

# Evaluando Adecuación Funcional y Usabilidad en Herramientas de Composición desde la Perspectiva del Usuario Final

Yadira Jarvio Hernández <sup>1</sup>, Perla Velasco-Elizondo <sup>2</sup> y Edgard Benítez-Guerrero <sup>1</sup>

zs13015663@estudiantes.uv.mx, pvelasco@uaz.edu.mx, edbenitez@uv.mx

<sup>1</sup> Universidad Veracruzana, Facultad de Estadística e Informática, Xalapa, VER, 91020, México.

<sup>2</sup> Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería, Zacatecas, ZAC, 98000, México.

DOI: 10.17013/risti.17.96-114

**Resumen:** Cada vez más personas realizan tareas apoyándose de sistemas que son contruidos a partir de la composición de componentes de software. 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 artículo se presenta una evaluación de herramientas de composición de componentes centrada en el usuario final. Una contribución importante de esta evaluación es que, considerando un conjunto de características intrínsecas al desarrollo centrado en composición y a los usuarios finales, se proponen un conjunto de funciones y heurísticas relevantes para detectar problemas generales de adecuación funcional y usabilidad. Estas funciones y heurísticas proveen una base de conocimiento que puede ser extendida en dominios específicos de aplicación para mejorar el diseño de herramientas de composición.

**Palabras-clave:** herramientas de composición; evaluación heurística; inspección de características; usuarios finales.

## *Evaluating Functional Appropriateness and Usability of Composition Tools from an End-User Perspective*

**Abstract:** A growing number of people perform tasks supported by systems that are built through the composition of software components. Despite the popularity of component-based software development and related tools, their use remains complicated for experts, not to mention for end users. This paper provides an evaluation of composition tools focused on end users. A major contribution of this assessment is that, given a set of characteristics that are intrinsic to composition-based software development and to end users, a set of relevant functions and heuristics to detect general problems of functional suitability and usability are proposed. These functions and heuristics provide a knowledge base that can be extended in specific application domains to improve the design of composition tools.

**Keywords:** composition tools; heuristic evaluation; characteristic inspection; end-user.

## 1. Introducción

En la actualidad cada vez más personas realizan diversas tareas apoyándose de sistemas de software. Se puede observar que muchos de estos sistemas 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). Usando mecanismos de composición específicos, estos componentes pueden combinarse para crear sistemas. Ejemplos de componentes en muchos de estos sistemas son los COTS (Commercial Off-The-Shelf) (Morisio, Seaman, Basili, Parra, Kraft, & Condon, 2002), los Servicios Web (Caldwell, Chawla, & Chopra, 2002), los Mashlets (Abiteboul, Greenspan, & Milo, 2008) o las (Web) APIs.

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. De forma similar, existen diversas herramientas, componentes y repositorios de componentes en dominios específicos que permiten el desarrollo de sistemas bajo este enfoque. Algunos ejemplos de herramientas y repositorios incluyen:

- *BioCatalogue*. Es un repositorio público de servicios web sobre biología (Bhagat, et al., 2010).
- *Taverna*. Es una herramienta de composición y ejecución de workflows basados en servicios web. Proporciona acceso a una amplia gama de servicios web y bases de datos principalmente del área de biología (School of Computer Science, 2010).
- *API de Twitter*. Es un componente de tipo web API el cual incluye métodos que permiten a los desarrolladores acceder a los datos básicos de Twitter, así como interactuar con Twitter Search (Twitter, 2015).
- *Yahoo Pipes!*. Es una herramienta de composición y publicación de aplicaciones web tipo mashups. Proporciona varios componentes para la creación de dichas aplicaciones (Yahoo!, 2014).

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:

- a. Son expertos en su dominio profesional pero inexpertos en materia de desarrollo de sistemas.
- b. 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.
- c. Los sistemas que construyen tienen una arquitectura “data-flow”. En términos generales, una arquitectura “data-flow” se caracteriza por tener un conjunto de componentes que realizan una transformación sucesiva de flujos de datos (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 e *igraph* para R (team, 2015) o biólogos realizando análisis de cadenas de proteínas con workflows creados con la herramienta de composición Taverna y servicios Web en BioCatalogue.

A pesar de la popularidad del desarrollo centrado en composición y las herramientas existentes para soportarlo hay evidencia sobre problemas que hacen que el uso de éstas sigue siendo complicado. Además, existe muy poco material que permita comprender esta problemática en un contexto de usuarios finales. Este artículo reporta una evaluación de herramientas de composición de componentes centrada en el usuario final. En contraste con trabajos relacionados, una contribución importante de esta evaluación es que, considerando un conjunto de características intrínsecas al desarrollo centrado en composición y a los usuarios finales, se proponen y utilizan un conjunto de funciones y heurísticas relevantes para detectar problemas generales de *adecuación funcional* y de *usabilidad*. En términos generales, la adecuación funcional tiene que ver con la capacidad de la herramientas de composición de incorporar funciones que satisfacen necesidades declaradas o implícitas. La usabilidad tiene que ver con la facilidad de uso de dichas herramientas. Las funciones y heurísticas identificadas proveen una base de conocimiento que puede ser extendida en dominios específicos de aplicación para mejorar el diseño de herramientas de composición.

El artículo está organizado de la siguiente manera. La sección 2 discute trabajos relacionados con la evaluación funcional y de usabilidad de herramientas de composición de componentes. En la sección 3 se describen las particularidades de la evaluación realizada, incluyéndose las funciones y heurísticas propuestas. 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 términos generales la funcionalidad de un producto de software puede entenderse como la capacidad que éste tiene para tomar datos de entrada, procesarlos y proveer los resultados esperados a través de funciones específicas. El estándar ISO/IEC 25100 (ISO/IEC, 2011), que es un modelo de calidad de producto, considera la funcionalidad como una de las ocho características relevantes a tener en cuenta para evaluar las propiedades de un producto software determinado. En este estándar la funcionalidad es denotada como *adecuación funcional* y se define como “la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.”

Por otra parte, 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 (ISO/IEC, 1998) y 25100 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”. Similarmente, en el estándar ISO/IEC 25100 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”.

En el pasado se han realizado evaluaciones de funcionalidad y usabilidad 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, 2007), (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 de funcionalidad y usabilidad, estas evaluaciones presentan dos limitaciones importantes: (i) son evaluaciones de una sola herramienta lo cual hace difícil el determinar si los problemas detectados se presentan en otras, y (ii) las evaluaciones no consideran características intrínsecas al desarrollo centrado en composición que impactan en la funcionalidad y usabilidad de herramientas de composición de componentes cuando son utilizadas por usuarios finales. La Tabla 1 describe estas características.

### 3. Descripción de la Evaluación

En las siguientes secciones se describen los detalles de la evaluación de la adecuación funcional y de la usabilidad de algunas herramientas representativas para soportar la composición automática de componentes.

#### 3.1. Herramientas Seleccionadas

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:<sup>1</sup>

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.

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

Características	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, por ejemplo, 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, por ejemplo, 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, por ejemplo, desempeño, disponibilidad. La calidad de 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.

Tabla 1 – Características intrínsecas al desarrollo centrado en composición que impactan en la funcionalidad y usabilidad de herramientas de composición de componentes cuando son utilizadas por usuarios finales.

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!.

### 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 %.”*

#	Nombre	Características			
		SD	DD	WD	DI
	Apache ODE, (Apache, 205).	✓	✓	✓	✓
	Apatar, (Apatar, 2014).	×	✓	✓	✓
	BonitaSoft, (Bonitasoft, 2015).	✓	✓	✓	✓
	Dapper, (Al Sarraj, 2014).	✓	×	✓	×
	Enhydra Shark, (Together Teamsolutions Co., 2011).	✓	×	✓	✓
	Google Mashup, (Al Sarraj, 2014).	✓	×	✓	×
	Intel Mash Maker, (Al Sarraj, 2014).	✓	×	×	×
	Jaw Flow, (Garcês, De Jesus, Cardoso, & Valente, 2009).	✓	×	✓	×
	JBoss, (RedHat, 2014).	✓	×	✓	✓
	JackBe, (Al Sarraj, 2014).	✓	✓	✓	×
	JOpera, (JOpera.org, 2013).	✓	×	✓	✓
	LONI Pipeline, (Laboratory, 2015).	✓	✓	✓	✓
	Marmite, (Patel , Na , Latih, Wills, Shukur , & Mull, 2010).	✓	✓	×	×
	Microsoft Popfly, (Al Sarraj, 2014).	✓	✓	✓	×
	OpenKapow, (Kapow, 2015).	✓	✓	✓	×
	Potluck, (Al Sarraj, 2014).	×	×	✓	×
	QedWiki, (Patel , Na , Latih, Wills, Shukur , & Mull, 2010).	✓	×	✓	×
	RSSBus, (RSSBus, 2014).	×	×	✓	×
	RUNA WFE, (RunaWFE, 2013).	×	×	✓	✓
	Taverna, (School of Computer Science, 2010).	✓	✓	✓	✓
	Wave Maker, (Muñoz Jiménez, 2010), (WaveMaker, 2014).	✓	✓	✓	✓
	Yahoo Pipes, (Yahoo!, 2014).	✓	✓	✓	✓
	YAWL, (Garcês, De Jesus, Cardoso, & Valente, 2009).	✓	×	✓	×

Tabla 2 – Herramientas consideradas en la evaluación.

Para poder crear ese sistema, se consideraron 5 componentes de tipo servicio web; 3 son provistos por diferentes fabricantes y 2 fueron desarrollados por los autores de este trabajo. Estos componentes se describen con mayor detalle en la Tabla 3. Es importante mencionar que los atributos de calidad de cada componente se obtuvieron de la página del fabricante de cada servicio, y en el caso de los componentes desarrollados por los autores de este trabajo, sus respectivos atributos de calidad se calcularon de forma local.

### 3.3. Método de Evaluación

Se pretende detectar un conjunto general, (no de dominio específico) de problemas en el proceso de composición de componentes. Se prefieren métodos prácticos, efectivos y con poca demanda en tiempo, esto debido a la naturaleza, alcance y tiempo disponible para el desarrollo de este trabajo. Es por esto que los métodos de inspección se consideran adecuados para este trabajo, ya que son métodos en donde un grupo de expertos examinan aspectos relacionados con el funcionamiento del software y la usabilidad de la interfaz (Hollingsed & Novick, 2007). Entre los métodos de inspección relevantes se encuentran: Recorrido cognitivo, Inspección de estándares, Recorrido de usabilidad plural, Inspección de características y Evaluación heurística.

ID	Operación	Descripción	Calidad de Servicio	Ubicación
C1	Búsqueda de información textual	Recibe el tema a buscar, realiza una búsqueda en twitter del tema indicado y devuelve una lista de comentarios.	R: 483 ms D: 100%	<a href="https://dev.twitter.com/">https://dev.twitter.com/</a>
C2	Búsqueda de imágenes	Recibe el tema a buscar, realiza una búsqueda en instagram del tema indicado y devuelve una lista de imágenes.	R: 485 ms D: 100%	<a href="http://developers.instagram.com/">http://developers.instagram.com/</a>
C3	Traducción de inglés a español	Recibe una lista de información textual y devuelve la lista traducida al idioma español.	R: 400 ms D: 100%	<a href="https://developers.google.com/">https://developers.google.com/</a>
C4	Ordena información textual	Recibe la lista de comentarios y la ordena de forma ascendente tomando en cuenta la fecha de creación.	R: 150 ms D: 90%	Estos componentes fueron desarrollados por los autores de este trabajo
C5	Ordena información textual		R: 90 ms D: 100%	

Tabla 3 – Componentes considerados en la evaluación.

La Inspección de características y la Evaluación heurística fueron los métodos seleccionados para evaluar la funcionalidad y la usabilidad de las herramientas de composición de componentes seleccionadas. En las siguientes secciones se presentan los detalles correspondientes.

#### 3.3.1. Evaluando Funcionalidad: Inspección de Características

Considerando nuestro objetivo de identificar un conjunto de problemas generales de adecuación funcional en herramientas de composición de componentes que sean relevantes al contexto de usuarios finales, se considera que la inspección de características es un método apropiado. La inspección de características es un método en el cual un grupo de expertos inspecciona un producto de software para determinar si provee un conjunto de funciones que son generalmente extraídas a partir de la definición de un escenario de uso y de sus tareas asociadas.

El estándar ISO/IEC 25010 considera la adecuación funcional desde tres perspectivas:

1. **Completitud funcional:** grado en el que las funciones provistas por el sistema corresponden a las funciones especificadas.
2. **Correctitud funcional:** grado en el que las funciones del sistema proveen resultados correctos con el nivel de precisión requerido.
3. **Pertinencia funcional:** grado en el que las funciones del sistema facilitan la realización de las tareas y los objetivos especificados.

Tomado en cuenta lo anterior, así como también las características intrínsecas al desarrollo centrado en composición (mostradas en la Tabla 1), en la Tabla 4 se describen las funciones propuestas para evaluar la adecuación funcional de las herramientas de composición. En esta evaluación se medirá el radio de funciones presentes en relación con el total de funciones consideradas en la evaluación; los detalles del cálculo de ésta métrica se describen en la sección 4.1.

<b>Función</b>	<b>Descripción</b>	<b>Perspectiva</b>
<b><i>Generación de la Composición Automáticamente (CA)</i></b>	Denota la capacidad de la herramienta para generar el código de la composición de componentes de forma automática.	Completitud funcional
<b><i>Verificación de la Composición Automáticamente (VA)</i></b>	Denota la capacidad de la herramienta para comprobar de forma automática que una composición sea correcta sintácticamente.	Correctitud funcional
<b><i>Escalabilidad (ES)</i></b>	Denota la capacidad de la herramienta de permitir al aumento de componentes y/o operaciones.	Pertinencia funcional

Tabla 4 – Funciones consideradas en la evaluación.

### **3.3.2. Evaluando Usabilidad: Evaluación Heurística**

Para evaluar problemas generales de usabilidad en herramientas de composición de componentes que sean relevantes al contexto de usuarios finales, se considera que la evaluación heurística es un método apropiado (Nielsen & Molich, 1990).

La evaluación heurística es un método en el cual 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. La palabra “heurística” viene de la griega “eureka”, y en el contexto de la evaluación heurística, se refiere a una serie de principios generales que consideran los expertos para llevar a cabo dicha evaluación. Estos principios se adecúan para asegurar que se adaptan al contexto de la interfaz del software que se va evaluar.

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 heurística es considerada económica, intuitiva y utilizable en las primeras fases del proceso de desarrollo, esto para detectar la mayor cantidad posible de problemas de usabilidad antes de que el software esté en producción. Además permite



hallar tanto problemas mayores como menores (Martins, Alexandra, & Rocha, 2013), (Saltiveri, Fransi, & Salla, 2013).

Existen varias propuestas de heurísticas o principios de diseño que pueden ser considerados en la evaluación heurística de la usabilidad. En la Tabla 5 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.

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 5, 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 6 define las sub-heurísticas que se consideraron para la evaluación del conjunto de heurísticas y principios presentados en la Tabla 5. Estas sub-heurísticas se seleccionaron de listas predefinidas --por ejemplo, (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. En esta evaluación se calculará el *nivel del severidad del error* provocado por el desapego a las sub-heurísticas. Esta métrica es muy común en la evaluación heurística; los detalles del cálculo de ésta métrica se describen en la sección 4.2.

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

Tabla 5 – Conjunto de heurísticas y principios seleccionados para la evaluación.

### 3.4. Participantes en la Evaluación

La evaluación de las herramientas seleccionadas fue realizada por cinco evaluadores, cuatro de ellos Licenciados en Informática y actualmente alumnos del cuarto semestre del programa de Maestría en Sistemas Interactivos Centrados en el Usuario de la Universidad Veracruzana. 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.

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	

Tabla 6 – Conjunto de sub-heurísticas seleccionadas para la evaluación.

## 4. Resultados Obtenidos

En las siguientes secciones, se muestran los resultados consolidados de las evaluaciones realizadas por los expertos.

### 4.1. Resultados de la Evaluación de la Adecuación Funcional

En las Tablas 7 y 8 se muestran los resultados obtenidos por los cinco expertos (E1 - E5) en la evaluación de la adecuación funcional de las herramientas de composición seleccionadas. En la tablas, si la herramienta provee la función se denota con un 1 y si no con un 0. La columna “R” es el promedio obtenido de los valores de los cinco expertos.

Como se puede observar, la mayoría de las herramientas permiten generar la composición automáticamente (función CA). Solamente LONI PipeLine no cuenta con

esta capacidad. También se puede ver que todas las herramientas permiten verificar de forma automática la correctitud de la composición (función VA). Cuando se trata de aumentar componentes y/o operaciones a la composición (función ES), la mayoría de las herramientas cuentan con esta capacidad; la excepción fue Yahoo Pipes!.

Función	Apache ODE					BonitaSoft					LONI Pipeline							
	E1	E2	E3	E4	E5	R	E1	E2	E3	E4	E5	R	E1	E2	E3	E4	E5	R
CA	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
VA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ES	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tabla 7 – Resultados de la evaluación de la adecuación funcional - I.

Función	Taverna					Wave Maker					Yahoo Pipes!							
	E1	E2	E3	E4	E5	R	E1	E2	E3	E4	E5	R	E1	E2	E3	E4	E5	R
CA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
VA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ES	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

Tabla 8 – Resultados de la evaluación de la adecuación funcional - II.

La adecuación funcional (*AF*) de cada herramienta (*h*), podría ser evaluada usando la siguiente fórmula:

$$AF(h) = 1 - A/B$$

donde:

*A* denota el número de funciones ausentes o problemáticas en la evaluación,  
*B* denota el número de funciones consideradas,

quedando como sigue:

$$AF(\text{Apache ODE}) = 1 - 0/3 = 1$$

$$AF(\text{BonitaSoft}) = 1 - 0/3 = 1$$

$$AF(\text{LONI Pipeline}) = 1 - 1/3 = 0.6$$

$$AF(\text{Taverna}) = 1 - 0/3 = 1$$

$$AF(\text{Wave Maker}) = 1 - 0/3 = 1$$

$$AF(\text{Yahoo Pipes!}) = 1 - 1/3 = 0.6$$

Considerando que un valor de  $AF$  lo más cercano a 1 es mejor, solamente las herramientas de composición LONI Pipeline y Yahoo Pipes! tienen un valor no óptimo de adecuación funcional.

#### 4.2. Resultados de la Evaluación de Usabilidad

En la Tabla 9 y la Tabla 10 se muestran los resultados obtenidos por cada experto (E1 – E5) durante la evaluación de usabilidad de las herramientas de composición seleccionadas. Los valores denotan el nivel de severidad del error (NS) detectado en la evaluación el cual se calcula como sigue:

$$NS = (F) \times (I)$$

donde:

$F$  denota la frecuencia con la que se incurre en el error.  
 $I$  denota el impacto asociado a cada sub-heurística.

La frecuencia ( $F$ ) puede tomar los siguientes valores:

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

El impacto ( $I$ ) puede tomar los siguientes valores:

- 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.

Con base en lo anterior, el nivel de severidad (NS) puede tomar los siguientes valores:

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

En la Tabla 9 y la Tabla 10 el valor mostrado en las columnas E1, E2, E3, E4, E5 indica el nivel de severidad del error (NS) detectado por cada experto. La columna "R" es el promedio obtenido de los valores de los cinco expertos. Este promedio está graficado en la Figura 1 y se observa lo siguiente.

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 9 – Resultados de la evaluación de usabilidad - I.

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

Tabla 10 – Resultados de la evaluación de usabilidad - II.

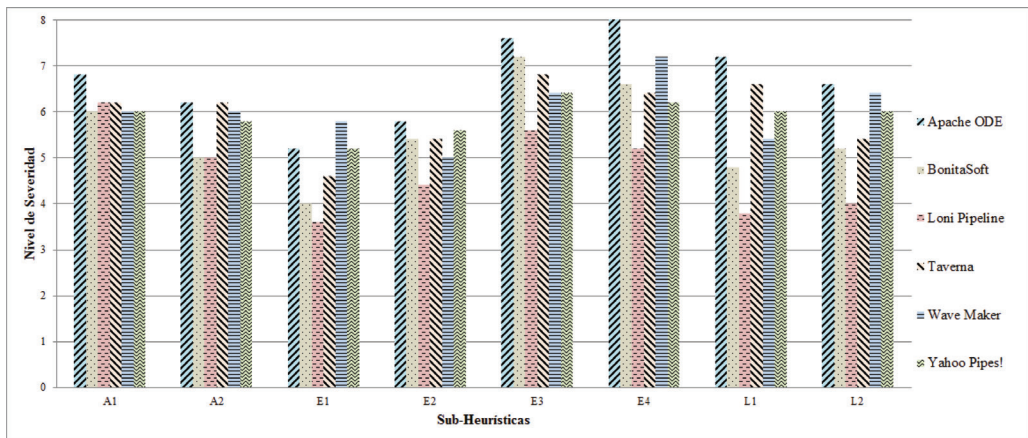


Figura 1 – Resultados obtenidos de la evaluación heurística.

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 los mensajes 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.

#### Conclusiones y Trabajo Futuro.

En este trabajo se presentó una evaluación de herramientas de composición para detectar problemas generales de adecuación funcional y 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 funciones y heurísticas que fueron identificadas considerando un conjunto de aspectos intrínsecos al desarrollo centrado en composición y a los usuarios finales.

A partir de los resultados de la evaluación realizada se puede concluir que las herramientas evaluadas presentan pocos problemas de adecuación funcional y varios problemas de usabilidad para un usuario final. Entre los más importantes de usabilidad se encuentran el nulo soporte ofrecido para la especificación de aspectos funcionales y de calidad del servicio del sistema a construir y para la selección de componentes basada en dichas especificaciones. Además de que las herramientas no proveen funciones que brinden el soporte necesario a los usuarios para resolver los errores que puedan tener al hacer una composición, ya que como se observó, las herramientas no muestran las causas ni las posibles soluciones a dichos errores.

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 --por ejemplo, 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 becarío: 297377).

## Referencias

- Abiteboul, S., Greenspan, O., & Milo, T. (2008). Modeling the mashup space. 10th ACM workshop on Web information and data management (pp. 87-94). ACM.
- Al Sarraj, W. (2014). Toward Usability Evaluation Criteria for Web Mashup Makers for End-Users. *International Journal of Advanced Computer Technology* , 3, 23-32.
- Apache, S. F. (2005). Apache ODE. From <http://ode.apache.org/>
- Apatar, I. (2014). Apatar. Connecting data. From <http://www.apatar.com/>
- Baryannis, G., & Plexousakis, D. (2010). Automated Web Service Composition: State of the Art and Research Challenges. Foundation for Research & Technology - Hellas - Institute of Computer Science.
- Beek, M., Bucchiarone, A., & Gnesi, S. (2007). A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods. Second International Conference on Internet and Web Applications and Services (pp. 15-21). IEEE Computer Society .
- Bhagat, J., Tanoh , F., Nzuobontane, E., Laurent , T., Orłowski, J., Roos, M., et al. (2010). BioCatalogue: a universal catalogue of web services for the life sciences. *Nucleic Acids Research* , 689-694 .
- Bonitasoft, I. (2015). Bonitasoft. From <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*. From <http://martinfowler.com/articles/microservices.html>



- Garcês, R., De Jesus, T., Cardoso, J., & Valente, P. (2009). Open Source Workflow Management Systems: A Concise Survey. In *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. *17th Monterey Conference on Large-Scale Complex IT Systems: Development, Operation and Management* (pp. 157-182). Springer-Verlag.
- González, M. P., Lorés, J., & Pascual, A. (2006). Evaluación Heurística. In *La Interacción Persona Ordenador* (pp. 1-40). AIPO Press.
- Hassan Montero, Y., & Ortega Santamaría, S. (2009). Informe APEI sobre usabilidad. APEI.
- Hollingsed, T., & Novick, D. G. (2007). Usability inspection methods after 15 years of research and practice. *Annual ACM International Conference on Design of Communication* (pp. 249-255). ACM.
- Insfran, E., Cedillo, P., Fernández, A., Abrahão, S., & Matera, M. (2012). Evaluating the Usability of Mashups Applications. *Eighth International Conference on the Quality of Information and Communications Technology* (pp. 323-326). IEEE Computer Society.
- ISO/IEC. (1998). ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 11 Guidance on usability.
- ISO/IEC. (2011). ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuARE) -- System and software quality models. ISO.
- JOpera.org. (2013). JOpera. Process Support for Web Services. From <http://www.jopera.org/>
- Kapow. (2015). Kapow Software. A Kofax Company. From <http://www.kapowtech.com/>
- Knoke, D., & Yang, S. (2008). *Social Network Analysis*. SAGE Publications, Inc.
- Laboratory, o. N. (2015). Loni Pipeline. From <http://pipeline.bmap.ucla.edu/>
- Malinga, M., Gruner, S., & Koschmider, A. (2013). Quality and Usability of Mashup Tools: Criteria and Evaluation. *South African Institute for Computer Scientists and Information Technologists Conference* (pp. 154-159). ACM.
- A.I. Martins, Q. Alexandra, N.P. Rocha (2013). Avaliação de Usabilidade: Uma Revisão Sistemática da Literatura. *Revista Ibérica de Sistemas y Tecnologías de la Información*. Vol. 11, 31-43.
- Mehandjiev, N., Namoune, A., Wajid, U., Macaulay, L., & Sutcliffe, A. (2010). End user service composition: Perceptions and requirements. *8th European Conference on Web Services* (pp. 139-146). IEEE Computer Society.
- Minhas, S. S., Sampaio, P., & Mehandjiev, N. (2012). A Framework for the Evaluation of Mashup Tools. *Ninth International Conference on Services Computing* (pp. 431-438). IEEE Computer Society.

- Moriso, M., Seaman, C., Basili, V., Parra, A., Kraft, S., & Condon, S. E. (2002). COTS-based software development: processes and open issues. *Journal of Systems and Software*, 61 (3), 189-190.
- Muñoz Jiménez, J. A. (2010). Programación con AJAX. From <http://jamj.webcindario.com/programacion/ajax/AJAX.pdf>
- Nielsen, J. (1994). Enhancing the Explanatory Power of Usability Heuristics. *Conference on Human Factors in Computing Systems* (pp. 152--158). 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. (2010). A Study of Mashup as a Software Application Development Technique with Examples from an End-User Programming Perspective. *Journal of Computer Science*, 6 (12), 1406-1415.
- Portchelvi, V., Prasanna Venkatesan, V., & Shanmugasundaram, G. (2012). Achieving Web Services Composition – a Survey. 2 (5), 195-202.
- RedHat. (2014). JBoss Developer. From <http://www.jboss.org/>
- Roy Chowdhury, S. (2012). Assisting End-user Development in Browser-based Mashup Tools. *International Conference on Software Engineering* (pp. 1625-1627). IEEE Computer Society.
- RSSBus, I. (2014). RSSBus. From <http://www.rssbus.com/>
- RunaWFE. (2013). RunaWFE. From <http://wf.runa.ru/>
- T.G. Saltiveri, E.C. Fransi, Y. M Salla (2013). Análisis de usabilidad de cooperativas del sector de la fruta y aceite en el área de Lleida. *Revista Ibérica de Sistemas y Tecnologías de la Información*. Vol. 11, 31-43.
- School of Computer Science, U. o. (2010). Taverna. From <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*. Pearson Addison Wesley.
- Taylor, R. N., Medvidovic, N., & Dashofy, E. M. (2009). *Software Architecture: Foundations, Theory, and Practice*. Wiley Publishing.
- team, T. i. (2015). igraph R package. From <http://igraph.org/r/>
- ter Beek, M., Bucchiarone, A., & Gnesi, S. (2007). Web Service Composition Approaches: From Industrial Standards to Formal Methods. *International Conference on Internet and Web Applications and Services*, (pp. 15-21). IEEE Computer Society.
- Together Teamsolutions Co., L. (2011). Together - Professional Open Source. From <http://www.together.at/prod/workflow/tws>

- Tognazzini, B. (2014). First Principles of Interaction Design. From Ask.Tog: <http://asktog.com/atc/principles-of-interaction-design/>
- Twitter, I. (2015). REST APIs - Twitter Developers. From <https://dev.twitter.com/rest/public>
- WaveMaker, I. (2014). WaveMaker. From <http://www.wavemaker.com/>
- Yahoo!, I. (2014). Yahoo Pipes! From <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. International Conference on Information Integration and Web-based Applications and Services (pp. 222-229). ACM.