

# Revisión Sistemática de Literatura: Visualización de Software

Croys Alvarado Ávila, Luis Vargas Jiménez y Adonay Jiménez Ulate

Escuela de Ingeniería,  
Universidad Latinoamericana de Ciencia y Tecnología,  
ULACIT, Urbanización Tournón, 10235-1000  
San José, Costa Rica  
calvaradoa699@ulacit.ac.cr, lvargasj770@ulacit.ac.cr, ajimenezu@ulacit.ac.cr  
<http://www.ulacit.ac.cr>

**Abstract.** Esta investigación se enfoca en un análisis de mapeo sistemático que utiliza una fuente de información de 27 artículos correspondientes a los años 2005 y 2014, los cuales fueron obtenidos de La Segunda Conferencia de Trabajo IEEE en Visualización de Software. Los autores de estas publicaciones exponen una temática variada sobre el tema Visualización de Software. Con base en la pregunta o problema planteado para este trabajo se hace uso de los 5 pasos establecidos por Petersen. Para este estudio de mapeo sistemático se incluye un análisis de resultados que aprovecha elementos visuales como el uso de gráficos y tablas, se facilita la exposición de los resultados obtenidos, se da respuesta a la pregunta y problemática planteada, además de aportar un tema de interés a investigadores de la Visualización de Software (VS) y la Analítica Visual (VA).

**Keywords:** visualización, técnicas, software, Analítica Visual, Mapeo Sistemático

## 1 Introducción

El propósito fundamental de esta investigación es determinar el papel de la visualización y análisis en la simplificación de las tareas de desarrollo y mantenimiento de software, además de centralizarse en el uso que se le ha dado a estas herramientas de visualización en los años 2005 y 2014. Se investigó artículos enfocados en diversas herramientas orientadas a visualización de software y en casos de uso particular los propuestos por los autores de esos conceptos. De los mismos se procederá a extraer, en forma grupal, datos o información que se convertirán en el insumo para un depósito o reservorio documental (hoja en Excel) que venga a facilitar búsquedas futuras mediante la extracción o consulta de información referente a este tema y que contribuirá significativamente a la elaboración del presente artículo investigativo. Como parte del trabajo investigativo se utilizará motores de búsqueda del calibre de Google y EBSCO con el propósito de localizar o encontrar documentación o información referente a

nuestro campo de estudio. Además se provee un material a seleccionar, clasificar, identificar, revisar, y utilizar de acuerdo con el objetivo principal del proyecto actual. Se describe algunas de las herramientas que se utilizan para facilitar el proceso de visualización de software y se indica cuáles, de acuerdo al uso cuantitativo y cualitativo que cada herramienta examinada, cuáles son las más utilizadas y de mayor valía. Aunado a lo dicho, se determina tareas compatibles con la visualización de software considerando los procesos de desarrollo y mantenimiento de sistemas, los tipos de validaciones empleados para la prueba de soluciones propuestas por cada autor de artículo o documento. Al finalizar el proceso investigativo se realiza un resumen del mismo, así como la inclusión de las conclusiones a las que se llega después del trabajo científico efectuado a todos los documentos investigados.

Este trabajo es una continuación de la investigación de González (González-Torres, 2015).

## 2 Metodología

En esta sección se explica cuál es el tipo de metodología desarrollada para lograr el objetivo deseado.

### 2.1 Pregunta de Investigación

La finalidad de esta revisión sistemática de literatura enfocada en la visualización de software es poder contestar la siguiente pregunta de investigación

¿Cómo se han utilizado la visualización y la analítica visual en las tareas de desarrollo y mantenimiento de software?

### 2.2 Criterios de Inclusión y Exclusión

Por ser, el estudio de mapeo sistemático, realizado a una cantidad relativamente pequeña y selecta de publicaciones o artículos, los mismos fueron considerados para el presente trabajo en su totalidad.

### 2.3 Periodo del estudio

Las publicaciones estudiadas fueron publicadas en los proceedings de IEEE International Workshop on Visualizing Software for Understanding and Analysis(VISSOFT).

### 2.4 Papers estudiados

El estudio incluye 16 publicaciones del año 2014 y 11 del año 2005.

## 2.5 Relevancia de los papers

El estudio que se realiza trata de artículos completos y relevantes al área de estudio, los cuales han sido aceptados en la conferencia especializada de visualización de Software, citada anteriormente.

## 2.6 Criterios de Clasificación

Las categorías de clasificación utilizadas en este trabajo están basadas en la investigación de González TESIS-GONZALEZ-2015, (22 descritas por González) de las cuales se analizan 8 de ellas:

1. Mejoramiento de la calidad del software
2. Diseño de software y modelado
3. Apoyo a la depuración
4. Apoyo a la ingeniería inversa
5. Pruebas de Software
6. Colaboración y conciencia del equipo
7. Comprensión de dependencias
8. Comprensión de las arquitecturas de sistemas

## 3 Resultados

Se obtuvo un total de 76 autores distribuidos entre las 27 publicaciones o artículos analizados. Las divulgaciones contaron con la participación de 1, 2, 3, 4, 5 y hasta 6 autores por publicación. Aun cuando la mayoría de las publicaciones contaban con más de 1 autor, algunos de ellos sólo tuvieron participación conjunta con otros autores en 1 sólo artículo (este fue en el mayor de los casos). En la tabla 1 se puede observar Título del Artículo, Autores por Artículo, Número de Artículos, Número de Autores y Número de Artículos por autor. Para un mejor entendimiento de la tabla 1, se adjudicó el color azul para determinar autores que participaron 1 única vez en la investigación y elaboración de un único y exclusivo artículo (en forma conjunta con otros autores), el color rojo para aquellos autores que participaron en 2 artículos diferentes y, el color amarillo para aquellos autores que participaron en 4 artículos o publicaciones distintas (en esta investigación sólo 1 autor participó en 4 artículos).

**Tabla # 1**

Título del Artículo	AUTORES POR ARTICULO	# DE ARTICULOS	# DE AUTORES	# DE ARTICULOS POR AUTOR
Action-Based Visualization	Leiri Järveläinen			
Combining Text and Tactile Views of Code	Huam-Matt Jernum Jens Nohr			
Integrating Assembly Diagrams into Spreadsheet Environments	Michael Heuser			
Integrating Diagrams for Visualizing Loop Memory Models	Daniel Kubez			
Search Space Pruning Constraints Visualization	Arnaud Besson			
Understanding the Syntax/Code Interplay in an embedded programming language	Blair Hagren Jakob Kurzak			
Using a Task-Oriented Framework to Characterize Visualization Approaches	Jens Nohr			
Validation of Software Visualizations: Tools, A Systematic Mapping Study	Marcelo Scholtz Omar Bonassar			
Feature Relations Graphs: A Visualization Paradigm for Feature Constraints in Software Product Lines	Trevor Zaidi			
How Developers Visualize Complex Messages: A Foundational Approach to Navigation Communication	Adam Malczak Kewei Lohsch			
Crunch? Vignette: A Visual Analytics Tool for Understanding Source and Test Co-Evolution	Bernett Eas David Ross			
Abstractly Structured Visualizations of Assemblable Content Flows based on Regular Expressions	Işıl Turgut Ayşe Wilschütz			
Visualizing Developer Interactions	Robert Meehl Andreas Meier			
Visualizing the Evolution of Systems and their Library Dependencies	Rishi Chakraborty Gowtham Kalya			
McMinn: A Memory-Based Visualization of Software Evolution	Chen Di Boyan			
Interactive Visualization for High-Software Development Teams	Chen Di Boyan			
DiB: The Software Architecture Browser	Chen Di Boyan			
Iterative Exploration of CML Diagrams	Richard Sharp			
Visual Exploration of Combined Architectural and Metric Information	Chunhui F. J. Liao			
Visual Problems for the Visualization of Software Metrics	Maurice Tannier			
White Cards: With Visualization of Evolving Software in 3D	Danyu He Johannes Lorenz			
Multi-Level Method Understanding Using Hierarchies	Cedric Henning			
Towards More Feasibility in Software Visualization Tools	Stephan Döcker Michael Lorenz			
Interactive Visual Mechanisms for Exploring Source Code Evolution	Marco D'Amico Harald Oll			
Towards More Feasibility in Software Visualization Tools	Bernd F. Raus			
Interactive Visual Mechanisms for Exploring Source Code Evolution	Thorsten Schäfer			
Visualizing Visual Mechanisms for Exploring Source Code Evolution	Benjamin Tröh			
<b>TOTAL DE ARTICULOS ANALIZADOS</b>		<b>26</b>	<b>72</b>	<b>1</b>
		<b>6</b>	<b>3</b>	<b>2</b>
		<b>4</b>	<b>1</b>	<b>4</b>

Fig. 1. Enfoque de investigación + Año de artículos + Tipo de Validación + Foco de investigación

La tabla 1 muestra como de los 76 autores que participaron en los 27 artículos 72 (Autores por Artículo en color azul) tuvieron participación (1 vez en un solo artículo) en 26 de los artículos. 3 autores (Autores por artículo color rojo) participaron en 2 artículos distribuidos en 6 de los 27 artículos analizados y, finalmente 1 autor (Autores por artículo color amarillo) participó en 4 de los 27 artículos.

Con el propósito de ayudar a reforzar la información anterior el gráfico 1, nos muestra la distribución o cantidad de autores por publicación:



En los resultados del gráfico 1, podemos observar los 27 títulos de artículos analizados, así como la cantidad de autores por artículo. Los datos se distribuyen de la siguiente manera: 2 artículos fueron elaborados por solamente 1 autor (no necesariamente la misma persona), 10 artículos elaborados por sólo 2 autores, 7 artículos elaborados por solamente 3 autores, en 6 de los artículos participaron únicamente 4 autores y, por último en 2 de los artículos participaron 6 autores. Se concluye que de acuerdo con los artículos investigados, la mayoría de los autores limitan sus investigaciones a solo un estudio.

**Tabla 1. Enfoque de investigación+año de artículos+tipo de validación+foco de investigación**

Enfoque de Investigación	Año		Tipo de Validación				Foco de la Investigación			
	2005	2014	Caso de Estudio Usuario	Caso de Estudio Experimental	Uso de Herramienta	Reporte de Experiencia	Evaluación	Propuesta de Solución	Validación	
Caso de Estudio	3	4	7						7	
Evaluativo	4	4			3			8		
Lecciones aprendidas	4	6	5	5		10				
Estudio Sistemático de Mapeo		2	2						2	

**Fig. 3.** Enfoque de investigación + Año de artículos + Tipo de Validación + Foco de investigación

La publicación de los artículos analizados correspondió a los años 2005 y 2014 siendo este último, el de mayor cantidad de publicaciones. El tipo de validación que predomina en estas publicaciones es el Caso de estudio, con un total de 9 de los 27 artículos, 7 de los cuáles tienen un enfoque de investigación también basado en el Caso de estudio. Sin embargo, respecto a la técnica, la predominante fue Lecciones aprendidas, también denominadas conocimientos adquiridos sobre un proceso o sobre una o varias experiencias. En lo que concierne al Foco de investigación, el Reporte de experiencia es el de mayor clasificación.

**Tabla 2. Tareas+año de artículos+tipo de validación+foco de investigación**

Tareas	Año		Tipo de Validación					Foco de la investigación			
	2005	2014	Caso de Estudio	Estudio de Usuario	Caso de Uso	Experimental	Uso de Herramienta	Reporte de Experiencia	Evaluación	Propuesta de Solución	Validación
Mejoramiento de la calidad del software	1	4	1	1	2	1	0	2	2	0	1
Diseño de software y modelado	3	2	1	2	0	0	2	2	2	0	1
Apoyo a la depuración	0	5	3	1	0	1	0	2	0	0	3
Apoyo a la ingeniería inversa	1	0	0	0	1	0	0	0	1	0	0
Pruebas de Software	0	4	1	1	0	1	1	2	1	0	1
Conciencia del equipo y la colaboración	3	0	1	0	0	2	0	2	0	0	1
Comprensión de dependencias	0	1	1	0	0	0	0	0	0	0	1
Comprensión de las arquitecturas de sistemas	3	0	1	0	2	0	0	0	2	0	1

Fig. 4. Tareas + Año de artículos + Tipo de validación de investigación

Como se puede observar en la tabla 3, se realizó un análisis que involucra un Crosstabs, en el cuál interviene diversas variables, tales como tareas, años de artículos, tipo de validación y foco de investigación, contra la variable tareas. Los resultados obtenidos revelan que para el año 2014 se dio un aumento en los estudios relacionados con las tareas de Mejoramiento de la calidad del software, apoyo a la depuración y pruebas del software. En lo que respecta al tipo de validación, los casos de estudios predominan, enfocados en la tarea de depuración, lo cual brinda mayores herramientas a los desarrolladores. Y el Foco de investigación continúa siendo el Reporte de experiencia, dándose un balance entre las diferentes tareas.

		Visualizaciones											
		Treemaps	Matrix layout	Bar charts	Timeline	Heatmap	Node-link diagrams	UML diagrams	Graphs	Circular graph	Interaction diagrams	Fractals	Otros
Mejoramiento de la calidad del software					1	1	1						1
Diseño de software y modelado	1					1	1			2	1		1
Apojo a la depuración	2	1	2			1	1		1	1			
Apojo a la ingeniería inversa	1							1					1
Pruebas de Software			2	2				4	2				2
Conciencia del equipo y la colaboración							2						2
Comprensión de dependencias				1					1				1

Fig. 5. Tareas + Visualización

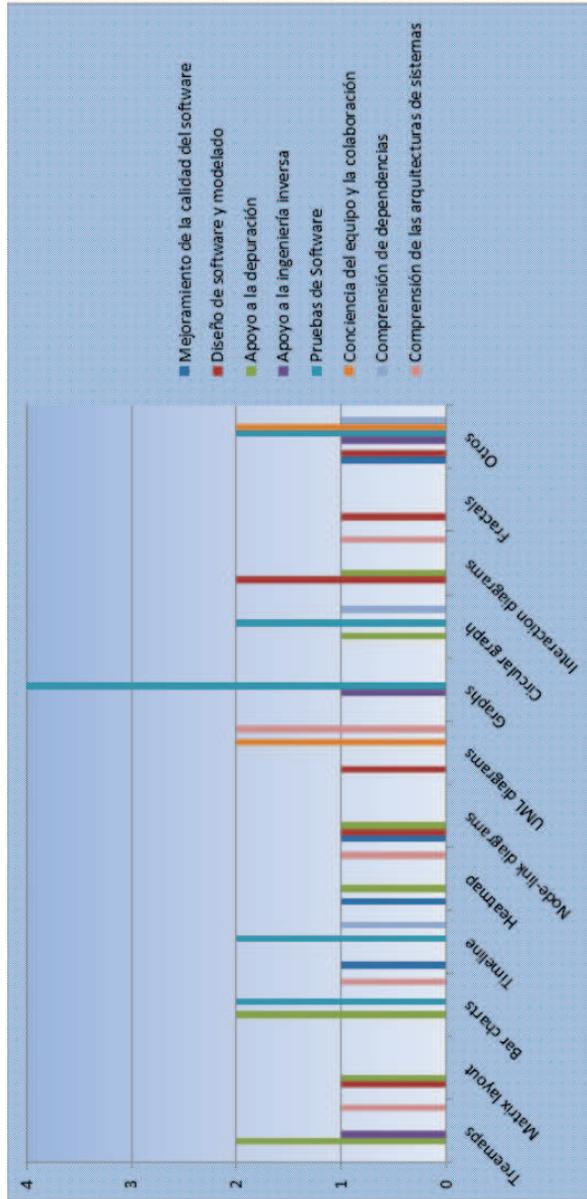


Fig. 6. Tareas + Visualización

El gráfico 2 muestra los resultados obtenidos de la tabla 4 en donde se cruza la variable tareas contra la variable tipo de visualización. De manera general, la tarea Mejoramiento de la calidad del software utiliza la mayor cantidad de visualizaciones, siendo el de mayor uso, las visualizaciones tipo gráficos. La cantidad de visualizaciones, relacionadas con las tareas restantes es similar, y podemos apreciar que giran en torno a una o dos tareas, esto en la distribución y análisis realizados a los 27 artículos estudiados. De manera general, podemos decir que se cuenta con muchos tipos de visualizaciones, sin embargo de todos ellos los más utilizados son los gráficos, seguidos por treemaps, bar chart, línea de tiempo Diagramas UML, gráficos circulares, diagramas interactivos entre otros, todos con un total de 2 visualizaciones por tarea.

## 4 Conclusiones

En este trabajo de investigación se han analizado y estudiado 27 de artículos, 16 del año 2009 y 11 del 2014. Luego de una ardua investigación, se puede decir que la pregunta de investigación planteada al inicio del documento fue contestada de manera satisfactoria y se puede llegar a las siguientes conclusiones:

Existe una gran gama de herramientas de visualización de software, sin embargo, no todas son ajustables a las expectativas y necesidades de los desarrolladores, de ahí que las mismas no son utilizadas en forma óptima.

De acuerdo con el resultado de este estudio en particular (27 artículos), la mayoría de los autores limitan sus investigaciones a solo un estudio, sin embargo, aunque en menor medida algunos participan en 2, 3 y 4 estudios, todos relacionados a el campo de investigación.

Contamos con muchos tipos de v fueron los gráficos, seguidos por Treemaps, Bar chart, línea de tiempo, Diagramas UML, gráficos circulares, diagramas interactivos entre otros, los cuales facilitan la comprensión de los programas a los desarrolladores.

Ajustando la opinión de Reiss a esta investigación, indicaremos que se debe trabajar en adecuar las herramientas de Visualización de Software a los programas y sistemas actuales y futuros, si se desea que la investigación en este campo tenga el éxito que se requiere permitiendo a la Visualización de software ser más flexible y que de verdad interactúe con los requerimientos y expectativas de los desarrolladores.

De acuerdo con el análisis de resultados y el estudio, en general realizado a los 27 artículos, concluye que la pregunta planteada al principio de este trabajo fue resuelta en forma satisfactoria.

## References

- Blouin, A., Moha, N., Baudry, B., & Sahraoui, H. (2014). Slicing-based techniques for visualizing large metamodels. *2014 Second IEEE Working Conference on Software Visualization*. pages 15

- González-Torres, A. (2015). *Evolutionary visual software analytics* (Unpublished doctoral dissertation). Universidad de Salamanca. pages 2
- Haugen, B., & akub Kurzak. (2014). Search space pruning constraints visualization. *2014 Second IEEE Working Conference on Software Visualization*. pages 15
- Homer, M., & Noble, J. (2014). Combining tiled and textual views of code. *2014 Second IEEE Working Conference on Software Visualization*. pages 15
- Jääskeläinen, A., Järvinen, H.-M., & Virtanen, H. (2014). Action-based visualization. *2014 Second IEEE Working Conference on Software Visualization*. pages 15
- Kulesz, D., Scheurich, J., & Beck, F. (2014). Integrating anomaly diagnosis techniques into spreadsheet environments. *2014 Second IEEE Working Conference on Software Visualization*. pages 15
- Martinez, J., Ziadi, T., Mazo, R., Bissyande, T. F., Klein, J., & Traon, Y. L. (2014). Feature relations graphs: A visualisation paradigm for feature constraints in software product lines. *2014 Second IEEE Working Conference on Software Visualization*. pages 15
- Noble, J. (2014). Livecoding the synthkit:littlebits as an embodied programming language. *2014 Second IEEE Working Conference on Software Visualization*. pages 15
- (Jääskeläinen, Järvinen, & Virtanen, 2014) (Homer & Noble, 2014) (Blouin, Moha, Baudry, & Sahraoui, 2014) (Kulesz, Scheurich, & Beck, 2014) (Noble, 2014) (Haugen & akub Kurzak, 2014) (?, ?) (Martinez et al., 2014)