

# VISUALIZACIÓN DE LA EVOLUCIÓN DE CALIDAD DE UN SOFTWARE MEDIANTE DIAGRAMAS DE KIVIAT 3D

Juan José López Giraldo, Jose Manuel Ramírez Cruz, Sandra Victoria Hurtado Gil

## RESUMEN

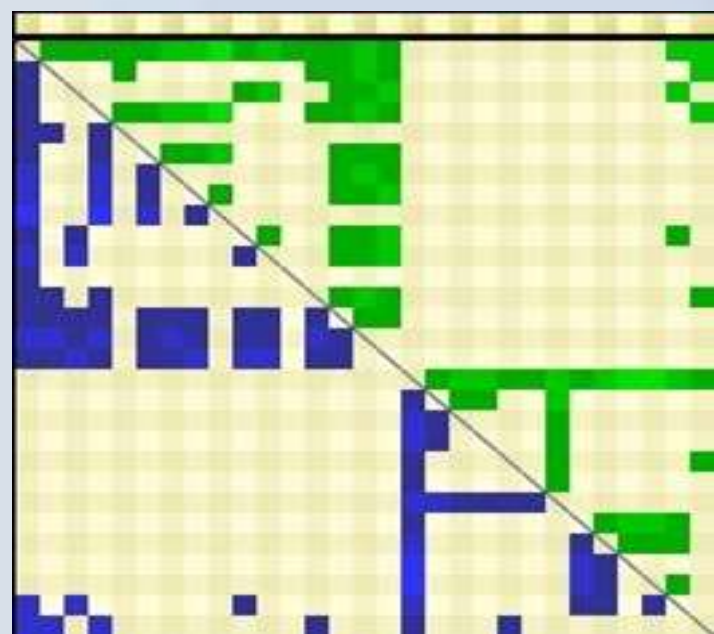
En busca de reducir el tiempo que un ingeniero de software usa para entender las métricas de calidad de su software y su evolución en el tiempo, se desarrollará un aplicativo para visualizar, por medio de un diagrama de Kiviat 3D interactivo, estas métricas en diferentes versiones del código. Se llevará a cabo el proyecto en tres etapas: 1) la caracterización del sistema de visualización de software -estableciendo las métricas-, 2) la implementación del sistema y 3) la validación de la visualización implementada, todo el proceso guiado por la metodología ÁgilUC.

## PALABRAS CLAVE

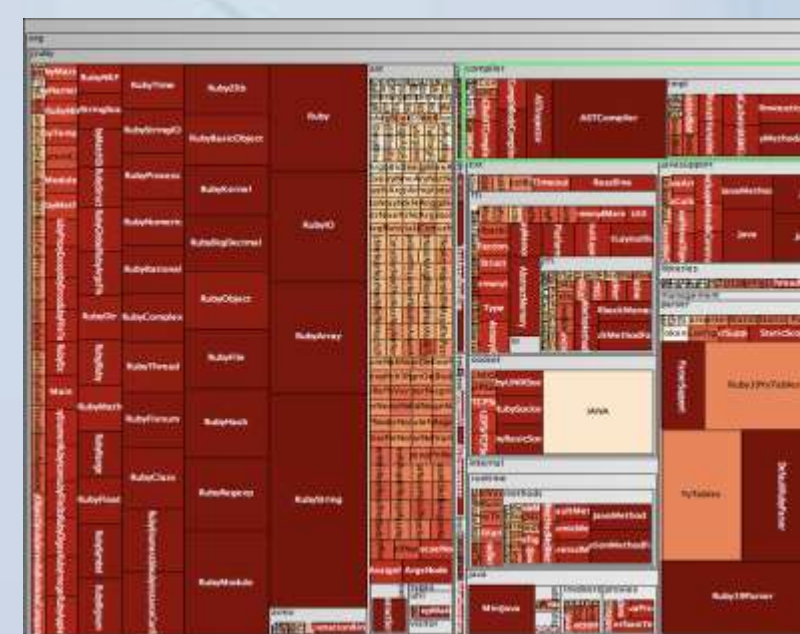
Visualización de información, Análisis estático de código, Diagrama de Kiviat, Evolución del software, Principios de diseño de software, Métricas de calidad de software.

## 1 PLANTEAMIENTO DEL PROBLEMA

Los sistemas actuales de software son cada vez más complejos, lo cual hace que los ingenieros de software ocupen un tiempo considerable en comprender y analizar sistemas en etapas de diseño, desarrollo y mantenimiento [1]. La visualización de datos ha demostrado tener un impacto positivo, ya que ayuda a reducir tiempo y agrega dinamismo a estas tareas[2]. Existen diferentes tipos de visualizaciones de un código, cada una con ventajas y desventajas:



Matriz de dependencias  
(<https://www.ndepend.com/docs/dependency-structure-matrix-dsm>)



Treemap  
(<http://www.softviscollection.org/vis/metrics-treemap/>)

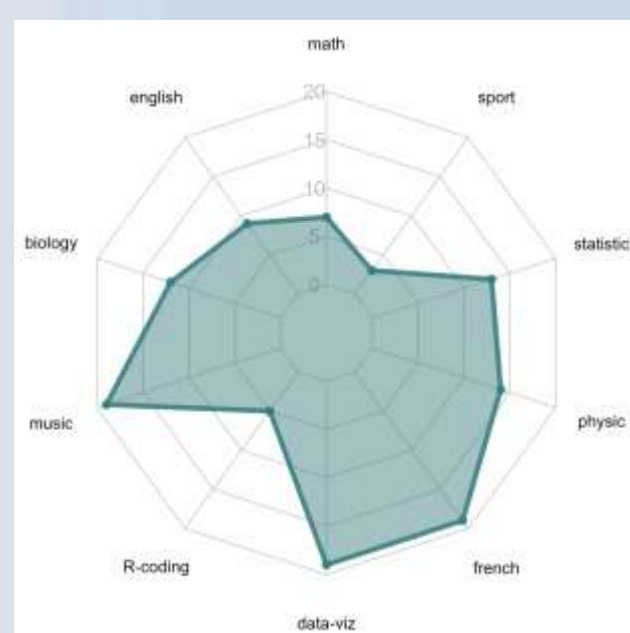


Diagrama de Kiviat  
(<https://www.data-to-viz.com/caveat/spider.html>)

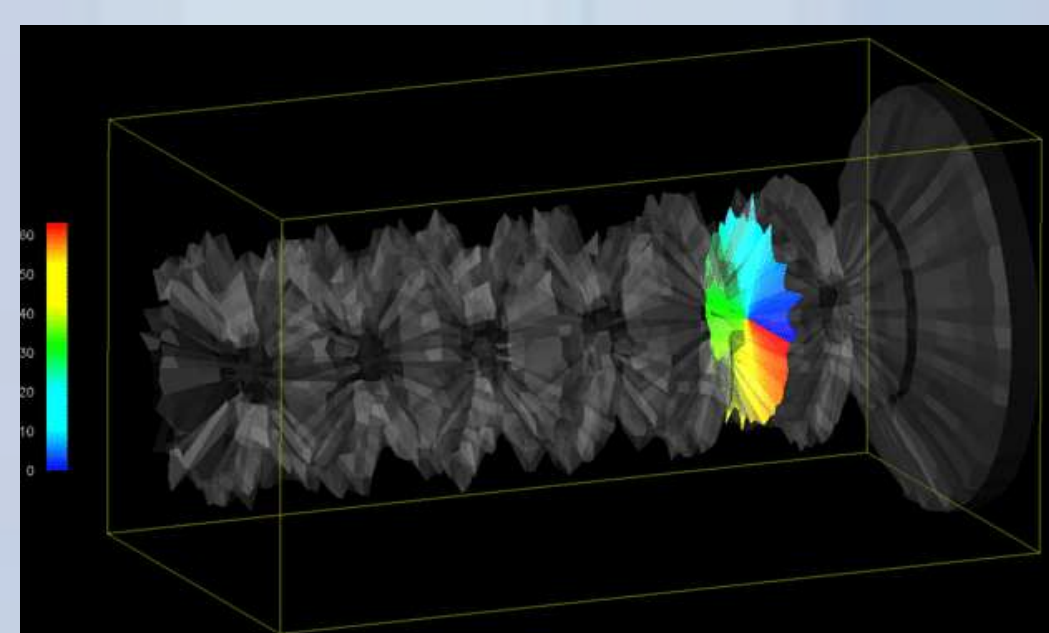


Diagrama de Kiviat 3D [3, 4]

## 2 OBJETIVOS

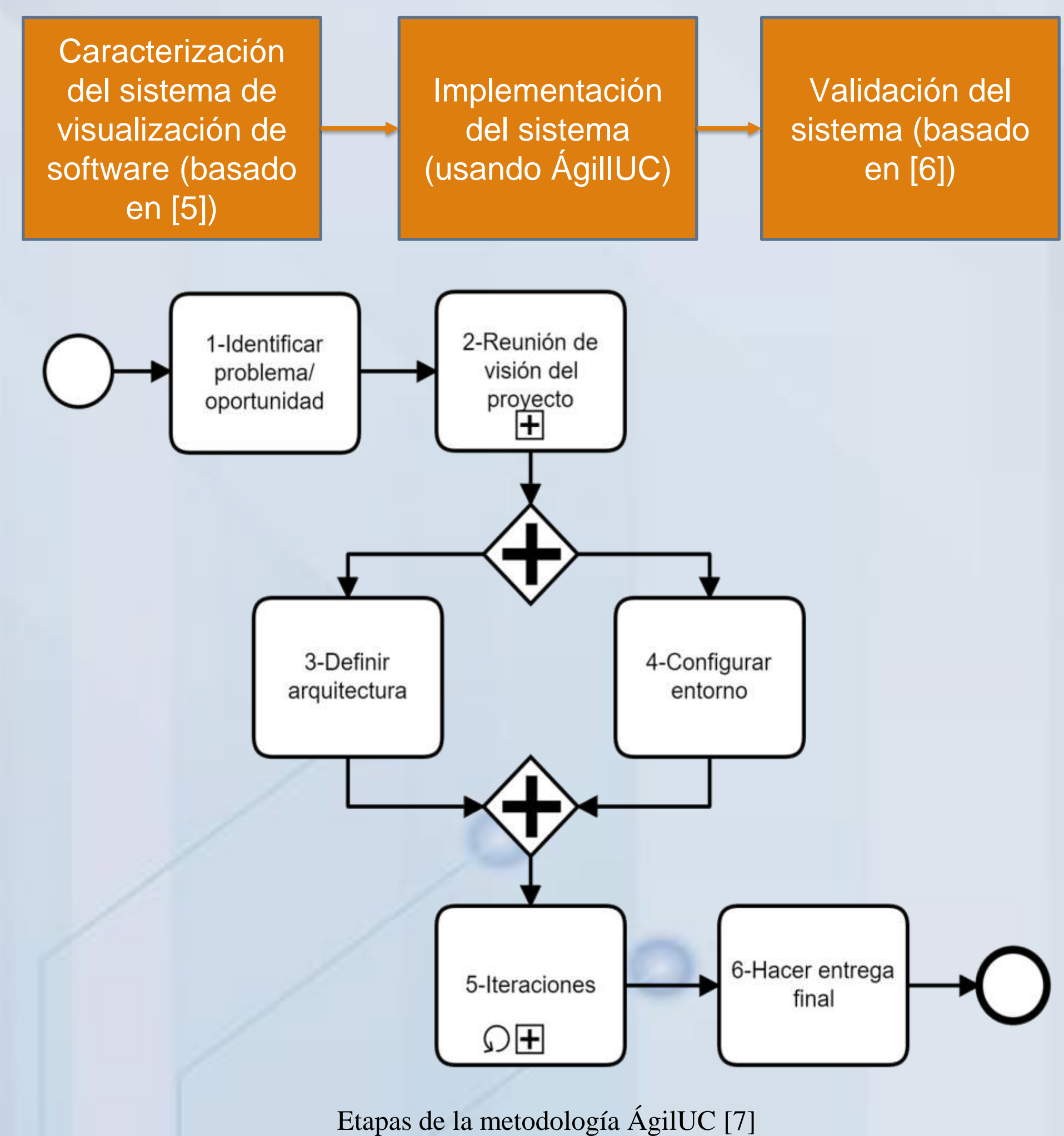
### General

Desarrollar una aplicación para visualizar la evolución de un conjunto de métricas relacionadas con calidad del diseño de un software mediante un diagrama de Kiviat 3D.

### Específicos

- Definir un conjunto de métricas que permita evaluar la calidad del diseño de un software, a partir de los principales principios del diseño.
- Identificar un analizador estático de código que permita obtener datos del código para calcular las métricas.
- Diseñar e implementar la aplicación que visualice las métricas seleccionadas, haciendo uso del analizador identificado.
- Validar la aplicación en sus aspectos de funcionalidad y efectividad.

## 3 METODOLOGÍA



## 4 RESULTADOS ESPERADOS

- Comparación de formatos de visualización.
- Diseño del software (mockups, API, diagramas de clases).
- Software funcional.
- Diseño y resultado de pruebas funcionales.
- Evaluación de la efectividad de la visualización (encuestas o grupos focales)

## 5 REFERENCIAS

1. Brito, F., Goulão, M., & Esteves, R. (1995). Toward the Design Quality Evaluation of Object Oriented Software Systems. Proc. of 5th International Conference on Software Quality.
2. Storey, M. A. D., Wong, K., & Müller, H. A. (2000). How do program understanding tools affect how programmers understand programs? Science of Computer Programming, 36(2), 183–207. [https://doi.org/10.1016/S0167-6423\(99\)00036-2](https://doi.org/10.1016/S0167-6423(99)00036-2)
3. Kerren, A., & Jusufi, I. (2010). 3D Kivat diagrams for the interactive analysis of software metric trends. Proceedings of the ACM Conference on Computer and Communications Security.
4. Hackstadt, S. T., & Malony, A. D. (1995). Visualizing parallel programs and performance. IEEE Computer Graphics and Applications, 15(4), 12-14.
5. Mattila, A. L., Ihantola, P., Kilamo, T., Luoto, A., Nurminen, M., & Väättäjä, H. (2016). Software visualization today - Systematic literature review. Proceedings of the 20th International Academic Mindtrek Conference.
6. Merino, L., et al. (2018). A systematic literature review of software visualization evaluation. Journal of Systems and Software.
7. Hurtado-Gil, S. V. (2019). ÁgilUC: Software development process for small teams and a strategy for its teaching. REVISTA EDUCACION EN INGENIERIA, 15(29), 21–27.