

Tendencias de la Ingeniería de Software

Impacto en las Tecnologías de Información y Comunicación

• Jezreel Mejía Miranda • Mirna A. Muñoz Mata • Alma Y. Quiñonez Carrillo • Hugo A. Mitre Hernández • José A. Mora Soto •

Tendencias en la Ingeniería de Software

Impacto en las Tecnologías de Información y Comunicación

EDITADO POR:

• Jezreel Mejía Miranda • Mirna A. Muñoz Mata • Alma Y. Quiñonez Carrillo • Hugo A. Mitre Hernández • José A. Mora Soto •

PUBLICADO POR:



Centro de Investigación
en Matemáticas, A.C.

Jalisco s/n, Valenciana, C.P. 36240

Tel. +52 (473) 732 7155,

Fax. +52 (473) 732 5749

www.cimat.mx

*Tendencias en la Ingeniería del Software.
Impacto en las Tecnologías de Información y Comunicación*

D.R. © Primera edición, 2016

D.R. © Centro de Investigación en Matemáticas, A.C.

Jalisco s/n, Valenciana
C.P. 36240, Guanajuato, Gto., México
Apdo. Postal 402
<http://www.cimat.mx>

El cuidado de la edición estuvo a cargo de Jezreel Mejía Miranda, Mirna A. Muñoz Mata, Alma Y. Quiñonez Carrillo, Hugo A. Mitre Hernández y José A. Mora Soto.

ISBN: 978-607-96212-6-1

Este libro no puede ser reproducido total ni parcialmente, por ningún medio electrónico o de otro tipo, sin autorización escrita del editor.

This book may not be reproduced, whole or in part, by any means, without written permission from the publisher.

Impreso y hecho en México.

Printed and Made in Mexico.

Agradecemos su apoyo para la realización del presente libro.



Prólogo de la serie

Este libro está dedicado a la publicación de los trabajos de investigación con aplicaciones reales y de actualidad, los cuales están relacionados a las tendencias en la Ingeniería del Software aplicados a las Tecnologías de Información y Comunicación.

Prólogo:

La Ingeniería de Software (IS) permite el establecimiento y uso de principios de ingeniería robustos, orientados a obtener productos y servicios de software de alta calidad, económico, fiable, eficiente y que satisfaga las necesidades del usuario. Por lo tanto, la IS se ha convertido sin duda una área indispensable para el desarrollo de productos y servicios orientados en las Tecnologías de Información y Comunicación (TICs) ya que éstas se han convertido en un motor que mueve a la sociedad en todos sus sectores.

En este contexto, los trabajos publicados en este libro abordan temas relacionados a: la definición y validación de requerimientos de sistemas basada en herramientas y bajo el esquema de patrones; Aspectos eurísticos para el mejoramiento de herramientas desarrolladas bajo componentes; Método de validación de software con base en el modelo CMMI; Desarrollo de software para un entorno universitario y de la utilización de nuevos componentes como la Graph API de Facebook para la difusión de información. Asimismo, se presenta un modelo para la integración de las MiPymes, Sociedad y Gobierno a través del uso de las TICs. Además, en el contexto de la educación se presenta un sistema de detección de emociones para la recomendación de recursos educativos y un modelo para implementar un sistema de gestión del conocimiento en la educación Dual. En el área de seguridad de las TICs se presenta una propuesta de contenido y controles de seguridad para un sitio web de un CSIRT. Para el sector Salud se presenta una solución alternativa para el uso de TICs específicamente en la endoscopia y finalmente, se presenta un trabajo en progreso acerca del establecimiento del estado del arte sobre marcos de trabajo, métodos y metodologías para evaluar la implementación de metodologías ágiles en Pymes.

Los trabajos publicados en este libro abordan nuevos enfoques o áreas de interés mencionados anteriormente, presentando investigaciones que abarcan tanto el entorno académico como el industrial, y que permiten dimensionar los retos a los cuales la IS se enfrenta actualmente.

Editores:

Jezreel Mejia
Mirna Muñoz
Alma Y. Quiñonez Carrillo
Hugo A. Mitre Hernández
José A. Mora Soto

Comité Científico

Adriana Peña Perez-Negron	Universidad de Guadalajara	México
Alejandro Rodríguez Gonzalez	Universidad Politecnica de Madrid	España
Alma Maria Gómez Rodríguez	Universidad de Vigo	España
Alvaro Rocha	Universidade de Coimbra	Portugal
Angel Jordan	Carnegie Mellon University	Estados Unidos
Antoni Lluís Mesquida Calafat	Universidad de las Islas Baleares	España
Antonia Mas Pichaco	Universidad de las Islas Baleares	España
Antonio Pereira	Instituto Politécnico de Leiria	Portugal
Antonio de Amescua Seco	Universidad Carlos III de Madrid	España
Arturo Méndez Penín	Universidad de Vigo	España
Carla Pacheco	Universidad Tecnológica de la Mixteca, Oaxaca	México
Carlos Lara López	CIMAT Unidad Zacatecas	México
Diego Martín de Andrés	Universidad Carlos III de Madrid	España
Edrisi Muñoz Mata	CIMAT Unidad Zacatecas, México	México
Edwin León Cardenal	CIMAT Unidad Zacatecas	México
Elisabet Cápon	Swiss Federal Institute of Technology, Zürich (ETHZ)	Suiza
Enrique Barreiro Alonso	Universidad de Vigo	España
Fernando Moreina	Universidade Portucalense	Portugal
Francisco Ortín Soler	Universidad de Oviedo	España
Giner Alor Hernandez	Instituto Tecnológico de Orizaba	México
Gloria P. Gasca Hurtado	Universidad de Medellín	Colombia
Gonzalo Cuevas Agustín	Universidad Politécnica de Madrid	España
Gonzalo R. Luzardo Morocho	Escuela Superior Politécnica del Litoral	Ecuador
Gustavo Illescas	Universidad Nacional del Centro de la Provincia de Buenos Aires	Argentina
Henrique Mamede	Universidade Aberta de Lisboa	Portugal
Hugo Arnoldo Mitre	CIMAT Unidad Zacatecas, México	México
Ivan A. Garcia Pacheco	Universidad Tecnológica de la Mixteca, Oaxaca	México
Jaime A. Guzmán Luna	Universidad Nacional de Colombia	Colombia
Javier García Guzman	Universidad Carlos III de Madrid	España
Joao Barroso	Universidad Tras-os-Montes e Alto Douro	Portugal
Jörg Thomaschewski	Universidad de Vigo -Hochschule Emden-Leer	España
Jose A. Calvo Manzano Villalon	Universidad Politécnica de Madrid	España
Jose A. Mora Soto	CIMAT Unidad Zacatecas	México
Jose Antonio Cerrada Somolinos	Universidad Nacional de Educación a Distancia	España
Jose Baltasar García Perez-Schofield	Universidad de Vigo	España
Juan Carlos Gonzáles Moreno	Universidad de Vigo	España
Juan Manuel Cueva Lovelle	Universidad de Vigo	España
Leandro Rodríguez Linares	Universidad de Vigo	España
Leandro Rodríguez Linares	Universidad de Vigo	España
Luis Casillas	CUCEI de la Universidad de Guadalajara	México
Luis Fernández Sanz	Universidad de Alcalá	España
Luis Borges Goveia	Universidad Fernando Pessoa	Portugal
Luis J. Domínguez Pérez	CIMAT Unidad Zacatecas	México
Magdalena Arcilla Cobián	Universidad Nacional de Educación a Distancia	España
Manuel Pérez Cota	Universidad de Vigo	España
María José Lado Touriño	Universidad de Vigo	España
María Y. Hernández Pérez	Instituto de Investigaciones Eléctricas	México
Marion Lepmets	Regulated Software Research Centre, Dundalk Institute of Technology, Ireland	Estonia
Omar S. Gómez	Escuela Superior Politécnica de Chimborazo	Ecuador
Paul Clarke	Dundalk Institute of Technology	Irlanda
Perla Velasco-Elizondo	Universidad Autonoma de Zacatecas (UAZ)	México
Rafael Valencia García	Universidad de Murcia	España

Ramiro Goncalves	Universidade Trás-os-Montes	Portugal
Raúl Aguilar Vera	Universidad Autónoma de Yucatán	México
Ricardo Colomo Palacios	Østfold University College	Noruega
Rory O'Connor	Dublin City University	Irlanda
Santiago Mataloaga	Universidad ORT Uruguay	Uruguay
Sodel Vázquez Reyes	Universidad Autónoma de Zacatecas (UAZ)	México
Sulema Torres Ramos	CUCEI de la Universidad de Guadalajara	México
Tomas San Feliu Gilabert	Universidad Politécnica de Madrid	España
Ulises Juárez Martínez	Instituto Tecnológico de Orizaba	México
Ulrik Brandes	Universidad de Konstanz	Alemania
Valentine Casey	Dundalk Institute of Technology	Irlanda
Vianca Vega	Universidad Católica del Norte	Chile
Victor Saquicela	Universidad de Cuenca	Ecuador
Vítor Filipe	Universidade Trás-os-Montes e Alto Douro	Portugal
Yadira Quiñonez	Universidad Autónoma de Sinaloa	México
Yilmaz Murat	Çankaya University	Turquía

Índice general

Herramienta para administración y validación de requerimientos de sistemas	7
Definición de Requerimientos de Software a Través de Escenarios utilizando Patrones	15
Proceso basado en patrones de proceso para desarrollar aplicaciones Web	22
Evaluación Heurística de Herramientas de Composición de Componentes Centrada en el Usuario Final	32
Metamodelo para validación de software basado en el CMMI	41
Presupuesto-CUC y SIRU, dos Experiencias de Desarrollo De Software A La Medida Para Fortalecer La Gestión De La Universidad De La Costa: Presentación De Resultados, Impactos, Dificultades y Recomendaciones.	49
Propuesta de un modelo para la Implementación de un Sistema de Gestión del Conocimiento en la Educación Dual	55
Propuesta de un modelo para la integración de las MiPyMES, Sociedad y Gobierno a través del uso de las TIC de la Zona Metropolitana del Valle de Orizaba, Veracruz	62
Sistema de detección de emociones para la recomendación de recursos educativos	68
Propuesta de contenido y controles de seguridad para un sitio web de un CSIRT	75
Forming an alternate solution for endoscopy	83
Establishing the state of art of frameworks, methods and methodologies to assess the implementation and use of agile methodologies in SMEs: A systematic review	90

Evaluación Heurística de Herramientas de Composición de Componentes Centrada en el Usuario Final

Yadira Jarvio Hernández¹, Perla Velasco-Elizondo² y Edgard Benítez-Guerrero¹

¹ Universidad Veracruzana, Facultad de Estadística e Informática, Xalapa, VER, 91020, México.

² Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería, Ciudad Universitaria Siglo XXI, Zacatecas, ZAC, 98000, México.

zs13015663@estudiantes.uv.mx, pvelasco@uaz.edu.mx, edgardibenez@gmail.com

Resumen. Cada vez más personas realizan diversas tareas apoyándose de sistemas de software que son construidos a partir de la composición de componentes. A pesar de la popularidad del desarrollo centrado en composición y las herramientas existentes para soportarlo hay evidencia sobre que el uso de éstas sigue siendo complicado. Sin embargo, existe poca evidencia que permita comprender esta problemática en un contexto de usuarios finales. En este trabajo se realiza una evaluación heurística de herramientas de composición de componentes centrada en el usuario final. Una contribución importante de esta evaluación es que considerando un conjunto de aspectos intrínsecos al desarrollo centrado en composición y a los usuarios finales, se identifican un conjunto de heurísticas relevantes para detectar problemas generales de usabilidad. Las heurísticas y los problemas identificados proveen una base de conocimiento que puede ser extendida en dominios específicos de aplicación para mejorar el diseño de herramientas de composición.

Keywords: evaluación heurística, composición de componentes, usuarios finales.

1 Introducción

Muchas personas realizan diversas tareas apoyándose de sistemas de software que son construidos a partir de la composición de *componentes*. En términos generales, un componente es un elemento de software pre-existente que implementa alguna funcionalidad, la cual puede ser accedida a través de interfaces bien definidas (Sommerville, 2006). Los COTS (Commercial Off-The-Shelf) (Morisio, Seaman, Basili, Parra, Kraft, & Condon, 2002), los Servicios Web (Cauldwell, Chawla, & Chopra, 2002), los Mashlets (Abiteboul, Greenshpan, & Milo, 2008) o las (Web) APIs son ejemplos de componentes en estos sistemas.

Enfoques de desarrollo como el Desarrollo Basado en Componentes (Sommerville, 2006), las Líneas de Productos de Software (Clements & Northrop, 2001), la Arquitectura Orientada a Servicios (Erl, 2005) o más recientemente las arquitecturas de micro servicios (Fowler, 2014) han adquirido popularidad puesto que soportan el desarrollo de sistemas de software usando un enfoque centrado en la composición. Actualmente existen diversas herramientas y (repositorios de) componentes en dominios específicos que facilitan el desarrollo de sistemas bajo este enfoque.

La disponibilidad de herramientas y (repositorios de) componentes ha contribuido a que nuevas comunidades de usuarios incursionen en el desarrollo de este tipo de sistemas. Un ejemplo es la comunidad de usuarios finales (Mehandjiev, Namoune, Wajid, Macaulay, & Sutcliffe, 2010), (Garlan, Dwivedi, Ruchkin, & Schmerl, 2012), (Roy Chowdhury, 2012). En términos generales este tipo de usuarios y los sistemas que construyen presentan las siguientes características:

- Son inexpertos en materia de desarrollo de sistemas y, generalmente, expertos en algún dominio profesional.
- Soportan sus tareas diarias mediante sistemas de software que ellos mismos construyen a partir de la composición de componentes de software que son provistos por terceras partes.
- Los sistemas que construyen tienen una arquitectura “data-flow” (Taylor, Medvidovic, & Dashofy, 2009).

Ejemplos de estos usuarios incluyen a sociólogos realizando análisis de influencia en redes sociales (Knoke & Yang, 2008) con scripts que usan las APIs de Twitter (Twitter, 2015) e igraph para R (Team, 2015) o biólogos realizando análisis de cadenas de proteínas con workflows creados con la herramienta de composición Taverna (School of Computer Science, 2010) y servicios Web en BioCatalogue (Bhagat, et al., 2010).

A pesar de la popularidad del desarrollo centrado en composición y las herramientas existentes para soportarlo hay evidencia sobre que el uso de éstas sigue siendo complicado. Sin embargo, existe muy poco material que permita comprender esta problemática en un contexto de usuarios finales. En este trabajo se describe una evaluación heurística de herramientas de composición de componentes para

detectar problemas generales de usabilidad relevantes al contexto de un usuario final. En contraste con trabajos relacionados, una contribución importante de esta evaluación es que se usaron heurísticas que fueron identificadas considerando un conjunto de aspectos intrínsecos al desarrollo centrado en composición y a los usuarios finales. Las heurísticas y los problemas identificados proveen una base de conocimiento inicial que puede ser extendido en dominios específicos para mejorar el diseño de herramientas de composición.

El artículo está organizado como sigue. La sección 2 discute trabajos relacionados al tema de este artículo. En la sección 3 se describen los detalles de la evaluación realizada. En la sección 4 se muestran los resultados obtenidos de la evaluación. Finalmente, en la sección 5 se presentan las conclusiones y posibles líneas de trabajo futuro.

2 Trabajos Relacionados

En el contexto de la Ingeniería de Software la *usabilidad* es un atributo de calidad que tiene que ver con la facilidad de uso de un sistema de software. Existen diversas definiciones de usabilidad. Sin embargo, dos ampliamente adoptadas son las provistas por los estándares 9241 y 9216 de la ISO/IEC. El estándar ISO/IEC 9241 define usabilidad como “el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso” (ISO/IEC 9241-11, 1998). Similarmente, en el estándar ISO/IEC 9126 la usabilidad se define como “la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso” (ISO/IEC 9126, 1992).

La usabilidad se evalúa considerando un conjunto de características. No hay un consenso sobre el conjunto de características que deben ser consideradas. Sin embargo, características como eficiencia, facilidad de aprendizaje, reconocimiento antes que recuerdo, manejo de errores y satisfacción subjetiva son comunes en varios estándares y modelos de calidad (Seffah, Donyaee, B. Kline, & K. Padda, 2006).

En el pasado se han realizado evaluaciones relacionadas al desarrollo centrado en composición. Por ejemplo, evaluaciones de métodos y lenguajes para realizar la composición de componentes como BPEL, OWL-S, autómatas, redes de Petri, Álgebras de Procesos, (Beek, Bucchiarone, & Gnesi, 2006), (ter Beek, Bucchiarone, & Gnesi, 2007), (Feenstra, Janssen, & Wagenaar, 2007), (Baryannis & Plexousakis, 2010), (Portchelvi, Prasanna Venkatesan, & Shanmugasundaram, 2012). Igualmente se han reportado resultados de evaluaciones de herramientas de composición en trabajos como (Minhas, Sampaio, & Mehandjiev, 2012), (Insfran, Cedillo, Fernández, Abrahão, & Matera, 2012), (Yeltayeva, 2012). Todas estas evaluaciones reportan problemas relevantes. Sin embargo, la principal limitación de estas evaluaciones es que no consideran a los usuarios finales como usuarios potenciales de estas herramientas.

Por otra parte trabajos como (Zhao, Bhattarai, Liu, & Crespi, 2011), (Malinga, Gruner, & Koschmider, 2013) reportan resultados de evaluaciones a herramientas de composición para usuarios finales. Aunque se reportan problemas relevantes, estas evaluaciones presentan dos limitaciones importantes: son evaluaciones de una sola herramienta lo cual hace difícil el determinar si los problemas detectados se presentan en otras y, las evaluaciones no consideran ambos aspectos intrínsecos al desarrollo centrado en composición y a los usuarios finales. La Tabla 1 describe estos aspectos.

Tabla 1. Aspectos intrínsecos al desarrollo centrado en composición y a los usuarios finales, que impactan en la usabilidad de herramientas de composición de componentes cuando son utilizadas por usuarios finales.

Aspecto	Descripción
Selección de componentes disponibles	Existen muchos componentes con las mismas características. Al momento de realizar una composición se debe realizar la selección de un subconjunto particular que satisfaga las características funcionales requeridas, e.g., traducir a español. Idealmente esta selección debería ser asistida.
Detección de incompatibilidades entre componentes	Los componentes son desarrollados por diferentes fabricantes y podrían presentar incompatibilidades al momento de su composición, e.g., formato o semántica de los datos que intercambian. Idealmente, estas incompatibilidades deberían ser detectadas y resueltas automáticamente durante el proceso de composición.
Estimación de calidad de servicio de la composición	Cada componente define individualmente valores de atributos de calidad de servicio, e.g., desempeño, disponibilidad. La calidad servicio del sistema resultante es relevante e, idealmente, debería ser estimada automáticamente durante el proceso de composición.
Uso de un lenguaje de bajo nivel de abstracción	El usuario final no tiene conocimiento técnico profundo sobre desarrollo de sistemas. Por ello, se preferiría en las herramientas de composición el uso de un lenguaje poco técnico y con un alto nivel de abstracción.

3 Descripción de la Evaluación

En las siguientes secciones se describen los detalles de la evaluación.

3.1 Selección de Herramientas

La selección de herramientas de composición se inició con una búsqueda en fuentes bibliográficas y sitios web. Producto de esta búsqueda se obtuvo información sobre 23 herramientas. Posteriormente, se realizó un análisis de esta información para conservar solamente las herramientas que soportaran lo siguiente:¹

1. *Acceso a servicios pre-existentes* (SP). Que permitan acceder a componentes pre-existentes provistos por terceras partes; esto es porque generalmente los usuarios finales no desarrollan sus propios componentes y tienden a utilizar componentes pre-existentes.
2. *Facilidades “drag and drop”* (DD). Que provean un ambiente gráfico de composición de componentes con facilidades “drag and drop”; esto es porque se considera que este tipo de facilidades permite de forma práctica y poco técnica seleccionar y ensamblar componentes.
3. *Composiciones con una arquitectura “data-flow” en la forma de “workflows” o “mashups”* (WM). Que permitan construir composiciones de este tipo; esto debido a que generalmente los usuarios finales construyen sistemas de este tipo.
4. *Disponibilidad de la herramienta* (DI). Que se encuentren disponibles actualmente y no sean propietarias.

La Tabla 2 muestra las 23 herramientas indicándose el soporte o no soporte de los criterios listados anteriormente con los símbolos ✓ y × respectivamente. Como se puede apreciar las únicas herramientas que cumplen con los criterios son las siguientes: 1. Apache ODE, 3. BonitaSoft, 12. LONI Pipeline, 20. Taverna, 21. Wave Maker, 22. Yahoo Pipes!

Tabla 2. Herramientas consideradas en la evaluación.

#	Nombre	Características			
		S D	D D	W D	D I
1	Apache ODE, (Apache, 205).	✓	✓	✓	✓
2	Apatar, (Apatar, 2014).	×	✓	✓	✓
3	BonitaSoft, (Bonitasoft, 2015).	✓	✓	✓	✓
4	Dapper, (Al Sarraj, 2014).	✓	×	✓	×
5	Enhydra Shark, (Together Teamsolutions Co., 2011).	✓	×	✓	✓
6	Google Mashup, (Al Sarraj, 2014).	✓	×	✓	×
7	Intel Mash Maker, (Al Sarraj, 2014).	✓	×	×	×
8	Jaw Flow, (Garcês, De Jesus, Cardoso, & Valente, 2009).	✓	×	✓	×
9	JBoss, (RedHat, 2014).	✓	×	✓	✓
10	JackBe, (Al Sarraj, 2014).	✓	✓	✓	×
11	JOpera, (JOpera.org, 2013).	✓	×	✓	✓
12	LONI Pipeline, (Laboratory, 2015).	✓	✓	✓	✓
13	Marmite, (Patel , Na , Latih, Wills, Shukur , & Mull, 2010).	✓	✓	×	×
14	Microsoft Popfly, (Al Sarraj, 2014).	✓	✓	✓	×
15	OpenKapow, (Kapow, 2015).	✓	✓	✓	×
16	Potluck, (Al Sarraj, 2014).	×	×	✓	×
17	QedWiki, (Patel , Na , Latih, Wills, Shukur , & Mull, 2010).	✓	×	✓	×
18	RSSBus, (RSSBus, 2014).	×	×	✓	×
19	RUNA WFE, (RunaWFE, 2013).	×	×	✓	✓
20	Taverna, (School of Computer Science, 2010).	✓	✓	✓	✓
21	Wave Maker, (Muñoz Jiménez, 2010), (WaveMaker, 2014).	✓	✓	✓	✓
22	Yahoo Pipes, (Yahoo!, 2014).	✓	✓	✓	✓
23	YAWL, (Garcês, De Jesus, Cardoso, & Valente, 2009).	✓	×	✓	×

¹ Estos criterios fueron establecidos por los autores considerando principalmente las características de los usuarios finales descritas en la Sección 1.

3.2 Composición Realizada

Para evaluar estas herramientas se consideró un sistema, creado a partir de la composición de componentes de software, que debe automatizar lo siguiente: Se desea obtener una lista de comentarios de Twitter. Dichos comentarios deben estar ordenados por fecha de forma ascendente y traducidos al español. Además se necesita que la composición se ejecute en un tiempo aproximado de 900-1000 milisegundos y que tenga una disponibilidad de al menos 95 %. Para poder crear ese sistema, se utilizaron seis web APIs provistas por diferentes fabricantes y las seis herramientas seleccionadas.

3.3 Método de Evaluación

Considerando nuestro objetivo de identificar un conjunto de problemas generales de usabilidad en herramientas de composición de componentes que sean relevantes al contexto de usuarios finales, se escogió de entre otros métodos la evaluación heurística (Nielsen & Molich, 1990). En este tipo de evaluación un grupo de expertos inspecciona un producto de software con el fin de determinar si se adhiere o no a un conjunto de principios o heurísticas de diseño. Existen evidencias sobre que bastan entre tres y cinco expertos para encontrar, aproximadamente, el 75% de los problemas de usabilidad (González, Lorés, & Pascual, 2006).

La evaluación fue realizada por cinco evaluadores, cuatro de ellos Licenciados en Informática y actualmente alumnos del cuarto semestre un programa de Maestría en Sistemas Interactivos Centrados en el Usuario. El quinto evaluador es Licenciado en Informática y Maestro en Redes y Sistemas Integrados y cuenta con 3 años de experiencia laboral en el desarrollo de sistemas basados en componentes.

Existen varias propuestas de heurísticas a utilizar. En la Tabla 3 se listan algunas de las más relevantes en la literatura: las propuestas por Nielsen (Nielsen J. , 1994), Tognazzini (Tognazzini, 2014) y, Shneiderman y Plaisant (Shneiderman & Plaisant, 2010). Para facilitar su comprensión se han organizado en categorías. Teniendo en cuenta los aspectos que afectan la usabilidad en el contexto de los usuarios objeto de estudio y la tarea realizada, se indican con letras negritas cursivas cuáles de estas heurísticas han sido consideradas en esta evaluación. A continuación se describe esta selección.

Tabla 3. Conjunto de heurísticas y principios seleccionados para la evaluación.

Categoría	Descripción		
	Nielsen	Tognazzini	Shneiderman y Plaisant
Retroalimentación	Visibilidad del estado del sistema	Navegación visible <i>Anticipación</i>	Retroalimentación informativa
Control sobre el ambiente	Control y libertad del usuario	Eficiencia del usuario Autonomía Interfaces explorables	Revertir acciones Control por parte del usuario
Manejo de errores	<i>Prevención de errores</i> <i>Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores</i>		<i>Manejo de errores</i> <i>Manejo de errores</i>
Aprendizaje	<i>Relación entre el sistema y el mundo real</i> <i>Reconocimiento antes que recuerdo</i> Flexibilidad y eficiencia de uso Consistencia y estándares Ayuda y documentación	Valores por defecto <i>Aprendizaje</i> Uso de metáforas Objetos humanos <i>Aprendizaje</i> Uso de metáforas Objetos humanos Consistencia	Reducir la carga de memoria de corto plazo Uso de atajos Consistencia
Otros	Estética y diseño minimalista	Valores por defecto Daltonismo Ley de Fitts Reducción de la latencia Legibilidad Guardar el estado Protege el trabajo del usuario	

El principio de anticipación, definido por Tognazzini, hace referencia a que los sistemas deberían anticiparse a las necesidades y deseos del usuario. Se debe evitar que los usuarios tengan que buscar o recordar mucha información durante el uso del sistema y, en cambio, mostrarle la información y herramientas necesarias para realizar su trabajo. De esta forma, la consideración de este principio en el diseño de las herramientas de composición facilitaría a un usuario final realizar la selección de los componentes que satisfagan no solo las características funcionales requeridas (aspecto: selección de componentes disponibles en la Tabla 1), sino también los que satisfagan la calidad servicio de la composición requeridas para la composición (aspecto: estimación de calidad de servicio de la composición en la Tabla 1).

Como se observa en la Tabla 3, en la categoría manejo de errores hemos agrupado heurísticas y principios, propuestos por los tres autores, relacionados con este tema. En todos estos se comparte la visión de que el sistema debe ser capaz de reconocer, diagnosticar y recuperarse de errores. Así, el considerar las heurísticas y principios en esta categoría en el diseño de las herramientas de composición evitaría que el usuario final tuviera que detectar, diagnosticar y resolver manualmente incompatibilidades entre los componentes que va a utilizar (aspecto: detección de incompatibilidades entre componentes en la Tabla 1).

Heurísticas como relación entre el sistema y el mundo real y reconocimiento antes que recuerdo, propuestas por Nielsen, establecen que el sistema debe hablar el lenguaje de los usuarios prefiriendo los conceptos y frases familiares a éstos. Por otra parte el principio de aprendizaje de Tognazzini indica que se debe reducir la curva de aprendizaje del uso sistema. De esta forma, estas heurísticas y principios en el diseño de las herramientas de composición contribuye a mejorar el nivel abstracción del lenguaje utilizado (aspecto: uso de un lenguaje de bajo nivel de abstracción en la Tabla 1).

La Tabla 4 define las sub-heurísticas que se consideraron para la evaluación del conjunto de heurísticas y principios presentados en la Tabla 3. Estas sub-heurísticas se seleccionaron de listas predefinidas --e.g., (Nielsen J. , 1994), (Shneiderman & Plaisant, 2010), (Tognazzini, 2014). Algunas sub-heurísticas fueron adaptadas para corresponder mejor al contexto de los usuarios objeto de estudio, la composición realizada y las herramientas evaluadas.

Tabla 4. Conjunto de sub-heurísticas seleccionadas para la evaluación.

Aspecto	Heurísticas y Principios	Sub-heurísticas
Selección de componentes disponibles	Anticipación	A1. Al inicio del proceso de composición el usuario puede especificar los aspectos funcionales y de calidad del servicio que espera de la composición.
Estimación de calidad de servicio de la composición		A2. Durante el proceso de composición se sugiere al usuario los componentes que satisfacen la funcionalidad y calidad del servicio.
Detección de incompatibilidades entre componentes.	Prevención de errores	E1. Durante el proceso de composición se previene que los usuarios cometan errores.
	Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores	E2. Durante el proceso de composición, si se detectan errores se informan las posibles causas.
	Manejo de Errores	E3. Durante el proceso de composición, si se detectan errores se sugieren posibles soluciones para resolverlos.
		E4. Los mensajes de error, e información relacionada, se comunican usando un lenguaje poco técnico.
Uso de un lenguaje de bajo nivel de abstracción	Relación entre el sistema y el mundo real	L1. Los íconos son claros y usan un lenguaje visual intuitivo y poco técnico.
	Reconocimiento antes que recuerdo	L2. El lenguaje de composición es claro y poco técnico.
	Aprendizaje	

4 Resultados Obtenidos

En la Tabla 5 y la Tabla 6 Tabla 5. Resultados de la evaluación heurística - I. se muestran los resultados obtenidos de la evaluación heurística de usabilidad de las herramientas de composición seleccionadas, los cuales deben ser interpretados considerando lo siguiente.

Valor que representa la frecuencia con la que se incurre en el error:

- 0 = nunca.
- 1 = a veces.
- 2 = muchas veces.

Valor que representa el impacto asociado a cada sub-heurística:

- 0 = No es problema de usabilidad.
- 1 = Problema sin importancia, no es necesario arreglarlo a menos que haya tiempo.
- 2 = Problema de poca importancia, arreglarlo no tiene mucha importancia.
- 3 = Problema grave, es importante arreglarlo.
- 4 = Problema catastrófico, es vital arreglarlo.

El valor mostrado en las columnas E1, E2, E3, E4, E5 indica el nivel de severidad del error detectado por cada experto. El nivel de severidad se calcula multiplicando los valores de frecuencia e impacto descritos antes. Este nivel indica qué tan grave es el error detectado en la evaluación y se interpreta como sigue:

- 0 = No es un error.
- 1 - 2 = Error sin importancia.
- 3 - 4 = Error de grado medio.
- 5 - 6 = Error grave.
- 7 - 8 = Error catastrófico

Tabla 5. Resultados de la evaluación heurística - I.

Sub- heurísticas	Apache ODE						BonitaSoft						LONI Pipeline					
	E1	E2	E3	E4	E5	R	E1	E2	E3	E4	E5	R	E1	E2	E3	E4	E5	R
A1	8	6	8	6	6	6.8	8	6	8	2	6	6	8	6	8	1	8	6.2
A2	8	6	8	3	6	6.2	6	6	6	1	6	5	8	1	8	2	6	5
E1	6	3	6	3	8	5.2	3	4	4	3	6	4	6	1	3	2	6	3.6
E2	6	3	6	8	6	5.8	6	6	6	3	6	5.4	6	1	6	3	6	4.4
E3	8	6	8	8	8	7.6	6	8	8	8	6	7.2	6	2	8	6	6	5.6
E4	8	8	8	8	8	8	8	3	8	8	6	6.6	6	3	8	3	6	5.2
L1	8	8	8	6	6	7.2	8	3	6	1	6	4.8	6	2	3	2	6	3.8
L2	8	6	8	3	8	6.6	8	2	8	2	6	5.2	6	2	3	3	6	4

Tabla 6. Resultados de la evaluación heurística - II.

Sub- heurísticas	Taverna						Wave Maker						Yahoo Pipes!					
	E1	E2	E3	E4	E5	R	E1	E2	E3	E4	E5	R	E1	E2	E3	E4	E5	R
A1	8	6	8	3	6	6.2	8	6	8	2	6	6	8	6	8	2	6	6
A2	8	6	8	3	6	6.2	8	6	8	2	6	6	6	6	8	3	6	5.8
E1	6	6	2	3	6	4.6	8	3	8	4	6	5.8	6	2	8	4	6	5.2
E2	6	3	6	6	6	5.4	6	2	8	3	6	5	6	2	8	6	6	5.6
E3	8	8	8	4	6	6.8	8	4	8	6	6	6.4	8	4	8	6	6	6.4
E4	8	6	6	6	6	6.4	8	8	8	6	6	7.2	8	3	6	8	6	6.2
L1	8	8	8	3	6	6.6	8	3	8	2	6	5.4	8	2	8	6	6	6
L2	8	3	8	2	6	5.4	8	8	8	2	6	6.4	8	2	8	6	6	6

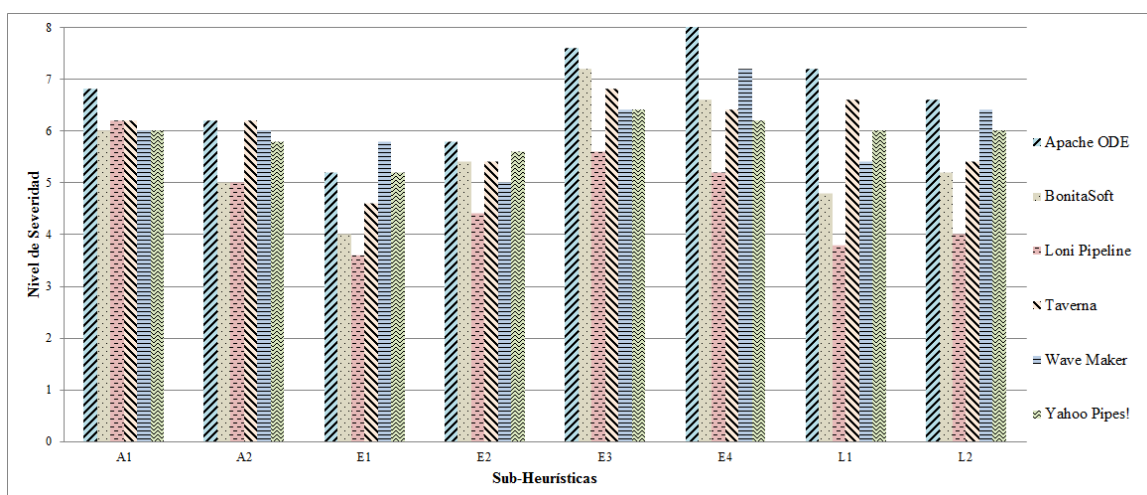


Fig. 1. Resultados obtenidos de la evaluación heurística.

En la Tabla 5 y la Tabla 6 la columna “R” es el promedio obtenido de los valores de los cinco expertos. Este promedio está graficado en la Fig. 1 y se observa lo siguiente.

La anticipación es un problema que va de grave a catastrófico (valores de 5 a 7), debido a que ninguna de las herramientas ofrece soporte para especificar, al inicio del proceso de composición, los aspectos funcionales y de calidad del servicio que se espera del sistema a construir (sub-heurística A1). Como consecuencia de lo anterior, ninguna de las herramientas evaluadas ofrece soporte para realizar la selección de componentes considerando los aspectos funcionales y de calidad de servicio requeridos (sub-heurística A2). El usuario final tiene siempre que resolver todo esto de forma manual.

En lo que se refiere al manejo de errores se detectó que es un problema que va de grado medio a catastrófico (valores de 4 a 8), ya que algunas herramientas brindan cierto soporte para prevenir errores (sub-heurística E1), sin embargo cuando se presentan errores, los mensajes de error no brindan información de la causa y tampoco de cómo resolver el error presentado (sub-heurísticas E2 y E3), además el mensaje se comunican al usuario usando un lenguaje técnico (sub-heurística E4).

Finalmente, el uso de un lenguaje de bajo nivel en la composición es un problema que va desde grado medio hasta catastrófico (valores de 4 a 7), esto porque todas las herramientas utilizan íconos poco intuitivos y un lenguaje muy técnico para los usuarios finales (sub-heurísticas L1 y L2), sin embargo algunas tienen un lenguaje que puede ser más técnico que el utilizado en otras y dado que los evaluadores tienen conocimientos en desarrollo de sistemas, no calificaron el lenguaje utilizado como un lenguaje muy técnico.

5 Conclusiones y Trabajo Futuro

En este trabajo se presentó una evaluación heurística de herramientas de composición de componentes para detectar problemas generales de usabilidad relevantes al contexto de un usuario final. En contraste con algunos trabajos relacionados, una contribución importante de la evaluación presentada es que se usaron heurísticas que fueron identificadas considerando un conjunto de aspectos intrínsecos al desarrollo centrado en composición y a los usuarios finales.

Los problemas identificados con nivel de severidad alto y que ocurren frecuentemente incluyen: el uso de un lenguaje técnico, el poco soporte ofrecido para la especificación de aspectos funcionales y de calidad del servicio y, el poco soporte ofrecido para la selección de componentes basada en dichas especificaciones.

Aunque estos problemas son generales, consideramos que es importante conocerlos pues proveen un conjunto inicial que puede ser considerado para, como trabajo futuro, mejorar las muchas herramientas de composición actuales. Igualmente estos problemas proveen la base para realizar futuras evaluaciones considerando contextos más específicos a determinados tipos de usuarios finales --e.g., sociólogos, biólogos.

Con base en los resultados obtenidos, consideramos que la evaluación heurística fue un método apropiado para detectar problemas de usabilidad generales. Sin embargo, en dominios de aplicación específicos sería necesario utilizar otras técnicas de evaluación; preferentemente aquellas que involucren a usuarios finales.

Agradecimientos

Los autores desean reconocer a los revisores anónimos de este artículo por sus útiles sugerencias. La primera autora agradece a CONACYT-Mexico el apoyo para la realización de sus estudios de Maestría en Sistemas Interactivos Centrados en el Usuario (No. de becario: 297377).

Referencias

- Abiteboul, S., Greenspan, O., & Milo, T. (2008). Modeling the mashup space. Proceedings of the 10th ACM workshop on Web information and data management (pp. 87--94). California, USA: ACM.
- Al Sarraj, W. (25 de October de 2014). Toward Usability Evaluation Criteria for Web Mashup Makers for End-Users. *International Journal of Advanced Computer Technology (IJACT)*, 3, 23--32.
- Apache, S. F. (2015). Apache ODE. Obtenido de <http://ode.apache.org/>
- Apatar, I. (2014). Apatar. Connecting data. Obtenido de <http://www.apatar.com/>
- Baryannis, G., & Plexousakis, D. (2010). Automated Web Service Composition: State of the Art and Research Challenges. Technical Report ICS-FORTH/TR-409. Foundation for Research & Technology - Hellas - Institute of Computer Science.
- Beek, M., Bucchiarone, A., & Gnesi, S. (2006). A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods. In Technical Report 2006TR-15, Istituto. IEEE CS Press.
- Bhagat, J., Tanoh, F., Nzuobontane, E., Laurent, T., Orłowski, J., Roos, M., . . . Globe, C. A. (10-15 de Septiembre de 2010). BioCatalogue: a universal catalogue of web services for the life sciences. Obtenido de <https://www.biocatalogue.org/>
- Bonitasoft, I. (2015). Bonitasoft. Obtenido de <http://www.bonitasoft.com/>
- Cauldwell, P., Chawla, R., & Chopra, V. (2002). Servicios Web XML. España: Anaya Multimedia-Anaya Interactiva.
- Clements, P., & Northrop, L. (2001). Software Product Lines: Practices and Patterns. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Erl, T. (2005). Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall.
- Feenstra, R. W., Janssen, M., & Wagenaar, R. W. (2007). Evaluating Web Service Composition Methods: the Need for Including Multi-Actor Elements. *Electronic Journal of e-Government*, 5(2), 153--163.
- Fowler, M. (2014). Microservices. Obtenido de <http://martinfowler.com/articles/microservices.html>
- Garcês, R., De Jesus, T., Cardoso, J., & Valente, P. (2009). Open Source Workflow Management Systems: A Concise Survey. En *GWDL: A graphical workflow definition language for business workflows*. Springer Berlin Heidelberg.
- Garlan, D., Dwivedi, V., Ruchkin, I., & Schmerl, B. (2012). Foundations and Tools for End-user Architecting. Proceedings of the 17th Monterey Conference on Large-Scale Complex IT Systems: Development, Operation and Management (pp. 157--182). Oxford, UK: Springer-Verlag.
- González, M. P., Lorés, J., & Pascual, A. (2006). Evaluación Heurística. En *La Interacción Persona Ordenador* (pp. 1-40). AIPO Press.
- Insfran, E., Cedillo, P., Fernández, A., Abrahão, S., & Matera, M. (2012). Evaluating the Usability of Mashups Applications. *Quality of Information and Communications Technology (QUATIC)*, 2012 Eighth International Conference on the (pp. 323--326). Lisbon: IEEE.
- ISO/IEC 9126. (1992). Information technology-Software product evaluation-Quality characteristics and guidelines for their use.
- ISO/IEC 9241-11. (1998). ISO/IEC, 9241-11. Ergonomic requirements for office work with visual display terminals (VDT)s - Part 11 Guidance on usability.
- JOpera.org. (2013). JOpera. Process Support for Web Services. Obtenido de <http://www.jopera.org/>
- Kapow. (2015). Kapow Software. A Kofax Company. Obtenido de <http://www.kapowtech.com/>
- Knoke, D., & Yang, S. (2008). Social Network Analysis. SAGE Publications, Inc.
- Laboratory, o. N. (2015). Loni Pipeline. Obtenido de <http://pipeline.bmap.ucla.edu/>
- Malinga, M., Gruner, S., & Koschmider, A. (2013). Quality and Usability of Mashup Tools: Criteria and Evaluation. Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (pp. 154--159). East London, South Africa: ACM.
- Mehandjiev, N., Namoune, A., Wajid, U., Macaulay, L., & Sutcliffe, A. (2010). End user service composition: Perceptions and requirements. *Web Services (ECOWS)*, 2010 IEEE 8th European Conference on (pp. 139--146). Ayia Napa: IEEE.
- Minhas, S. S., Sampaio, P., & Mehandjiev, N. (2012). A Framework for the Evaluation of Mashup Tools. *Services Computing (SCC)*, 2012 IEEE Ninth International Conference on (pp. 431--438). Honolulu, HI: IEEE.
- Morisio, M., Seaman, C., Basili, V., Parra, A., Kraft, S., & Condon, S. E. (2002). COTS-based software development: processes and open issues. *Systems and Software*, 189--190.
- Muñoz Jiménez, J. A. (May de 2010). Programación con AJAX. Obtenido de <http://jamj.webcindario.com/programacion/ajax/AJAX.pdf>
- Nielsen, J. (1994). Enhancing the Explanatory Power of Usability Heuristics. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 152--158). Boston, Massachusetts, USA: ACM.
- Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *SIGCHI Conference on Human Factors in Computing Systems* (pp. 249--256). ACM.
- Patel, A., Na, L., Latih, R., Wills, C., Shukur, Z., & Mull, R. (2 de November de 2010). A Study of Mashup as a Software Application Development Technique with . *Journal of Computer Science*, 6(12), 1406.

- Portchelvi, V., Prasanna Venkatesan, V., & Shanmugasundaram, G. (2012). Achieving Web Services Composition - a Survey. 2, pp. 195--202. Software Engineering.
- RedHat. (2014). JBoss Developer. Obtenido de <http://www.jboss.org/>
- Roy Chowdhury, S. (2012). Assisting End-user Development in Browser-based Mashup Tools. Proceedings of the 34th International Conference on Software Engineering (pp. 1625--1627). Zurich, Switzerland: IEEE Press.
- RSSBus, I. (2014). RSSBus. Obtenido de <http://www.rssbus.com/>
- RunaWFE. (2013). RunaWFE. Obtenido de <http://wf.runa.ru/>
- School of Computer Science, U. o. (01-03 de Noviembre de 2010). Taverna. Obtenido de <http://www.taverna.org.uk/>
- Seffah, A., Donyaee, M., B. Kline, R., & K. Padda, H. (2006). Usability measurement and metrics: A consolidated model. Software Quality Control, 4(2), 159--178.
- Shneiderman, B., & Plaisant, C. (2010). Designing the User Interface: Strategies for Effective Human-Computer Interaction: Fifth Edition. Addison-Wesley Publ. Co.
- Sommerville, I. (2006). Ingeniería de Software. España: Pearson Addison Wesley.
- Taylor, R. N., Medvidovic, N., & Dashofy, E. M. (2009). Software Architecture: Foundations, Theory, and Practice. Wiley Publishing.
- Team, T. i. (2015). igrph R package. Obtenido de igrph R package: <http://igraph.org/r/>
- ter Beek, M., Bucchiarone, A., & Gnesi, S. (2007). Web Service Composition Approaches: From Industrial Standards to Formal Methods. Internet and Web Applications and Services, 2007. ICIW '07. Second International Conference on (pág. 15). Mome: IEEE.
- Together Teamsolutions Co., L. (2011). Together - Professional Open Source. Obtenido de <http://www.together.at/prod/workflow/tws>
- Tognazzini, B. (10--15 de Noviembre de 2014). First Principles of Interaction Design. Obtenido de Ask.Tog: <http://asktog.com/atc/principles-of-interaction-design/>
- Twitter, I. (2015). REST APIs - Twitter Developers. Obtenido de REST APIs - Twitter Developers: <https://dev.twitter.com/rest/public>
- WaveMaker, I. (2014). WaveMaker. Obtenido de <http://www.wavemaker.com/>
- Yahoo!, I. (2014). Yahoo Pipes! Obtenido de <https://pipes.yahoo.com/pipes/>
- Yeltayeva, K. (2012). Usability Study of the Taverna Scientific Workflow Workbench. University of Manchester, Faculty of Engineering and Physical Sciences . University of Manchester.
- Zhao, Z., Bhattarai, S., Liu, J., & Crespi, N. (2011). Mashup services to daily activities: end-user perspective in designing a consumer mashups. Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services (pp. 222--229). Ho Chi Minh City, Vietnam: ACM.

Tendencias en la Ingeniería de Software

Impacto en las Tecnologías de Información y Comunicación

ISBN: 978-607-96212-6-1



9 786079 621261



Centro de Investigación
en Matemáticas, A.C.

Jalisco s/n, Valenciana, C.P. 36240
Tel. +52 (473) 732 7155,
Fax. +52 (473) 732 5749
www.cimat.mx