El pilotaje de la iniciativa se ejecuta en tres provincias del sur de Ecuador, que abarcan los ecosistemas de la Costa, los Andes y la Amazonía montañosa. Esta región es la más pequeña del territorio de Ecuador, pero también la más biodiversa. Se aplica un enfoque de innovación social para vincular a beneficiarios, docentes investigadores y estudiantes en mejorar el proceso de toma de decisiones públicas y privadas y mejorar la gestión de desarrollo sostenible.

Los paquetes de trabajo (WP, *Working Packages*) ayudan a contribuir al cumplimiento exitoso de los objetivos centrales de la iniciativa a través de proyectos de investigación propuestos por investigadores de diferentes dominios de conocimiento (Ver figura 1-a)). Cada paquete de trabajo está compuesto por proyectos de investigación que mejoran la comprensión científica del medio ambiente y agregan valor a los indicadores sociales, biológicos, ambientales y de infraestructura.

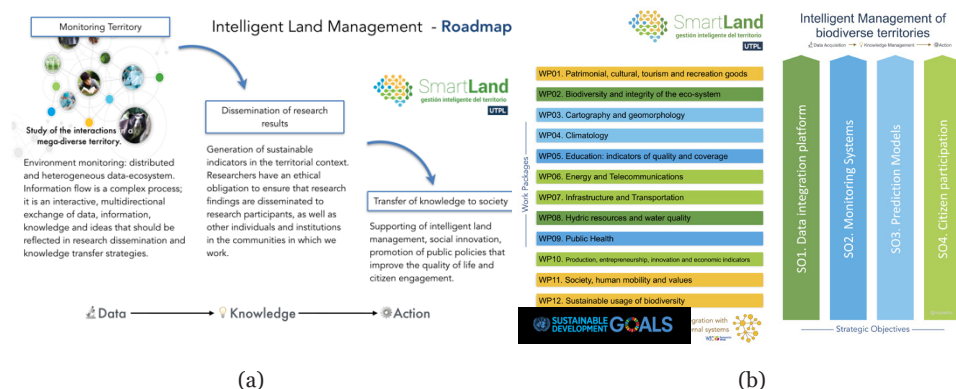


Figura 1 – Organización general de la iniciativa SmartLand. (a) *Roadmap* sobre la gestión del conocimiento. (b) Paquetes de trabajo y objetivos estratégicos en SmartLand

Los paquetes de trabajo comparten transversalmente cuatro objetivos estratégicos (SO, *Strategic Objectives*) para la gestión de la información, que se reflejan en la disponibilidad de una arquitectura de datos web abierta que abarca la recopilación, preservación de datos, reutilización, análisis, modelado predictivo y la visualización de datos (Ver figura 1-b). Cada grupo de investigación trabaja en sus propias preguntas de investigación, mientras comparte cuatro Objetivos Estratégicos y busca contribuir a los ODS.

SO-1. Consolidar un sistema para la preservación del conocimiento que garantice el acceso libre. SmartLand promueve la creación de una infraestructura tecnológica que, desde un enfoque distribuido, ayuda a recopilar, monitorear y reutilizar datos e información de alta heterogeneidad en diversos campos del conocimiento: biodiversidad, ecosistemas, economía, territorio, emprendimiento, sociedad, valores e identidad, entre otros.

SO-2. Mejorar nuestra comprensión de los indicadores y su monitoreo permanente. Se promueve una estrategia de recolección sostenible de datos, que se basa en redes de sensores y sistemas de monitoreo de indicadores a largo plazo.

SO-3. Mejorar la gestión de los territorios a través del uso eficiente de los recursos, la preservación de su patrimonio y el uso sostenible. SmartLand busca explotar, modelar, monitorear y visualizar diversas variables, así como desarrollar escenarios probabilísticos, ordenanzas territoriales virtuales y mejorar el análisis de la toma de decisiones.

SO-4. Fortalecer las capacidades y potencialidades de los ciudadanos. Se promueve la participación responsable de la ciudadanía mediante el fortalecimiento de la identidad y el valor de los seres humanos con el medio ambiente.

Aspectos destacados de la iniciativa. Se proponen acciones clave en cada WP. De esta forma, el iniciativa complementa las orientaciones más específicas que se espera que desarrollen los actores locales, cada una de las cuales refleja las necesidades y prioridades únicas de sus electores y al mismo tiempo comparte la visión y los objetivos comunes

descritos aquí. La iniciativa se desarrolla desde el 2014. SmartLand ejecutó 108 proyectos con aproximadamente 300 investigadores involucrados (84 internacionales) y 2000 estudiantes. Se han implementado herramientas de medición que pueden ser utilizadas como apoyo en la toma de decisiones relacionadas con el desarrollo sostenible, ya que proporcionan información sobre el progreso hacia los objetivos definidos. SmartLand genera de indicadores a través de sistemas de monitoreo consolidados y diferentes observatorios sobre ecosistemas, conflictos socio-ambientales, turismo, movilidad, comunicación estratégica, empresas familiares, y territorio y economía.

4. Interoperabilidad de Datos y Web Semántica

4.1. Desafío: un ecosistema para la interoperabilidad semántica

La interoperabilidad implica llegar a acuerdos, compartir estrategias y tecnologías que integren e interoperen todos los elementos de información que conforman el alcance de la iniciativa, y que de otra manera actuarían como silos de información. La arquitectura semántica de SmartLand debe garantizar la interoperabilidad de diferentes tipos de datos y sistemas (ver Figura 2).

En consecuencia, se deben abordar los siguientes desafíos: (i) heterogeneidad en la representación de datos y semántica, que da como resultado una variedad de modelos de datos y formatos (ii) La heterogeneidad de los sistemas que gestionan los datos y que admiten varios modos de acceso y consulta. (iii) El desafío de procesar los grandes volúmenes de datos y la solución de muchos problemas en la sociedad, los negocios y la academia; (iv) La pérdida del contexto interpretativo o la distorsión del significado de los datos cuando influyen en otros límites semánticos en los que los significados, la terminología y el vocabulario son diferentes: (v) Veracidad, diversidad de escalas, privacidad, mal uso y / o abuso de material publicado, entre otros.

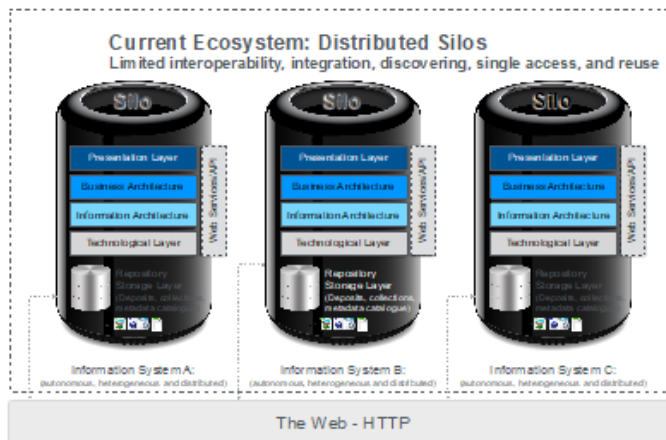


Figura 2 – El desafío de SmartLand: Conectar silos de datos, mejorar el reuso y la integración de información que apoye la construcción de indicadores sobre desarrollo Sostenible

4.2. Aportar datos a un Espacio Global de Información

La Web Semántica (*Semantic Web*, SW) se concibe como la evolución de la Web actual hacia un espacio de información global (Heath & Bizer, 2011), autónomo, y descentralizado. La SW, liderada desde el World Wide Web Consortium (W3C), potencia las posibilidades de compartir a través de la Web datos y significados, en formatos procesables directamente por las máquinas, y alcanzar una mayor interoperabilidad e integración, y con un mínimo de costos técnicos.

Según Tim Berners-Lee, el inventor de la web y el fundador del proyecto LOD, la SW no solo busca publicar datos en la web, sino que también busca crear vínculos entre los datos, lo que ayudaría a explorar el espacio global de información, tanto para usuarios humanos, como por usuarios máquinas (Berners-Lee et al., 2001). Con los datos de enlace, cuando consultamos un recurso, podemos acceder a otros recursos de información publicados en la Web (Heath & Bizer, 2011).

Esta visión presenta cinco desafíos principales: (1) la identificación global de recursos (W3C, 2008); (2) el uso de ontologías formales como una forma de especificar acuerdos específicos de contenido para compartir y reutilizar el conocimiento entre seres humanos y entidades de software (Gruber, 1995; W3C OWL Working Group, 2012); (3) el modelado distribuido del mundo con un modelo de datos compartidos (Cyganiak, Wood, & Lanthaler, 2014; Harris & Seaborne, 2013; W3C, 2011); (4) la infraestructura donde los datos, esquemas y ontologías formales se pueden publicar, compartir, encontrar y reutilizar para apoyar las actividades de intercambio de conocimientos; y, (5) calidad y *provenance* de la información (N. Piedra, Chicaiza, Lopez, & Caro, 2017). Los Datos Vinculados brindan la oportunidad para que los editores contextualicen sus datos, agregando riqueza y profundidad. La aplicación de Linked Data está documentada en diversos trabajos, ver (Guillén, Lloret, & Gutiérrez, 2016; Hallo, Luján-Mora, & Morga, 2017; N Piedra et al., 2015).

En (Berners-Lee, 2006) se plantean 4 principios de diseño para enlazar datos por medio de la Web: (1) Usar URIs (Identificadores de Recursos Uniformes) como nombres para las cosas; (2) Usar URIs HTTP para que las personas puedan buscar esos nombres; (3) Cuando alguien busca un URI, proporcionar información útil, usando los estándares RDF* (Cyganiak et al., 2014) y SPARQL (Harris & Seaborne, 2013)); (4) Incluir enlaces a otros URIs para descubrir más cosas; además, desde nuestra visión, se incluye un punto adicional: (5) el uso de ontologías (W3C OWL Working Group, 2012) para desarrollar la SW.

5. Framework para integración semántica en SmartLand

5.1. Niveles de Interoperabilidad en SmartLand-LD

En SmartLand-LD se abordan diferentes aspectos de coordinación, producción y uso de datos para el desarrollo sostenible. Esta sección presenta la arquitectura del marco semántico SmartLand que se ha diseñado en base a los requisitos definidos en las secciones previas. El concepto subyacente de SmartLand es de una constelación de componentes de varias escalas y tipos (tanto digitales como analógicos), que están

conectados por medio de múltiples redes, y que proporcionan datos continuos con respecto a los factores ambientales, biológicos, sociales y económicos.

El Framework SmartLand-LD implementa cuatro niveles de apertura e interoperabilidad, basado en (Piedra Pullaguari, 2016). La armonización de estos niveles permite alcanzar los objetivos de interoperabilidad semántica pretendida, la misión y apoyar el logro de la visión de futuro planteada en el contexto de este trabajo. A través de estos niveles de interoperabilidad, y los desafíos subyacentes, se puede convertir el potencial semántico de los datos abiertos en una realidad que amplía el descubrimiento y reuso de variables e indicadores relacionados con el desarrollo inteligente y sostenible. Estos niveles se integran de manera que es posible alcanzar objetivos más complejos.

En la Figura 3 se describen los cuatro niveles de interoperabilidad del marco de trabajo de SmartLand-LD. El **nivel de interoperabilidad legal** se refiere a habilitar globalmente el máximo grado de acceso, descubrimiento, uso, y reuso de datos, variables, indicadores y otros contenidos digitales; sino existe una legislación en este contexto, se debe motivar su creación. Se fundamenta en la legislación internacional, regional y nacional relacionada con el movimiento de acceso abierto a los datos y el acceso abierto al conocimiento.

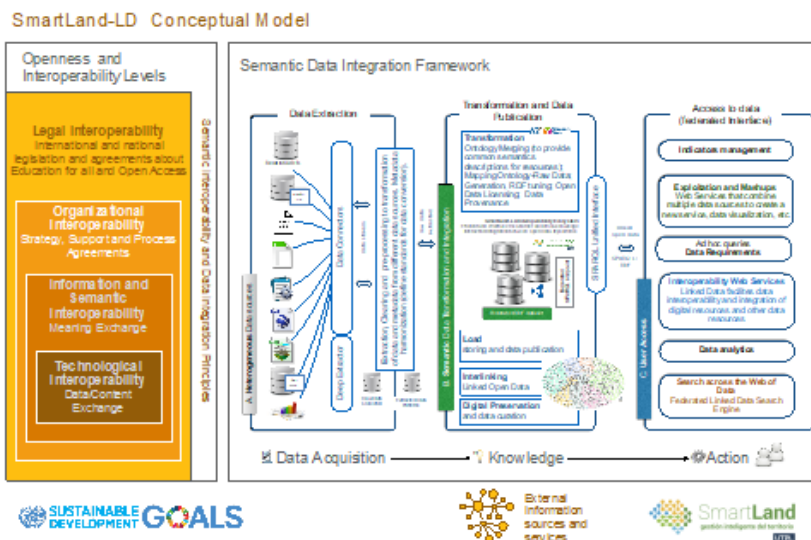


Figura 3 – Framework para la generación de grafos de conocimiento en SmartLand.

El siguiente **nivel de interoperabilidad es organizacional**. Este nivel trata sobre mejorar la eficiencia y la coordinación de esfuerzos, tanto al interior de los proyectos e iniciativa SmartLand, como con otros actores claves relacionados con el Desarrollo Sostenible. Este nivel es clave porque minimiza la sobrecarga (no razonable) de descubrir, usar y re-usar datos abiertos. Es importante establecer la visión de apertura

e interoperabilidad, y el soporte requeridos. Los acuerdos de interoperabilidad pueden, por ejemplo, tomar la forma de memorandos de entendimiento, o niveles de acuerdo de servicios que especifiquen las obligaciones de cada parte participante en los procesos de apertura e interoperabilidad; así como los niveles de servicio esperados, los procedimientos de soporte/escalamiento, datos de contacto, información de procedencia de datos (*provenance*), niveles de calidad de la información, haciendo referencia, cuando sea necesario, a los acuerdos subyacentes en los niveles semánticos y técnicos. Sobre este nivel se implementan **el nivel de interoperabilidad semántica** que habilita el intercambio de datos y significados, se cubren módulos diversos de extracción, desambiguación, modelado semántico, transformación de RDF, y publicación de información en grafos de conocimiento abierto. A **nivel de interoperabilidad tecnológica**, se cubre la funcionalidad que permite el intercambio de datos y contenidos a través de la Web. La interoperabilidad es implementada a través de diferentes componentes de extracción de datos, transformación de datos, publicación de *Linked Data*, y la capa de aplicaciones de usuarios basadas en datos; por ejemplo, un mecanismo de búsqueda federada de datos, un sistema de recomendación de indicadores para apoyar la toma de decisiones informada, aplicaciones de visualización de datos y geográficas, entre otras. A continuación se describen varios componentes tecnológicos del marco de trabajo:

5.2. Recolector de Datos

La información contenida en las colecciones de datos relacionadas con desarrollo sostenible es vital para la toma de decisiones, por lo que almacenar, indexar y recuperar información específica de estas colecciones es clave (Vázquez-Reyes, de León-Sigg, Velasco-Elizondo, Villa-Cisneros, & Briceño-Muro, 2017). El recopilador de datos es un componente central en las tareas de recolección, almacenamiento y preservación de diferentes conjuntos de datos de la iniciativa SmartLand. Los datos recopilados son preservados en el repositorio de datos abiertos Ambar (ver <http://ambar.utpl.edu.ec>). Ambar está basado en CKAN (CKAN, n.d.) que proporciona herramientas colaborativas y abiertas para agilizar la publicación, el intercambio, la búsqueda y el uso de datos, y establecer diferentes períodos de retención de datos. Es posible importar datos de una variedad de fuentes primarias, y no está limitado a datos estructurados. Ambar es extensible a través de su API. Cada vez que se detectan datos nuevos, el componente automáticamente realiza tareas de transferencia de datos hacia al proceso ETL. Los datos recolectados luego son integrados a SmartLand-LD.

5.3. Extracción de Datos

El Proceso de Extracción de datos es un aspecto muy importante de SmartLand-LD. La extracción de datos es un proceso que implica la recuperación de todos los formatos y tipos de patrones de datos. Para cumplir con este requisito, se han desarrollado varias herramientas para manejar este proceso. El proceso de extracción de datos en general se realiza a través del módulo del conector de datos, con funcionalidad personalizada para cada tipo y naturaleza de las fuentes de datos. La fase de extracción implica la recuperación (extracción) de datos brutos de una fuente de datos dada utilizando el complemento apropiado, según la tecnología de la fuente. Los patrones de datos se importan en un

sistema de extracción intermedio antes de procesarse para la transformación de datos, donde posiblemente se rellenen con metadatos antes de exportarse a otra etapa en el flujo de trabajo de transformación y publicación de LD. Los datos extraídos se agregan a una base de datos relacional, cuya flexibilidad se garantiza con esquemas en forma de triples. La extracción de datos se ejecuta constantemente o en un cronograma definido por el usuario. Como el componente necesita hacer otros procesos y no solo extraerlos, se ha implementado funcionalidad que verifica constantemente nuevas actualizaciones de datos. Estos datos se usarán posteriormente para la transformación semántica.

5.4. Identificación de Recursos Global

Con el intercambio creciente de metadatos, diferentes sistemas de identificación colisionarán en un ecosistema global de datos. En SmartLand-LD, cada identificador es un URI cool y persistente (N. Piedra et al., 2017; W3C, 2008). Un URI es una cadena de caracteres utilizada para identificar el nombre de un recurso. Dicha identificación permite la interacción con representaciones del recurso a través de una red, típicamente la Web, usando protocolos específicos. Los esquemas que definen cada URI que especifican una sintaxis concreta y los protocolos asociados (Gregorio & et. al, 2012). La resolución es el mecanismo clave que permite a un sistema localizar y acceder al objeto identificado o a la información relacionada con este en la Web. Resolver un URI significa convertir una referencia de URI relativa a forma absoluta, o desreferenciar una referencia de URI o URI intentando obtener una representación del recurso que identifica. El componente ‘resolver’ en el software de procesamiento de documentos generalmente proporciona ambos servicios. Cualquier tipo de identificador persistente adicional es útil para enlazar y obtener más información contextual

5.5. Ontologías base

Un ontología es una conceptualización formal, sistematizada, explícita y consensuada de un dominio de trabajo. Una conceptualización es una visión simplificada y abstracta del mundo que deseamos representar con algún propósito. Cada base de conocimiento, sistema basado en conocimiento o agente de nivel de conocimiento está comprometido con alguna conceptualización, explícita o implícitamente (Gruber, 1995). Las ontologías centrales (*Core-Ontologies*) en SmartLand, son modelos de datos, explícitos, simplificado, consensuados, reutilizables y extensibles que capturan las características fundamentales de conocimiento (propiedades, relaciones y restricciones), en lo posible de una manera neutral al contexto. Estos vocabularios principales se usan en acuerdos de interoperabilidad que permiten que los sistemas de información, desarrollados de forma independiente, intercambien información, lo que hace posible que la información atraviese fronteras y dominios. Están diseñados para convertirse en la base de nuevas ontologías de dominio que se utilizarán para intercambiar información. También se usan como vocabularios base en procesos de mapeo y alineamiento con otros vocabularios. Los vocabularios centrales utilizados en SmartLand-LD son:

DCAT, esta especificación recomendada por el W3C facilita la interoperabilidad entre los catálogos de datos. Su caso de uso básico es permitir la búsqueda de información en el portal de datos SmartLand. Además, se mejora la publicación descentralizada

de catálogos, la visibilidad y se permite que las aplicaciones consuman fácilmente los metadatos de múltiples catálogos.

SDMX, se usa en procesos de intercambio de metadatos y datos estadísticos (en inglés: *The Statistical Data and Metadata eXchange*) entre organizaciones internacionales y miembros individuales.

RDF Data Cube, se usa como base central que admite la extensión de vocabularios que describan diferentes aspectos de flujos de datos estadísticos u otros conjuntos de datos multidimensionales. El modelo que sustenta el vocabulario Data Cube es compatible con el modelo de cubo que subyace a SDMX.

DCMI Metadata Terms, es una especificación autorizada de todos los términos de metadatos mantenidos por Dublin Core Metadata Initiative (DCMI), que proporciona elementos, refinamientos de elementos, esquemas de codificación y términos de vocabulario, una especificación.

Geodata, define cómo describir la información geográfica, además desde una perspectiva global se usarán los estándares ISO 19115: 2003, ISO 19139 y 19115: 20141.

FOAF, permite describir redes de colaboración, a través de tres tipos de red: redes sociales de colaboración humana, amistad y asociación; redes representacionales en términos fácticos, y redes de información que usan enlaces basados en la Web para compartir descripciones publicadas de forma independiente.

Void, permite expresar los conjunto de datos RDF. Además, Void es vital en acciones de descubrimiento y catalogación de datos en la web.

PROV, proporciona una base para representar la información de procedencia sobre las entidades, actividades y actores involucrados en la producción de los elementos de datos.

5.6. Flujo de Publicación de Datos Abiertos y Enlazados

El objetivo de este componente la publicación de los metadatos extraídos como *Linked Data*. El primer paso es diseñar y asignar URIs persistentes a los datos y describirlos en RDF utilizando vocabularios consensuados. En el caso de los datos estructurados (por ejemplo, almacenados en bases de datos relacionales, archivos CSV, Excel o XML), la funcionalidad de transformación de RDF está basada en RDF wrappers y APIs. Mientras que, para los datos no estructurados, se han empleado técnicas de minería de textos que actúan como extractores de entidades, claves para descubrir las entidades de datos que serán descritas como Linked Data. El proceso de transformación SmartLand-LD consiste en los siguientes pasos:

Pre-procesamiento de datos - desambiguación y limpieza de datos. Este componente aporta a la calidad de los contenidos que se incorporan al flujo de datos del framework. Se realizan tareas de limpieza, remplazamientos, y desambiguación de datos de acuerdo con diferentes métodos de resolución de conflictos (Nelson Piedra, Chicaiza, Lopez-Vargas, & Caro, 2016).

Modelado de datos. El ciclo de vida para el desarrollo de nuevos recursos ontológicos se describe en la figura 4.



Figura 4 – Ciclo de vida para la construcción de recursos ontológicos en SmartLand.

Conversión de datos extraídos a RDF. Los mapeos de vocabulario se expresan usando un lenguaje de mapeo interno. El lenguaje proporciona transformaciones simples, así como transformaciones estructurales más complejas (1-a-n y n-a-1) y transformaciones de valores de propiedades, se normalizan diferentes unidades de medida, y es posible ejecutar manipulaciones complejas de datos.

Especificar licencias. De acuerdo a las características de cada conjunto de datos se define una licencia del grafo RDF. Ejemplo de licencia CC BY-SA.

Enlazar datos a otros conjuntos de datos. Los grafos de conocimiento descritos en RDF, se vinculan con otros datos, a fin de proporcionar contexto y enriquecer su significado. Con respecto al proceso de vinculación, se dispone de un módulo de enlazado y validación desarrollado a medida (se usó como base Silk Link Discovery Framework). Entre las fuentes externas están: datos del Banco Mundial (<http://data.worldbank.org>), GeoNames (<http://geoname.org>), DBpedia (<http://dbpedia.org>), y DBpedia Latinoamérica (<http://es-la.dbpedia.org>). Para especificar la condición que debe ser cierta para que dos entidades se consideren duplicadas, el usuario puede aplicar diferentes métricas de similitud, como métodos de comparación de cadena, fecha o URI, a valores de propiedades múltiples de una entidad o entidades relacionadas. El proceso de resolución de identidad descubre los alias de URI en los datos de entrada y los reemplaza con un único URI basado en la heurística de coincidencia proporcionada por el usuario. Además, se agregan enlaces owl:sameAs que apuntan a los URI originales, lo que hace posible que las aplicaciones vuelvan a referirse a las fuentes de datos originales en la Web.

Publicación. Una vez que los datos extraídos se han transformado en RDF, estos datos son almacenados y publicados en un triplestore. En nuestro escenario, decidimos que los datos convertidos, se describen con VoID, se anota la procedencia con PROV y se cargan en el Virtuoso, edición abierta. VoID.. Publicado el LD, cualquier sistema externo es capaz de ejecutar consultas semánticas (utilizando el lenguaje de consulta

SPARQL) para extraer y reutilizar la información que considere oportuno. Los avances hacia compartir indicadores en la forma de datos abiertos enlazados ofrecen una serie de ventajas: habilitar la interoperabilidad semántica, incrementa la integración y el reuso de datos con un pequeño impacto en los sistemas de información actuales; apoya la creatividad y la innovación a través de la participación activa de los ciudadanos. A través de una URI, las máquinas y los usuarios humanos deben poder recuperar descripciones semántica de los recursos de conocimiento. Los usuarios máquina obtienen datos RDF y los humanos obtienen una representación legible de la información en formatos HTML. La comunicación tanto entre los usuarios humanos, como los usuarios máquinas, y el RDF store se realiza en forma de consultas SPARQL.

5.7. Guías sobre Gestión de Datos para alcanzar la Interoperabilidad

La gestión de información de proyectos SmartLand se desarrolla en un entorno distribuido, autónomo y heterogéneo, y para alcanzar la interoperabilidad e integración de datos, se han acordado las siguientes buenas prácticas de datos:

1. Usar datos abiertos y plataformas abiertas para movilizar el conocimiento colectivo. Para que la iniciativa SmartLand funcione de manera efectiva, y para que la sociedad obtenga todos los beneficios de los esfuerzos científicos, es crucial que los datos científicos sean abiertos y legibles por máquina. Los datos abiertos tienen el potencial de ayudar a las comunidades a comprender, evaluar e implementar soluciones tecnológicas adaptadas a las necesidades y objetivos específicos de la comunidad.
2. Definir las licencias de uso de datos. Todos los datos o colecciones de datos que se comparten deben publicarse con una declaración clara y explícita sobre los permisos y restricciones que se han aplicado.
3. Transformar datos existentes en datos semánticamente enlazados. Establecer la semántica y las ontologías que deben ser reutilizadas en un dominio particular. Además, dado que se usarán varias ontologías, será necesario que los conceptos de diferentes ontologías se alineen y relacionen entre sí.
4. Satisfacer el requisito 'Cool URI' de SW, de modo que mejore la interoperabilidad entre sistemas independientes y haga que los URI sean lo más consistentes y persistentes posible.
5. Hacer que los potencialmente "datos enlazables" realmente se vinculen y permitan consultas complejas en un espacio de datos global y distribuido. La esencia de LD es permitir que los datos se conecten entre sí.
6. Habilitar una forma de consumir datos y semántica de manera que más usuarios puedan usarlos. Esto mejoraría la situación actual en la que la mayoría de los datos enlazados se obtienen a través de puntos de acceso SPARQL, que suponen que los usuarios estén familiarizados con las ontologías de dominio que se han usado para describir los datos.
7. Implementar un procedimiento para rastrear la procedencia de las piezas de datos. Dado que un proceso de publicación de datos enlazados transformará las fuentes de datos existentes en datos enlazados, descritos en RDF, es necesario establecer la trazabilidad de la transformación.

6. Conclusiones y Trabajos Futuros

La calidad y la información oportuna son vitales para que los gobiernos, el sector académico, las organizaciones internacionales, la sociedad civil, el sector privado y el público en general puedan tomar decisiones informadas. Por ello, la Gestión Inteligente y Sostenible del Territorio, SmartLand, requiere la recopilación, el procesamiento, el análisis y la difusión de una cantidad sin precedentes de datos e indicadores a nivel local, nacional, regional y mundial y de múltiples partes interesadas. Por lo tanto, la creación de capacidad en gestión de datos e indicadores es fundamental.

Con el fin de monitorear la implementación el desarrollo sostenible, es clave mejorar la disponibilidad y calidad de los datos e indicadores sectoriales a nivel local, alineados a los esfuerzos de organizaciones internacionales relevantes. El esfuerzo modernización y fortalecimiento planteado a través de SmartLand-LD requiere el compromiso pleno, activo y centrado, así como la confianza de las partes interesadas para el funcionamiento eficaz de una iniciativa de datos abiertos e indicadores sobre desarrollo sostenible.

El marco de trabajo describe los pasos necesarios para modernizar y fortalecer la interoperabilidad en los sistemas de indicadores. SmartLand-LD aborda acciones, con un enfoque particular en la construcción de la infraestructura semántica y la capacidad necesaria para respaldar los requisitos de datos locales, que agreguen valor a la agenda nacional y global de desarrollo sostenible.

En esta trabajo se reconoce el papel crucial de la cooperación y la apertura entre las partes interesadas en apoyar los planes y esfuerzos de los territorios en la creación de capacidades. La experiencia y las habilidades de estas partes interesadas son recursos esenciales clave para el progreso, la sostenibilidad y la modernización. La producción de indicadores alineados con el desarrollo inteligente y sostenible requiere una interacción integral entre los proveedores de datos, las partes interesadas y los usuarios. La coordinación y la racionalización de estas actividades son necesarias para evitar la duplicación de esfuerzos y canalizar los esfuerzos hacia promover una Gestión Inteligente y Sostenible del territorio.

Como trabajo futuro está abordar los desajustes de coordinación que existen entre las variables e indicadores de los proyectos SmartLand y los indicadores que están alineados a la Agenda 2030, y avanzar hacia su representación semántica. La Agenda 2030 llama explícitamente a mejorar la creación de capacidad en la gestión de datos e indicadores para apoyar los planes nacionales para implementar los objetivos de desarrollo sostenible. Es esencial que se aborden esas deficiencias a fin de permitir una mejor utilización de las indicadores generadas en el cálculo de los indicadores nacionales, regionales y mundiales de los ODS. SmartLand está diseñado como un documento vivo, abierto a posibles ajustes en una etapa posterior para dar cuenta del desarrollo futuro, por lo tanto, periódicamente, SmartLand revisará y, si es necesario, actualizará su estrategia para alcanzar su efectividad.

Agradecimientos. Los autores expresan su gratitud a UTPL por todo el apoyo institucional a SmartLand, en especial, expresan su reconocimiento a José Barbosa Corbacho, Rector, y a todas las personas involucradas en esta iniciativa.

Referencias

- Acosta, S., Piedra, N., & Suárez, J. P. (2017). The Social Distribution of Knowledge: University Ethical Commitment to the Intelligent Management of the Territory – SmartLand. In *Higher Education in the World 6. Towards a Socially Responsible University: Balancing the Global with the Local*. (pp. 330–337). GUNi SERIES ON THE SOCIAL COMMITMENT OF UNIVERSITIES. Retrieved from http://www.guninetwork.org/files/download_full_report.pdf
- Ahvenniemi, H., Huovila, A., Pinto-Seppä, I., & Airaksinen, M. (2017). What are the differences between sustainable and smart cities? *Cities*, 60, 234–245. <https://doi.org/10.1016/j.cities.2016.09.009>
- Albino, V., Berardi, U., & Dangelico, R. M. (2015). Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, 22(1), 3–21.
- Balmford, A., Bennun, L., Ten Brink, B., Cooper, D., Côté, I. M., & Crane, P. (2005). The convention on biological diversity's 2010 target. *Science*, 307 (5707), 212–213.
- Berners-Lee, T. (2006). Linked data-design issues. Retrieved from <https://www.w3.org/DesignIssues/LinkedData.html>
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), 34–43.
- CKAN. (n.d.). The open source data portal software. Retrieved from <https://ckan.org/>
- Cutter, A., Osborn, D., Romano, J., & Ullah, F. (2015). *Sustainable Development Goals and integration: Achieving a better balance between the economic, social and environmental dimensions*. German Council for Sustainable Development. Retrieved from www.unmultimedia.org/photo/
- Cyganiak, R., Wood, D., & Lanthaler, M. (2014). RDF 1.1 Concepts and Abstract Syntax. *W3C Recommendation 25 February 2014*, (February), 263–270. <https://doi.org/10.1007/s13398-014-0173-7.2>
- Deakin, M., & Waer, H. Al. (2011). From intelligent to smart cities. *Intelligent Buildings International*, 3(3), 133–139. <https://doi.org/10.1080/17508975.2011.586673>
- Dirks, S., Gurdgiev, C., & Keeling, M. (2010). *Smarter cities for smarter growth: How cities can optimize their systems for the talent-based economy*. Somers, NY, USA: IBM Global Services,
- Ensor, P. (1988). The Functional Silo Syndrome. *AME Target*. <https://doi.org/ame.org/sites/default/files/documents/88q1a3.pdf>
- Gregorio, J. (2012). *RFC 6570 - URI Template*. Retrieved from <http://tools.ietf.org/html/rfc6570>
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human - Computer Studies*, 43(5–6), 907–928. <https://doi.org/10.1006/ijhc.1995.1081>

- Guillén, A., Lloret, E., & Gutiérrez, Y. (2016). TLH Suite: Semantic information annotation tool. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, (18), 99-103. <https://doi.org/10.17013/risti.18.99-113>
- Hák, T., Janoušková, S., & Moldan, B. (2016). *Sustainable Development Goals: A need for relevant indicators*. *Ecological Indicators*. <https://doi.org/10.1016/j.ecolind.2015.08.003>
- Hallo, M., Luján-Mora, S., & Morga, A. M. (2017). Cuadros de mando para gestionar el uso de bibliotecas digitales sobre datos enlazados. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, (22), 57-72. <https://doi.org/10.17013/risti.22.57-72>
- Harris, S., & Seaborne, A. (2013). *SPARQL 1.1 Query Language*. *W3C Recommendation*. <https://doi.org/citeulike-article-id:2620569>
- Heath, T., & Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space*. (J. Hendler & F. Van Harmelen, Eds.) (1st ed.). São Rafael, CA, USA: Morgan & Claypool Publishers. <https://doi.org/10.2200/S00334ED1Vo1Y201102WBE001>
- Hollands, R. G. (2008). Will the real smart city please stand up? *City*, 12(3), 303-320. <https://doi.org/10.1080/13604810802479126>
- Kates, R. W., Parris, T. M., & Leiserowitz, A. A. (2005). What is sustainable development? Goals, indicators, values, and practice. *Environment*, 47(3), 8-21. <https://doi.org/10.1080/00139157.2005.10524444>
- Mori, K., & Christodoulou, A. (2012). Review of sustainability indices and indicators: Towards a new City Sustainability Index (CSI). *Environmental Impact Assessment Review*, 32(1), 94-106.
- Nam, T., & Pardo, T. A. (2011). Conceptualizing smart city with dimensions of technology, people, and institutions. In *Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times* (pp. 282-291).
- Piedra, N., Chicaiza, J., Lopez-Vargas, J., & Caro, E. T. (2016). Guidelines to producing structured interoperable data from Open Access Repositories. In *Proceedings of Frontiers in Education Conference, FIE* (Vol. 2016-Novem). <https://doi.org/10.1109/FIE.2016.7757660>
- Piedra, N., Chicaiza, J., Lopez, J., & Caro, E. T. (2017). A rating system that open-data repositories must satisfy to be considered OER: Reusing open data resources in teaching. In *IEEE Global Engineering Education Conference, EDUCON*. <https://doi.org/10.1109/EDUCON.2017.7943089>
- Piedra, N., Chicaiza, J., Quichimbo, P., Saquicela, V., Cadme, E., López, J., & Tovar, E. (2015). Framework for the integration of digital resources based-on a semantic web approach. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, (E3), 55-70. <https://doi.org/10.17013/risti.e3.55-70>

- Piedra, N., & Suárez, J. P. (2017). SmartLand-LD: A Linked data approach for integration of heterogeneous datasets to intelligent management of high biodiversity territories. In *Advances in Intelligent Systems and Computing* (Vol. 688, pp. 207–218). https://doi.org/10.1007/978-3-319-69341-5_19
- Piedra Pullaguari, N. O. (2016). *Marco de trabajo de interoperabilidad semántica para la Educación Abierta: un enfoque para la Integración de Sistemas de Información de Recursos Educativos Abiertos autónomos, distribuidos y heterogéneos*. Madrid: Universidad Politécnica de Madrid.
- United Nations General Assembly. (2015). *Transforming our world: The 2030 agenda for sustainable development*. Retrieved from: <https://sustainabledevelopment.un.org/content/documents/7891Transforming%20Our%20World.pdf>. <https://doi.org/10.1007/s13398-014-0173-7.2>
- UNSG (2016). Progress towards the Sustainable Development Goals. *United Nations Economic and Social Council*, (July), 28. <https://doi.org/10.1017/S0020818300011322>
- Vázquez-Reyes, S., de León-Sigg, M., Velasco-Elizondo, P., Villa-Cisneros, J., & Briceño-Muro, S. (2017). Information retrieval through inverted index in Be Intelligent system. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, (21), 85-98. <https://doi.org/10.17013/risti.21.85-98>
- W3C. (2008). Cool URIs for the Semantic Web. *W3C Interest Group Note*, 49(December 2008), 1–15. Retrieved from <http://www.w3.org/TR/cooluris/>
- W3C. (2011). *Library Linked Data Incubator Group: Use Cases*. Retrieved from <https://www.w3.org/2005/Incubator/lld/XGR-lld-usecase-20111025/>
- W3C OWL Working Group. (2012). OWL 2 Web Ontology Language Document Overview. *OWL 2 Web Ontology Language*, (December), 1–7. Retrieved from <http://www.w3.org/TR/owl2-overview/>

Un modelo basado en agentes para simular la capacidad de absorción en organizaciones

Orlando López-Cruz¹

saudiel.lopez@javeriana.edu.co

¹ Pontificia Universidad Javeriana, Calle 147 14-69 T.5-704, Bogotá D.C. 110121, Colombia.

DOI: 10.17013/risti.26.122-139

Resumen: Se presenta una re-conceptualización del constructo ‘capacidad de absorción’ (ACAP) en las organizaciones que pretende ser implementable en un programa de ordenador para facilitar la simulación de los efectos de la adquisición, asimilación y explotación de conocimiento aprovechable del entorno de la organización, con el objetivo de generar ventaja competitiva e innovación en las organizaciones. A diferencia de otras investigaciones que se enfocan en producir refinamientos al constructo de la ACAP, ésta investigación se centra en una simulación por ordenador para diseñar la ACAP real en las organizaciones. Se sugiere de una falta de diseño en los modelos conceptuales inferidos de la revisión de la literatura y se propone un nuevo diseño implementable de la ACAP. Este nuevo diseño arroja luces sobre la implementación en el entorno organizacional de la ACAP capaz de sustentar la absorción efectiva de conocimiento en las organizaciones.

Palabras-clave: Capacidad de absorción; gestión de conocimiento; innovación tecnológica, modelos basados en agentes.

An agent-based model to simulate absorption capacity in organizations

Abstract: This paper introduces a reconceptualization of the ‘absorptive capacity’ (ACAP) construct in organizations pursuing to implement a computer program to assist the simulation the results of acquisition, assimilation and harnessing of knowledge available in the environment of the organization, with the aim of obtaining competitive advantage and produce innovation in organizations. In contrast to other researches focusing in refining the ACAP construct, this research emphasizes on a computer simulation to assist the design an actual ACAP in organizations. This paper suggests a lack of design of the ACAP conceptual models inferred from the reviewed literature and a new implementable design of the ACAP construct is proposed. This new design sheds light upon an actual implementation of the ACAP construct able to support effective knowledge absorption at organizations.

Keywords: Absorptive capacity; knowledge management, technological innovation; agent-based models.

1. Introducción

El constructo ‘capacidad de absorción’ (abreviado como ACAP en alguna literatura en inglés por *Absorptive Capacity*) está relacionado con la visión de las organizaciones basada en el conocimiento (*Knowledge-Based View-KBV*) (Kathleen M Eisenhardt & Santos, 2002; Grant, 1996; Nickerson & Zenger, 2004), con el concepto de conocimiento en la teoría del aprendizaje organizacional (Argyris & Schön, 1978; Levitt & March, 1988) y con la adopción de sistemas de gestión del conocimiento desde el enfoque del receptor (León-Sigg, Vázquez-Reyes, & Villa-Cisneros, 2017; Palos-Sánchez, Arenas-Márquez, & Aguayo-Camacho, 2017; Sousa, Costa, & Aparicio, 2017). La finalidad de esta conceptualización es encontrar la forma de generar ventaja competitiva en las organizaciones (Porter, 1998), a partir de conocimiento codificado o no (explícito o tácito) (Nonaka & Takeuchi, 1995; Nonaka, Toyama, & Nagata, 2000). Puesto que la visión basada en el conocimiento trata de explicar el desempeño de las empresas (Grant, 1996) con base en el conocimiento explotable, lo crítico es hallar la forma como el conocimiento puede apropiarse y explotarse para hacer sostenible la ventaja competitiva alcanzada.

Esta investigación presenta (i) el re-diseño del constructo de la ACAP que permite describir tanto la adquisición como la explotación del conocimiento con fines de sostener ventaja competitiva y (ii) los resultados de la implementación de dicho diseño en un programa de ordenador en un sistema basado en agentes orientado a la simulación interactiva con usuarios expertos.

Este documento se estructura de la siguiente forma: se presenta el constructo de la ACAP y el concepto de ‘rutina organizacional’ para luego relacionarlos, la metodología utilizada y luego se presenta la propuesta de re-diseño del constructo, su implementación y la presentación de resultados y análisis.

2. El constructo “capacidad de absorción”

El constructo “capacidad absorbente” o “capacidad de absorción” fue extendido, en la década de los 1990’s (Cohen & Levinthal, 1990), desde el dominio de la hidrología, la ecología y la medicina, al campo de las organizaciones como una capacidad dinámica de estos sistemas sociales humanos complejos. En ese sentido, es un concepto inspirado en sistemas complejos naturales.

Este constructo, ‘capacidad de absorción’ (Cohen & Levinthal, 1990), entendido como una medida de la aptitud de una organización (Helfat et al., 2009; D. Teece & Pisano, 1994; D. J. Teece, Pisano, & Shuen, 1997) para (i) escrutar su entorno y reconocer el valor de conocimiento externo nuevo, adquirirlo, (ii) asimilarlo, (iii) transformarlo, explotarlo con sus recursos internos y que le sirva como marco para la innovación, es una variable crítica para caracterizar las posibilidades para que haya una efectiva transferencia de conocimiento tecnológico a la organización sobre la que se hace la medición, proceso similar al de la elicitación de requerimientos, que es intensivo en la transferencia de información entre los que son «poseedores del conocimiento» desde el lado del cliente-usuario y un equipo de desarrollo de software (Donoso Barraza & Vega Zepeda, 2017).

2.1. Origen y desarrollo del constructo ACAP.

El constructo ‘capacidad de absorción’ ACAP (por su apócope en inglés de *Absorptive Capacity*) fue originalmente definido con una habilidad basada en el conocimiento previo (Cohen & Levinthal, 1990) para reconocer el valor de nueva información, asimilarla y aplicarla para propósitos comerciales. Esta propuesta de ACAP es altamente conceptual y, por lo mismo, conduce a un modelo que es difícilmente implementable en modelos por ordenador.

Se proponen cuatro periodos cronológicos del desarrollo del constructo ‘capacidad de absorción’.

El primer periodo es la aparición del constructo, el segundo periodo es el de fundamentación conceptual que ha permitido conformar un dominio del conocimiento. El hito de inicio de este periodo es la aparición de la primera re-conceptualización de la capacidad de absorción como una “capacidad dinámica” (Zahra & George, 2002). En un tercer periodo, se consolida la investigación empírica de capacidad de absorción en *business research* (Lewin, Massini, & Peeters, 2011) relacionándose con la investigación sobre ventaja competitiva, transferencia de tecnología (patentes e invenciones), aprendizaje organizacional y, finalmente, en el más reciente periodo se extiende el dominio de investigación de la capacidad de absorción a la ingeniería de organizaciones, se estrechan las relaciones con los conceptos de transferencia de conocimiento, sistemas basados en conocimiento (base de conocimiento + motores de inferencia o sistemas de razonamientos simbólico), adquisición de conocimiento, representación de conocimiento y capital humano.

El constructo “capacidad de absorción” emergió (Cohen & Levinthal, 1989, 1990, 1994) para referirse a una capacidad dinámica (Helfat et al., 2009; Prahalad & Hamel, 2006; D. Teece & Pisano, 1994; D. J. Teece et al., 1997) de una organización expresada en tres dimensiones, capacidad para:

- i. identificar o reconocer,
- ii. asimilar y
- iii. aplicar o aprovechar el conocimiento disponible en el entorno.

A inicios del siglo XXI, la capacidad de absorción fue re-conceptualizada (Zahra & George, 2002) como el conjunto de rutinas organizacionales y procesos por los cuales las organizaciones:

- i. adquieren,
- ii. asimilan,
- iii. transforman y
- iv. explotan

el conocimiento para producir una capacidad dinámica (Helfat et al., 2009; Prahalad & Hamel, 2006; D. Teece & Pisano, 1994; D. J. Teece et al., 1997).

Una organización cuenta con capacidades organizacionales operacionales (sustantivas) y dinámicas (Winter, 2000). Una capacidad organizacional es una rutina de alto nivel que, con procesos y recursos específicos de la organización, le confiere a sus directivos

un conjunto de opciones de decisión para producir resultados significativos de una clase particular, usualmente el logro de metas estratégicas, utilizando el know-how disponible y recursos no específicos de la organización (D. J. Teece, 1977; D. J. Teece et al., 1997; Winter, 2000). Una capacidad operacional se refiere a competencias y habilidades que determinan la efectividad de una organización para ejecutar sus actividades misionales cotidianas. Una capacidad dinámica organizacional es la capacidad para integrar, construir y reconfigurar capacidades operativas (D. J. Teece et al., 1997), así como competencias internas y externas para afrontar cambios rápidos del entorno (Zollo & Winter, 2002). Está orientada hacia el cambio organizacional, por lo que permite formular una respuesta apropiada e implementar un curso de acción de la organización, reflejando su carácter evolutivo.

Una nueva reconceptualización (Todorova & Durisin, 2007) retoma la dimensión de (i) identificar o reconocer el valor del conocimiento, separa la (ii) adquisición de la (iii) asimilación y deja esta última como una etapa o dimensión alternativa frente a la (iv) transformación de conocimiento, previa a la (iv) explotación (aplicación o aprovechamiento) del conocimiento. Además, quizás lo más valioso, es la inclusión de un ciclo de retroacción (*feedback*) que hace coherente al constructo con la dinámica que no puede ser capturada por los métodos analíticos (Todorova & Durisin, 2007), lo que revela una oportunidad de seleccionar una metodología de investigación apropiada a la investigación en ingeniería.

En una perspectiva de categorización, los resultados de los estudios sobre ACAP pueden ser clasificados en: identificación de los componentes del constructo ACAP (González, 2014; Nishiyama, Ikeda, & Niwa, 2000; Todorova & Durisin, 2007; Vega, Gutiérrez, & Fernández de Lucio, 2007), el papel de las condiciones contextuales y otros factores externos relacionados con la innovación en la organizaciones (Fabrizio, 2009) y sus aplicaciones (López-Paniagua, Nieto-Carlier, Rodríguez-Martín, González-Fernández, & Jiménez-Álvaro, 2011; Volberda, Foss, & Lyles, 2009), diseño de instrumentos de medición (Marín, Laureiro, & Forero, 2007), investigación experimental (Filgueiras Sainz de Rozas, Castro Fernández, & Rafull Suárez, 2013) y su relación con la transferencia de tecnología (López-Cruz, 2010).

2.2. El constructo ACAP y las rutinas organizacionales.

Pasados doce años de la publicación del artículo seminal (Cohen & Levinthal, 1990), el constructo ACAP fue re-conceptualizado en forma importante (Zahra & George, 2002) como un conjunto de rutinas organizacionales. Es decir, la ACAP es evocada como una capacidad dinámica que amplía las habilidades organizacionales para obtener y sostener la ventaja competitiva (Porter, 1998) y para generar cambios organizacionales de manera estratégica.

Al comparar ésta propuesta (Zahra & George, 2002) con el constructo original (Cohen & Levinthal, 1990), la ACAP se transforma en (i) una capacidad dinámica, (ii) las rutinas organizacionales son los componentes principales de la ACAP y (iii) se identifican condiciones bajo las cuales los componentes de la ACAP son capaces de crear valor. Esto permite conducir investigaciones sobre las diferencias entre dos organizaciones que operan en el mismo sector económico y, más importante aún, prescribir esas diferencias.

2.3. El constructo ACAP y las capacidades organizacionales.

Una capacidad organizacional es una rutina de alto nivel que, por medio de procesos y recursos específicos organizacionales, proveen a la alta gerencia elementos de juicio para tomar decisiones que producen resultados significativos de alguna clase en particular. Usualmente, estos resultados son metas estratégicas para usar conocimiento (*know-how*) disponible y recursos organizacionales no-específicos (D. J. Teece, 1977; Winter, 2000).

Las capacidades organizacionales proceden de tres factores:

- i. recursos,
- ii. procesos y
- iii. valores (Christensen & Overdorf, 2000).

De las capacidades organizacionales, que pueden ser capacidades operacionales (sustantivas) y capacidades dinámicas, las capacidades dinámicas se originan, entre otras fuentes, en las estructuras de la organización (Easterby-Smith, Graca, Antonacopoulou, & Ferdinand, 2005) lo cual implica que las capacidades dinámicas están determinadas funcionalmente por variables de la estructura de la organización. Es decir, las variables de la estructura de la organización determinan las capacidades dinámicas. Esto no implica que sean las únicas variables que determinan las capacidades dinámicas, puesto que se ha identificado a partir de otros estudios que no hay una relación causa efecto entre las variables de estructura de la organización y las capacidades dinámicas (Dozier & Chang, 2009). Además, al parecer tampoco hay una relación unidireccional causa-efecto entre las variables de estructura organizacionales y las capacidades dinámicas (Cordes-Berszinn, 2013).

Puesto que la ventaja competitiva de una organización (Porter, 1998) se fundamenta en sus capacidades dinámicas (D. Teece & Pisano, 1994), entonces las organizaciones necesitan desarrollarlas (Giovanni Dosi & Marengo, 2000; Kathleen M. Eisenhardt & Martin, 2000; D. J. Teece et al., 1997; Zollo & Winter, 2002) para adaptarse a su entorno dinámico.

Lo esencial de las competencias y las capacidades es que no pueden ser adquiridas en el mercado, de la misma forma que las propiedades internas de las organizaciones no pueden ser replicadas como un conjunto de unidades de negocio (a través de contratos formales en el mercado). Las capacidades organizacionales deben ser entendidas tanto en términos de estructuras organizacionales como de procesos administrativos (D. J. Teece et al., 1997 p.517) y no en términos del balance general, o de los estados financieros en general.

Los componentes de la ACAP –las rutinas organizacionales- (Bobrow & Winograd, 1977) – se agrupan en dos conjuntos: capacidad de absorción potencial y capacidad de absorción realizada. Las rutinas de ACAP potencial son adquisición de conocimiento y asimilación de conocimiento. Las rutinas de ACAP realizada son transformación y explotación. El objetivo de las rutinas de ACAP potencial es incentivar la receptividad al conocimiento externo, que es débilmente equivalente a la dimensión “reconocer el valor del conocimiento (Cohen & Levinthal, 1990). Las rutinas de ACAP realizado buscan apalancar el aprendizaje organizacional con base en el conocimiento absorbido.

Los investigadores de la primera década del siglo XXI sugirieron que estas rutinas organizacionales proveían mecanismos sistémicos, estructurales y procedimentales

para que la organización explotara el conocimiento y obtuviera beneficios de largo plazo pese a las barreras como la ‘apropiabilidad’ (es decir, las dinámicas institucionales como la propiedad intelectual, que afecta las capacidades organizacionales para proteger sus derechos de beneficiarse de los nuevos productos o procesos). La ‘apropiabilidad’ del conocimiento está en relación inversa con las propensión organizacional a invertir en investigación y desarrollo (Lane, Koka, & Pathak, 2006).

Sin embargo, este modelo (Zahra & George, 2002) exhibe ambigüedades (Todorova & Durisin, 2007), por ejemplo la omisión de rutinas que en la propuesta original son sólo componentes del constructo para “reconocimiento del valor del conocimiento externo”. La literatura advierte (Todorova & Durisin, 2007) que los modelos analíticos lineales (Cohen & Levinthal, 1990; Zahra & George, 2002) no describen en forma apropiada la complejidad de las relaciones causales de la ACAP que explican la dinámica de absorción a través de los procesos de aprendizaje organizacional (Lane et al., 2006).

Un nuevo modelo de la ACAP (Todorova & Durisin, 2007) (i) retoma el componente de identificación del valor del conocimiento (Cohen & Levinthal, 1990), extiende el alcance de la incidencia (bien sea positiva o negativa) de los mecanismos sociales de integración (Easterby-Smith et al., 2005) a todas las rutinas organizacionales teorizando los factores de contingencia, (iii) incluye en el conjunto de factores de contingencia a las relaciones de poder que influyen tanto a las rutinas de valoración del conocimiento, como a las rutinas de explotación del conocimiento, (iv) reconoce las relaciones complejas entre las dimensiones de asimilación y transformación, (v) redefine las rutinas de transformación y (vi) incluye un ciclo de retroacción que caracteriza la dinámica y complejidad del fenómeno.

Desde este punto de vista, la organización puede ser concebida como un repositorio de conocimiento. Una organización puede ser modelada como un instrumento para transferir y desarrollar conocimiento no sólo a otras organizaciones con las que se relacione (es decir, no sólo para crear y aplicar conocimiento, que es una visión estática), sino que se expone a un flujo de conocimiento. De aquí que, la ACAP parece ser el constructo más apropiado para explicar la forma como las organizaciones pueden realmente estar en capacidad de adquirir conocimiento y hacerlo productivo. Esto es importante porque estudios previos indican que la transferencia de conocimiento depende del nivel de la ACAP como también de la complejidad del conocimiento que está siendo transferido y de los aspectos sociales en entorno turbulentos (Dhanaraj, Lyles, Steensma, & Tihanyi, 2004).

Una estructura que se enfoca en la gestión del conocimiento (Huber, 1991) se despliega en cuatro procesos de aprendizaje: (1) adquisición de conocimiento, (2) distribución de información, (3) interpretación de información y (4) memoria organizacional. Esta estructura de procesos semeja y complementa el constructo de la ACAP.

3. Metodología

El paradigma de la investigación basada en el diseño se origina en las ciencias de lo artificial (A. Hevner, March, Park, & Ram, 2004; Simon, 1996), entre las que se cuenta la ingeniería. La finalidad prioritaria de las ciencias de la ingeniería no es crear teorías de validez universal, describir, entender, comprender, predecir o explicar la naturaleza de los fenómenos (lo que es). Todo lo anterior sí es prioridad de las ciencias naturales.

Para las ciencias de lo artificial es objetivo prioritario el diseño, construcción e integración de artefactos al mundo (lo que puede ser) (A. R. Hevner & Chatterjee, 2010), incrustarlos adecuadamente en la realidad con una finalidad, por lo que el aspecto relevante de las ciencias de lo artificial es la interfaz entre las leyes naturales que gobiernan los entornos interno y externo al artefacto con el fin último de alcanzar logros adaptando las leyes internas a las leyes externas. Por lo anterior, la investigación basada en diseño, en contraste con la investigación en ciencia, está enfocada en cómo deben ser las cosas, lo que advierte elementos deontológicos de este enfoque de investigación, concibiendo artefactos que cumplan con metas (Simon, 1996 p.113-114).

Ahora bien, las metas están asociadas con el espacio de problemas ubicado entre la situación actual y la situación deseada (Simon, 1996). Y en el campo de la investigación de sistemas de información un paso importante es la demostración de que un artefacto de tecnología de información (TI) es o no es adecuado para una un problema especificado, como también lo es la comparación de la utilidad de artefactos TI existentes dentro de contextos organizacionales específicos (March & Storey, 2008). Tanto los sistemas de información como las rutinas organizacionales son extensiones de las capacidades cognitivas humanas (A. R. Hevner & Chatterjee, 2010).

La investigación basada en el diseño se realiza en tres ciclos:

- i. el ciclo de relevancia, que vincula el entorno contextual del proyecto de investigación con las actividades de diseño: “una buena investigación basada en el diseño empieza por identificar y representar oportunidades en un entorno de aplicación real”;
- ii. el ciclo de rigor, que conecta las actividades de investigación basada en diseño con la base de conocimiento, la experiencia que provee información al proyecto de investigación, provee el conocimiento previo requerido para el proyecto de investigación y asegurar su innovación, y
- iii. el ciclo de diseño, que itera entre las actividades de núcleo de construcción de un artefacto, su evaluación y la retroacción subsecuente para refinar más el diseño (A. R. Hevner & Chatterjee, 2010).

La construcción del modelo basado en agentes se realizó mediante el protocolo ODD (Grimm et al., 2006; Müller et al., 2013) para separar el proceso de modelado de los detalles de implementación y del lenguaje de programación (Grimm et al., 2006). El modelado basado en agentes es recomendado en entornos en donde se hace imperativo explorar posibles comportamientos o escenarios en el contexto de la incertidumbre (Alexandre & Balsa, 2018). De esta forma, se elaboró el modelo conceptual de un sistema basado en agentes. En ese modelo, los agentes son organizaciones, el flujo de conocimiento se representa por ‘artefactos tecnológicos’ de la misma forma que el ‘conocimiento’ externo a las organizaciones y las barreras organizacionales para acceder al conocimiento. Estos describen el dominio situacional y las especificaciones del modelo para las fases del protocolo ODD (I) Resumen, (II) Conceptos de diseño y (III) Detalles.

El conocimiento es entendido y representado como rutinas organizacionales. Una organización del modelo tiene su propio conjunto de rutinas. Una rutina organizacional tiene acceso completo a los ‘artefactos tecnológicos’ y ‘el conocimiento’ en su vecindario (tienen la capacidad de identificar conocimiento en el flujo de conocimiento) con el fin de

decidir si “capturan” o no el conocimiento en el flujo externo. Además, una organización tiene una ‘base de conocimiento’ que es una lista de tareas para realizar sus actividades. El conocimiento decae en datos y, por eso, desaparece del modelo.

La verificación del modelo incluye la verificación del código en términos de (i) detección de errores y depuración en el programa de ordenar y (ii) evaluación de que el programa implementado corresponda con sus especificaciones. Esto lleva a evaluar si el modelo conceptual ha sido implementado en el lenguaje de programación. Esto se hizo antes del proceso de validación (Sankararaman & Mahadevan, 2015). La validación del sistema basado en agentes es una tarea difícil cuando se aborda de la misma manera que los modelos científicos tradicionales (Klügl, 2008; Parunak, Savit, & Riolo, 1998; Xiang, Kennedy, Madey, & Cabaniss, 2005). En esta investigación, la validación fue realizada por sujetos expertos (Niazi, Hussain, & Kolberg, 2009) vinculados a organizaciones del sector salud y de seguros.

Este estudio usó el software Netlogo 6.0, un proyecto colaborativo de NorthWestern University, para implementar y tratar por medio de un modelo de simulación la complejidad de la emergencia de rutinas organizacionales, es decir el macro-nivel, a partir de las acciones de ejecución de tareas de los individuos del sistema, es decir el micro-comportamiento, incluyendo su exposición a un flujo de información que representa conocimiento del entorno.

Netlogo ha sido referido como un entorno de software para modelamiento de sistemas basados en agentes (MSBA), junto con Repast, Swarm y MASON (Fioretti, 2013; Li, Mao, Zeng, & Wang, 2008; Tissue & Wilensky, 2004).

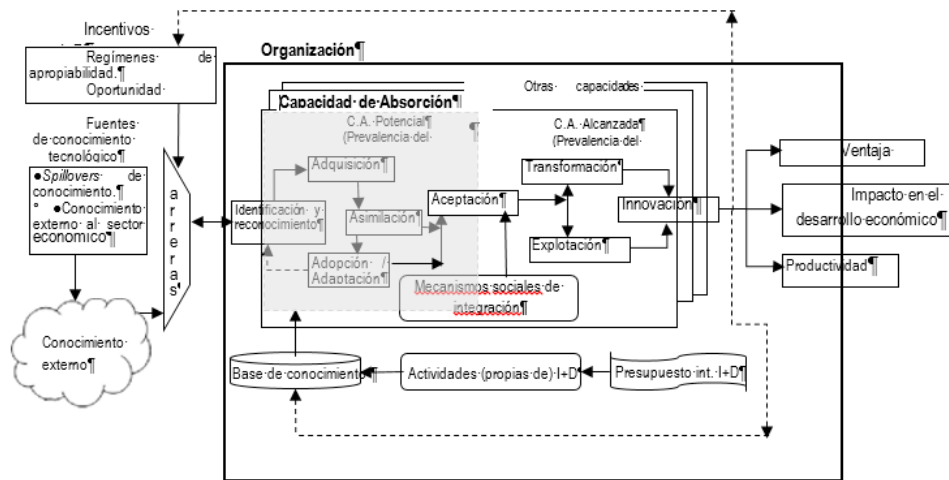
Como medio de validación se recurrió a las pruebas de campo con expertos en las organizaciones seleccionadas, de conformidad con los ciclos de validación basada en el diseño.

4. Configuración de la Capacidad de Absorción Organizacional

Un estudio previo (Lopez-Cruz & Obregon Neira, 2016) identifica al conocimiento como el elemento principal involucrado en la dinámica organizacional, confirmando que los elementos intangibles son los principales para generar identidad organizacional y ventaja competitiva sostenible (Calix, Vigier, & Briozzo, 2015). Con el fin de aprovechar esta ventaja, generar un simulador por ordenador y evitar las fallas de modelos previos, se propone el re-diseño del constructo de la ACAP. Mientras que algunos modelos de la ACAP (Cohen & Levinthal, 1990; Todorova & Durisin, 2007) carecen de una especificación explícita de las interrelaciones de las organizaciones y su entorno, el diseño que presenta esta investigación considera elementos ambientales como determinante de los componentes del constructo. Además, la meta de este nuevo modelo no es describir sino prescribir la forma como las organizaciones pueden implementar ACAP. Esto se hace suministrando a las organizaciones una ayuda para diseñar sus propios medios para identificar las interacciones entre ellas (las organizaciones) y su entorno de donde se toma el conocimiento.

En este rediseño, se identificaron varias rutinas organizacionales como elementos que causan que el conocimiento fluya y sea atrapado o no. A partir de esto, las rutinas

se organizan en clases o categorías: Identificación y reconocimiento, adquisición, asimilación, adopción/adaptación, aceptación, transformación, explotación e innovación. Finalmente, son ensambladas en un único artefacto por medio de flujos de conocimiento (ambiental) (figura 1).



Nota: La línea continua gruesa representa la frontera organizacional. Más que una muralla infranqueable, en esta representación la frontera representa la interacción entre la organización misma y su entorno, por lo que es importante que algunos componentes son "intersecados" por la frontera, con lo cual se quiere expresar que no sólo son elementos organizacionales, sino que además son componentes que inciden con el entorno como son: (i) ventaja competitiva, (ii) impacto en el desarrollo económico y (iii) productividad. Además, la representación de la frontera permite distinguir los ámbitos de los ciclos de retroacción de esta propuesta del diseño del constructo.

Figura 1 – Un re-diseño de la capacidad de absorción (ACAP).

4.1. Información, conocimiento y ACAP.

El conocimiento es diferente de información. El conocimiento puede ser obtenido únicamente por medio de procesos de cognición e interpretación (Tecuci, 1992). Las organizaciones de aprendizaje adquieren conocimiento por medio de aprendizaje adaptativo –asumiendo cambios ambientales y haciendo cambios menores en los modelos mentales (paradigmas) y vía aprendizaje generativo –que es creativo y produce cambios en el modelo “mental” organizacional (Senge, 1990). El conocimiento en las organizaciones se verifica en relación con las estructuras y rutinas organizacionales, dado que no se deriva de la información disponible en un miembro en particular de la organización pero emerge como una propiedad del sistema de aprendizaje materializado en las personas (i.e. capital humano) (Rodríguez, Aguirre, & Ordóñez, 2015). Además, está formado por la interacción entre diferentes procesos de aprendizaje que integrados a la organización (G. Dosi & Marengo, 1993).

El conocimiento tácito está incorporado parcialmente en hábitos y rutinas organizacionales. Esto hace que el conocimiento tácito sea irreducible a información (Tecuci, 1992) y que no pueda ser transmitido en forma codificada. Por otro lado, los agentes económicos

dependen del conocimiento tácito (Bareiss, Porter, & Murray, 1989) y, por su parte, todas las habilidades y decisiones dependen del conocimiento tácito. De aquí la importancia de tener un constructo que permita una implementación más real y concreta.

El diseño del constructo que se propone aquí (Figura 1) reconoce la ACAP como un constructo complejo (Morin, 2011). La ACAP está integrada por componentes distintos, pero que interactúan mutuamente, que a su turno le permiten a la organización interactuar con su entorno. Sus componentes son rutinas (ACAP potenciales y alcanzadas) para identificar y reconocer el valor del conocimiento del entorno organizacional, así como rutinas para asimilación, adopción/adaptación, aceptación, transformación, explotación e innovación (Figura 1).

En este modelo, la ACAP se reconoce como una capacidad dinámica (una de muchas capacidades dinámicas de la organización). Este diseño retoma la distinción entre la ACAP potencial para representar el dominio donde la acción individual prevalece, y la ACAP alcanzada, donde prevalece la acción y el conocimiento colectivo prevalece sobre el individual.

El rol de las interacciones con el entorno (Figura 1) –algunos de ellos con información de retroacción como resultado de la transformación del conocimiento y la explotación– es capturar la alteración real de los regímenes de apropiación, debidos a la explotación del conocimiento, por medio de patentes o *copyrights* sobre el conocimiento que ha sido creado internamente. Además, este diseño captura la retroacción interna de la base de conocimiento que es alterada por el propio presupuesto de I+D (presupuesto interno) y es alterado actividades de I+D internas, si existen.

La contribución principal del re-diseño de este constructo, que aquí se presenta, es la factibilidad de operacionalización cuando se implementa (para implementar) cada conjunto de rutinas organizacionales. Más aún, este diseño incluye varios ciclos de retroacción (líneas punteadas de derecha a izquierda en la Figura 1), que representan distintos elementos de conocimiento que alteran la ACAP en sí misma (como el eje dirigido que cruza desde ‘adopción/adaptación’ a ‘identificación y reconocimiento’) o dentro de la frontera de la organización (como el eje dirigido desde la ‘innovación’ hasta la ‘base de conocimiento’) o la organización y su entorno (como el eje dirigido desde ‘innovación’ hasta ‘incentivos de I+D. hasta donde se sabe, ningún otro modelo está más cerca de representar la naturaleza dinámica de la ACAP y sus ciclos de retroacción, que este modelo que se presenta aquí. Esto puede facilitar el desarrollo de modelos computacionales de simulación.

4.2. Implementación de la ACAP.

El modelo computacional de la ACAP fue implementado en NetLogo 6.0.1 (Wilensky, 1999) sobre sistema operacional Windows 10®. Una rutina organizacional se implementa como una lista ordenada de tareas 1 hasta n , $n \in \mathbb{N}$ (Miller, Pentland, & Choi, 2012). La rutina organizacional consiste de una k -upla de tareas ($1 \leq k \leq n$). Sea t_i ($1 \leq i < j < k < l \leq m$; $i, m \in \mathbb{N}$) una unidad mínima de acciones. La lista de tareas $[t_i, t_j, t_k, t_l]$ es una rutina organizacional integrada por cuatro tareas: t_i, t_j, t_k, t_l ; $i < j < k < l$. En general, cada agente en el flujo de conocimiento exhibe una rutina organizacional (una lista) $[t_1, t_2, \dots, t_n]$ donde n es el tamaño de la lista.

5. Resultados and análisis

Esta implementación facilitó la experimentación con diferentes configuraciones de la ACAP de una organización. Cada ejecución podría concluir en diferentes resultados, pero la relevancia era la obtención de distintos comportamientos como resultado de un cambio de uno o más parámetros iniciales (deslizadores de NetLogo) que controlan el flujo de conocimiento y de barreras. Además, el parámetro 'tiempo de vida' (ciclo de vida) tuvo sentido para los usuarios para representar distintos ciclos de negocios de diferentes productos de ICT. También, el 'tiempo de vida' permitió representar la obsolescencia del 'conocimiento' en las organizaciones. Esto es usualmente el caso, cuando, por ejemplo, una persona sabe cómo usar sistemas operacionales con líneas de comando como UNIX, iRMX, CP/M o MS-DOS y luego deben cambiar a sistemas operacionales con interface gráfica tipo GUI (*Graphical User Interface*).

Cuando los usuarios se concentraron en adquirir artefactos tecnológicos basados en conocimiento, bajo algunas circunstancias la cantidad de 'personas con conocimiento' en el flujo de conocimiento disminuyó

En algunas ejecuciones del simulador, las 'personas con conocimiento' en el flujo de conocimiento desaparecieron del todo. Los expertos explicaron esto como el resultado de la obsolescencia del conocimiento que las personas poseen y no existe una 'actualización' del conocimiento dentro de las organizaciones. Este es un resultado posible debido a que este modelo no representa organizaciones como universidades, institutos tecnológicos o centros de investigación. Un resultado simétrico aparece cuando el usuario se enfoca en reclutar 'personas con conocimiento' del flujo externo a la organización. Esto es un contexto extremo donde no hay industria tecnológica o comercio internacional. Así se genera un rezago tecnológico para las organizaciones y su vecindario o sector económico contextual.

Además de representar un rezago tecnológico, también representa la ausencia de políticas de investigación y desarrollo (organizacionales o nacionales) que tiendan a crear incentivos para la creación de nuevos productos (o servicios) que se puedan incorporar al flujo de conocimiento, lo que es otra forma de llamar a la innovación.

De acuerdo con los usuarios expertos, el ciclo de vida del conocimiento y el enfoque 'balanceado' en la adquisición de artefactos tecnológicos y 'personas con conocimiento' no garantiza el éxito de la organización en su contexto económico, que al parecer fue el comportamiento del modelo computacional en algunas situaciones. Además, un exceso en las barreras de conocimiento puede degradar el flujo de conocimiento en forma temprana.

6. Conclusiones

La creación de un modelo por ordenador que permita evaluar la configuración dinámica de la ACAP con base en el constructo ACAP ha permitido la validación funcional de esta capacidad dinámica en términos conceptuales de (i) identificar, (ii) asimilar y (iii) explotar el conocimiento disponible en el entorno (Cohen & Levinthal, 1990). El modelo de simulación que se implementó en NetLogo (Wilensky, 1999) también facilitó la observación del usuario experto sobre la forma como sus distintas decisiones de acceder al flujo externo de conocimiento, crearon ventaja competitiva y mejoraron la productividad organizacional.

Como consecuencia de lo anterior, una contribución de esta investigación es la propuesta de un re-diseño de la ACAP (figura 1), con ciclos de retroacción (*feedback*) externos e internos, en el que el conocimiento organizacional se representa mediante rutinas organizacionales y estas a su vez se implementan con una estructura de datos subyacente a un sistema basado en agentes por ordenador.

El segundo aporte de esta investigación es la evidencia de que el constructo de la ACAP es implementable en un modelo en ordenador basado en agentes no sólo para comprobar resultados estadísticos históricos, sino para propósitos de indagación y afinación de decisiones de diseño, cuyo beneficio es la capacidad de asistir en la prescripción de los componentes del constructo.

7. Investigación futura

Hay varias oportunidades para continuar el desarrollo de estos resultados. En primer lugar, puede realizarse una evaluación de la completitud del re-diseño de la ACAP con el fin de determinar si este constructo representa todas las rutinas organizacionales con las que cuentan las organizaciones innovadoras.

Además, un desarrollo computacional adicional puede ser conducido para enriquecer el modelo incluyendo el rol de las universidades y centros de investigación para ‘reproducir’ el conocimiento “incorporado” en la gente y en los artefactos tecnológicos que eventualmente fluyen dentro del flujo de conocimiento al que se exponen las organizaciones.

La implementación de rutinas organizacionales puede ampliarse para incluir otras categorías conceptuales que las entienden a ellas como (i) conjunto de reglas o procedimientos estándar y (ii) como tendencias que se vinculan a comportamientos previamente adquiridos que pueden ser ‘activados’ por los estímulos y contextos apropiados.

Referencias

- Alexandre, C., & Balsa, J. (2018). Um Sistema Multiagente no Combate ao Braqueamento de Capitais *RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação* (25), 1–17.
- Argyris, C., & Schön, D. A. (1978). *Organizational learning: A theory of action perspective* (Vol. 173). Reading, MA, USA: Addison-Wesley.
- Bareiss, E. R., Porter, B. W., & Murray, K. S. (1989). Supporting strat-to-finish development of knowledge bases. *Machine Learning*, 4, 259–283.
- Bobrow, D. G., & Winograd, T. (1977). An overview of KRL, a knowledge representation language. *Cognitive Science*, 1(1), 3–45.
- Calix, C. G., Vigier, H. P., & Briozzo, A. E. (2015). Capital intelectual y otros determinantes de la ventaja competitiva en empresas exportadoras de la zona norte de Honduras. *Suma de Negocios*, 6(14), 130–137.
- Cohen, W. M., & Levinthal, D. A. (1989). Innovation and learning: the two faces of R & D. *The Economic Journal*, 569–596.

- Cohen, W. M., & Levinthal, D. A. (1990). Absorptive capacity: a new perspective on learning and innovation. *Administrative science quarterly*, (35), 128–152.
- Cohen, W. M., & Levinthal, D. A. (1994). Fortune favors the prepared firm. *Management Science*, 40(2), 227–251.
- Cordes-Berszinn, P. (2013). *Dynamic capabilities: how organisational structures affect knowledge processes*. Berlin: Springer.
- Christensen, C. M., & Overdorf, M. (2000). Meeting the challenge of disruptive change. *Harvard business review*, 78(2), 66–77.
- Dhanaraj, C., Lyles, M. A., Steensma, H. K., & Tihanyi, L. (2004). Managing tacit and explicit knowledge transfer in IJVs: the role of relational embeddedness and the impact on performance. *Journal of International Business Studies*, 35(5), 428–442.
- Donoso Barraza, Í., & Vega Zepeda, V. (2017). Factores sociales y humanos que afectan el proceso de educación de requerimientos: una revisión sistemática. *RISTI-Revista Ibérica de Sistemas e Tecnologías de Informação*, (24), 69–83.
- Dosi, G., & Marengo, L. (1993). Toward an evolutionary theory of organizational competencies. In R. W. England (Ed.), *Evolutionary concepts in contemporary economics*. Ann Arbor: University of Michigan Press.
- Dosi, G., & Marengo, L. (2000). On the tangled discourse between transaction costs economics and competence-based views of the firms: Some comments. *Competence, governance, and entrepreneurship: Advances in economic strategy research*, 80–92.
- Dozier, K., & Chang, D. (2009). *The role of information technology in managing supply chain fluctuations*. Paper presented at the WSEAS Management, Marketing and Finances conference in Houston.
- Easterby-Smith, M., Graca, M., Antonacopoulou, E., & Ferdinand, J. (2005). *Absorptive capacity in practice: an empirical examination of Zahra and George's model*. retrieved from: <https://pdfs.semanticscholar.org/3230/08710fed3bf250ad0042cf3360e2690133de.pdf>
- Eisenhardt, K. M., & Martin, J. A. (2000). Dynamic capabilities: What are they. *Strategic management journal*, 21(1), 1105–1121.
- Eisenhardt, K. M., & Santos, F. M. (2002). Knowledge-based view: A new theory of strategy. *Handbook of strategy and management*, 1, 139–164.
- Fabrizio, K. R. (2009). Absorptive capacity and the search for innovation. *Research policy*, 38(2), 255–267.
- Filgueiras Sainz de Rozas, M. L., Castro Fernández, M., & Rafull Suárez, I. (2013). Determinación de la capacidad de absorción: estudio de caos en la empresa GEYSEL. *Ingeniería Energética*, XXXIV(3), 175–185.
- Fioretti, G. (2013). Agent-based simulation models in organization science. *Organizational Research Methods*, 2013(journals.sagepub.com).

- González, I. (2014). Modelo teórico de capacidad de absorción, innovación organizacional y emprendimiento. *Informe de Investigaciones Educativas*, 27(1), 290–340.
- Grant, R. M. (1996). Toward a knowledge-based theory of the firm. *Strategic management journal*, 17(S2), 109–122.
- Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., & Huse, G. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological modelling*, 198(1), 115–126.
- Helfat, C. E., Finkelstein, S., Mitchell, W., Peteraf, M., Singh, H., Teece, D., & Winter, S. G. (2009). *Dynamic capabilities: Understanding strategic change in organizations*: Hoboken: John Wiley & Sons.
- Hevner, A., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 28(1), 75–105.
- Hevner, A. R., & Chatterjee, S. (2010). *Design research in information systems : theory and practice*. New York, London: Springer.
- Huber, G. P. (1991). Organizational learning: The contributing processes and the literatures. *Organization science*, 2(1), 88–115.
- Klügl, F. (2008). *A validation methodology for agent-based simulations*. In Proceedings of the 2008 ACM symposium on Applied computing. Fortaleza, Ceara, Brazil: ACM.
- Lane, P. J., Koka, B. R., & Pathak, S. (2006). The reification of absorptive capacity: A critical review and rejuvenation of the construct. *Academy of management review*, 31(4), 833–863.
- León-Sigg, M. d., Vázquez-Reyes, S., & Villa-Cisneros, J. L. (2017). Factores que afectan la adopción de Tecnologías de Información en micro y pequeñas empresas: un estudio cualitativo. *RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação*, (22), 20–36.
- Levitt, B., & March, J. G. (1988). Organizational learning. *Annual review of sociology*, 14, 319–340.
- Lewin, A. Y., Massini, S., & Peeters, C. (2011). Microfoundations of internal and external absorptive capacity routines. *Organization science*, 22(1), 81–98.
- Li, X., Mao, W., Zeng, D., & Wang, F.-Y. (2008). *Agent-based social simulation and modeling in social computing*. Paper presented at the International Conference on Intelligence and Security Informatics. Taipei, Taiwan.
- López-Cruz, O. (2010). Transferencia de tecnología informática: entorno colombiano. *Revista de Tecnología - Journal of Technology*, 9(1), 19–23.
- Lopez-Cruz, O., & Obregon Neira, N. (2016). *Diseño de la capacidad de absorción en las organizaciones: propuesta de un nuevo constructo y literatura*. Paper presented at the Congreso Nacional e Internacional en Innovación en la Gestión de Organizaciones, Abril, 2016, Bogotá.

- López-Paniagua, I., Nieto-Carlier, R., Rodríguez-Martín, J., González-Fernández, C., & Jiménez-Álvarez, Á. (2011). Clases prácticas: Una herramienta esencial en la enseñanza de las ingenierías en el marco del Espacio Europeo de Educación Superior. *Dyna*, 86(5), 523–530.
- March, S. T., & Storey, V. C. (2008). Design science in the information systems discipline: an introduction to the special issue on design science research. *MIS quarterly*, 32(4), 725–730.
- Marín, A., Laureiro, D., & Forero, C. (2007). Innovation patterns and intellectual property in SMEs of a developing country. *Galeras de Administración*, 17.
- Miller, K. D., Pentland, B. T., & Choi, S. (2012). Dynamics of performing and remembering organizational routines. *Journal of Management Studies*, 49(8), 1536–1558.
- Morin, E. (2011). *Introducción al pensamiento complejo*. Barcelona, España: Gedisa.
- Müller, B., Bohn, F., Dreßler, G., Groeneveld, J., Klassert, C., Martin, R., & Schwarz, N. (2013). Describing human decisions in agent-based models—ODD+ D, an extension of the ODD protocol. *Environmental Modelling & Software*, 48, 37–48.
- Niazi, M. A., Hussain, A., & Kolberg, M. (2009). *Verification and validation of agent based simulations using the VOMAS (virtual overlay multi-agent system) approach*. Paper presented at the MAS&S at Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW), Turin, Italy, September 7–10, 2009.
- Nickerson, J. A., & Zenger, T. R. (2004). A knowledge-based theory of the firm—The problem-solving perspective. *Organization science*, 15(6), 617–632.
- Nishiyama, T., Ikeda, K., & Niwa, T. (2000). *Technology Transfer Macro-Process: A Practical Guide for the Effective Introduction of Technology*. Paper presented at the ICSE 2000, Limerick Ireland, June 4–11, 2000.
- Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company : how Japanese companies create the dynamics of innovation*. New York: Oxford University Press.
- Nonaka, I., Toyama, R., & Nagata, A. (2000). A firm as a knowledge-creating entity: a new perspective on the theory of the firm. *Industrial and corporate change*, 9(1), 1–20.
- Palos-Sánchez, P. R., Arenas-Márquez, F. J., & Aguayo-Camacho, M. (2017). La adopción de la tecnología cloud computing (SaaS): efectos de la complejidad tecnológica vs formación y soporte. *RISTI-Revista Ibérica de Sistemas e Tecnologías de Informação*, (22), 89–105.
- Parunak, H. V. D., Savit, R., & Riolo, R. L. (1998). Agent-based modeling vs. equation-based modeling: A case study and users' guide. Paper presented at the International Workshop on Multi-Agent Systems and Agent-Based Simulation, Sao Paulo, Brazil.
- Porter, M. E. (1998). *Competitive strategy : techniques for analyzing industries and competitors : with a new introduction*. New York: Free Press. Traducción a castellano-Estrategia Competitiva : Técnicas Para El análisis De Los Sectores Industriales y De La Competencia. México D.F.: Compañía Editorial Continental, 1998.

- Prahalad, C. K., & Hamel, G. (2006). *The core competence of the corporation*. Berlin: Springer.
- Rodríguez, A. S. D., Aguirre, R. C., & Ordóñez, L. S. (2015). Clima organizacional, migración tecnológica y apagón analógico en la televisión ecuatoriana. . *Suma de Negocios*, 6(14), 138–146.
- Sankararaman, S., & Mahadevan, S. (2015). Integration of model verification, validation, and calibration for uncertainty quantification in engineering systems. *Reliability Engineering & System Safety*, 138, 194–209.
- Senge, P. (1990). *The Fifth Discipline: The Art & Practice of The Learning Organization*. New York: Double Day Dell Publisher.
- Simon, H. A. (1996). *The sciences of the artificial* (Vol. 136). Cambridge, MA, USA: MIT press.
- Sousa, N. M., Costa, C. J., & Aparicio, M. (2017). Ba: um fator determinante no uso de sistemas de gestão do conhecimento. *RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação*, (22), 1–19.
- Tecuci, G. (1992). *Cooperation in knowledge base refinement*. Paper presented at the Machine Learning: Proceedings of the Ninth International Conference ML92.
- Teece, D., & Pisano, G. (1994). The dynamic capabilities of firms: an introduction. *Industrial and corporate change*, 3(3), 537–556.
- Teece, D. J. (1977). Technological transfer by multinational firms: the resource cost of transferring know-how. *Economic Journal*, 87(June), 242–261.
- Teece, D. J., Pisano, G., & Shuen, A. (1997). Dynamic capabilities and strategic management. *Strategic management journal*, 18(7), 509–533.
- Tisue, S., & Wilensky, U. (2004). *NetLogo: A Simple Environment for Modeling Complexity*. Paper presented at the International Conference on Complex Systems, May 16–21, Boston.
- Todorova, G., & Durisin, B. (2007). Absorptive capacity: valuing a reconceptualization. *Academy of management review*, 32(3), 774–786.
- Vega, J., Gutiérrez, A., & Fernández de Lucio, I. (2007). *An analytical model of absorptive capacity*. Paper presented at the DRUID Summer Conference, Copenhagen, 18–20 June.
- Volberda, H. W., Foss, N. J., & Lyles, M. A. (2009). Absorbing the Concept of Absorptive Capacity: How To Realize Its Potential in the Organization Field.
- Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling. Evanston, IL: Northwestern University.
- Winter, S. G. (2000). The satisficing principle in capability learning. *Strategic management journal*, 21(10–11), 981–996.

- Xiang, X., Kennedy, R., Madey, G., & Cabaniss, S. (2005). *Verification and validation of agent-based scientific simulation models*. Paper presented at the Agent-Directed Simulation Conference, San Diego, CA, USA.
- Zahra, S. A., & George, G. (2002). Absorptive capacity: A review, reconceptualization, and extension. *Academy of management review*, 27(2), 185–203.
- Zollo, M., & Winter, S. G. (2002). Deliberate learning and the evolution of dynamic capabilities. *Organization science*, 13(3), 339–351.

Crítérios Editoriais

A RISTI (Revista Ibérica de Sistemas e Tecnologias de Informação) é um periódico científico, propriedade da AISTI (Associação Ibérica de Sistemas e Tecnologias de Informação), que foca a investigação e a aplicação prática inovadora no domínio dos sistemas e tecnologias de informação.

O Conselho Editorial da RISTI incentiva potenciais autores a submeterem artigos originais e inovadores para avaliação pelo Conselho Científico.

A submissão de artigos para publicação na RISTI deve realizar-se de acordo com as chamadas de artigos e as instruções e normas disponibilizadas no sítio Web da revista (<http://www.risti.xyz/>).

Todos os artigos submetidos são avaliados por um conjunto de membros do Conselho Científico, não inferior a três elementos.

Em cada número da revista são publicados entre cinco a oito dos melhores artigos submetidos.

Criterios Editoriales

La RISTI (Revista Ibérica de Sistemas y Tecnologías de la Información) es un periódico científico, propiedad de la AISTI (Asociación Ibérica de Sistemas y Tecnologías de la Información), centrado en la investigación y en la aplicación práctica innovadora en el dominio de los sistemas y tecnologías de la información.

El Consejo Editorial de la RISTI incentiva autores potenciales a enviar sus artículos originales e innovadores para evaluación por el Consejo Científico.

El envío de artículos para publicación en la RISTI debe hacerse de conformidad con las llamadas de los artículos y las instrucciones y normas establecidas en el sitio Web de la revista (<http://www.risti.xyz/>).

Todos los trabajos enviados son evaluados por un número de miembros del Consejo Científico de no menos de tres elementos.

En cada número de la revista se publican cinco a ocho de los mejores artículos enviados.

Os asociados da AISTI reciben a RISTI gratuitamente, por correo postal. Torne-se asociado da AISTI. Preencha o formulário abaixo e envie-o para o e-mail aistic@gmail.com

Los asociados de la AISTI reciben la RISTI por correo, sin costo alguno. Hazte miembro de la AISTI. Rellena el siguiente formulario y remítelo al e-mail aistic@gmail.com



Formulário de Associado / Formulario de Asociado

Nome/Nombre: _____

Instituição/Institución: _____

Departamento: _____

Morada/Dirección: _____

Código Postal: _____ Localidade/Localidad: _____

País: _____

Telefone/Teléfono: _____

E-mail: _____ Web: _____

Tipo de Associado e valor da anuidade:

☐ Individual - 35€

☐ Instituição de Ensino ou I&D/Institución de Educación o I&D - 250€

☐ Outro (Empresa, etc.) - 500€

NIF/CIF: _____

Data/Fecha: __/__/____ Assinatura/Firma: _____



Revista Ibérica de Sistemas e Tecnologias de Informação
Revista Ibérica de Sistemas y Tecnologías de Información

©AISTI 2018 <http://www.aisti.eu>

