



UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE BOGOTÁ

FACULTAD DE INGENIERÍA
Vicedecanatura Académica
POSGRADOS
PRESENTACIÓN PROPUESTA

TESIS DE DOCTORADO: ☐ TESIS DE MAESTRÍA: ☐
TRABAJO FINAL DE MAESTRÍA: ☒ TRABAJO FINAL DE ESPECIALIZACIÓN ☐

1. PROPONENTE: Joan Sebastian Lopez Riaño CÉDULA: 1020741074
2. PROGRAMA: Maestría de Ingeniería de Sistemas y Computación
3. DIRECTOR PROPUESTO: Felipe Restrepo Calle
DEPARTAMENTO: Departamento de Ingeniería de Sistemas e Industrial
ASESORES:
4. TÍTULO: Metodología para la visualización de la evolución de código fuente
ÁREA: Visualización de Software
LÍNEA DE INVESTIGACIÓN: Ingeniería de Software
5. ANTECEDENTES Y JUSTIFICACIÓN:

6. IDENTIFICACIÓN DEL PROBLEMA:

En ingeniería de *Software* se entiende el concepto metáfora de visualización como “*aquella analogía que subyace a una representación gráfica de una entidad o concepto abstracto con el objetivo de transferir propiedades del dominio específico de la representación gráfica al dominio de la entidad abstracta o concepto que se quiere comprender*” (Diehl, 2007). Existen diversos ejemplos de metáforas de visualización aplicadas al código fuente, de las cuales algunas ya se han establecido como clásicos dentro del campo de estudio, como lo son las metáforas de ciudades (Wettel and Lanza, 2007), hasta otras propuestas más experimentales o artísticas como las de representaciones con plumas de aves (Beck, 2014) o sistemas solares (Graham et al., 2004).

Dependiendo del enfoque de cada autor, cada una de las metáforas encontradas en la literatura tiene unos usos y aplicaciones distintas, pero muchas de las mismas tienen en común el estudio de

artefectos de software en momentos específicos del tiempo, pues no es de su interés el análisis de la evolución y los cambios que pudieron haber afectado al código y en consecuencia a la visualización resultante (Salameh et al., 2016).

Se justifican entonces propuestas visuales que vayan más allá de una “fotografía” en un instante del tiempo de cada artefacto, expandidas con una funcionalidad que permita ver la evolución que ha tenido el código fuente. La aplicación propuesta buscar relacionar distintas áreas de la ingeniería de *Software*, como lo son la visualización de métricas de código fuente y la minería de repositorios abiertos con el propósito de permitir al usuario obtener información en tiempo real de los siguientes tópicos (obtenidos a partir de las visualizaciones):

- Visualización de distintos conjuntos de características estáticas del código fuente.
- Visualización del crecimiento y evolución de los distintos artefactos que componen una aplicación.

7. OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS:

Objetivo general:

Diseñar una metodología que permita visualizar la evolución en el tiempo del código fuente de una aplicación implementando una metáfora visual para la representación de las características de cada artefacto de *software* analizado.

Objetivos específicos:

1. Investigar el estado del arte de trabajos relacionados metodologías para la visualización de Software usando metáforas visuales y que analicen a su vez la evolución del código fuente.
2. Definir la metáfora visual a implementar y el conjunto de características a extraer de los artefactos de código fuente.
3. Desarrollar una aplicación de software que implemente la metodología propuesta y que tomando como entrada un conjunto de artefactos de código fuente produzca una visualización que represente distintas características a analizar de dichos elementos.
4. Validar la metodología desarrollada mediante la aplicación de la misma a algunos proyectos de código abierto que servirán de casos de estudio.

8. METODOLOGÍA:

Estrategia General

Para la efectiva implementación del proyecto se optará por una metodología exploratoria, es decir, que se partirá de una comprensión de la actualidad en el tema de visualización de Software , la

implementación de metáforas y como se ha tratado el aspecto de la naturaleza evolutiva del código fuente.

Con el propósito de llevar a cabo lo definido en el objetivo principal (diseñar una metodología para la visualización de la evolución del código fuente) se definirán un conjunto de posibles propuestas de metáforas visuales las cuales serán validadas mediante la construcción de una aplicación de software por cada una. Finalmente se espera aplicar la metodología propuesta a algunos casos de estudio y a partir de los productos obtenidos poder elaborar y exponer las conclusiones del trabajo realizado.

Los cuatro objetivos específicos se repartirán en cuatro fases de evolución del proyecto, las cuales se pueden resumir como : Levantamiento del estado del arte en el tema de visualizaciones enfocadas en la evolución del software, definición de la metáfora visual y sus características, implementación de la metodología y finalmente validación de la misma mediante algunos casos de estudio.

Fases del proyecto

Fase I: Construcción del estado del arte

Representa el esfuerzo inicial por comprender el estado del arte en el tema de visualizaciones de software, la evolución del mismo y la implementación de metáforas visuales para la representación de las distintas características a analizar, clasificándolas según su relevancia y/o utilidad.

Fase II: Definición de la metáfora visual

Después de llegar a unas conclusiones respecto al estado actual del campo de estudio se planea explorar algunas propuestas para la metáfora final a seleccionar, escogiendo aquella que cuente con las bases más sólidas que justifiquen su usabilidad y pertinencia. De igual manera se definirá el conjunto de características a extraer tanto de los artefactos de código fuente.

Fase III: Implementación de la metodología

Definida la metáfora visual y las características del código fuente que se desea representar, se procederá a la implementación de dicha metodología mediante la construcción de una aplicación que tenga como objetivo generar las visualizaciones sobre los artefactos de código de un proyecto open source como caso de estudio. Para la realización del prototipo se utilizará la metodología de desarrollo *SCRUM*¹ simplificada, la cual, aplicada a un equipo unipersonal con sprints o iteraciones

¹ “...Es un modelo de referencia que definen un conjunto de prácticas y roles, y que pueden tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto” (Schwaber and Beedle, 2001).

semanales permitirá agilizar la generación cada semana de nuevas versiones que se acerquen más a la visualización deseado.

Fase IV: Validación de la metodología

En esta fase final se planea realizar un análisis exploratorio que tenga como objetivo sustentar la justificación de la metáfora visual seleccionada al estudiar distintas visualizaciones generadas por medio de la metodología propuesta aplicada a algunos artefactos de código fuentes que serán tratados como casos de estudio.

9. ACTIVIDADES A DESARROLLAR:

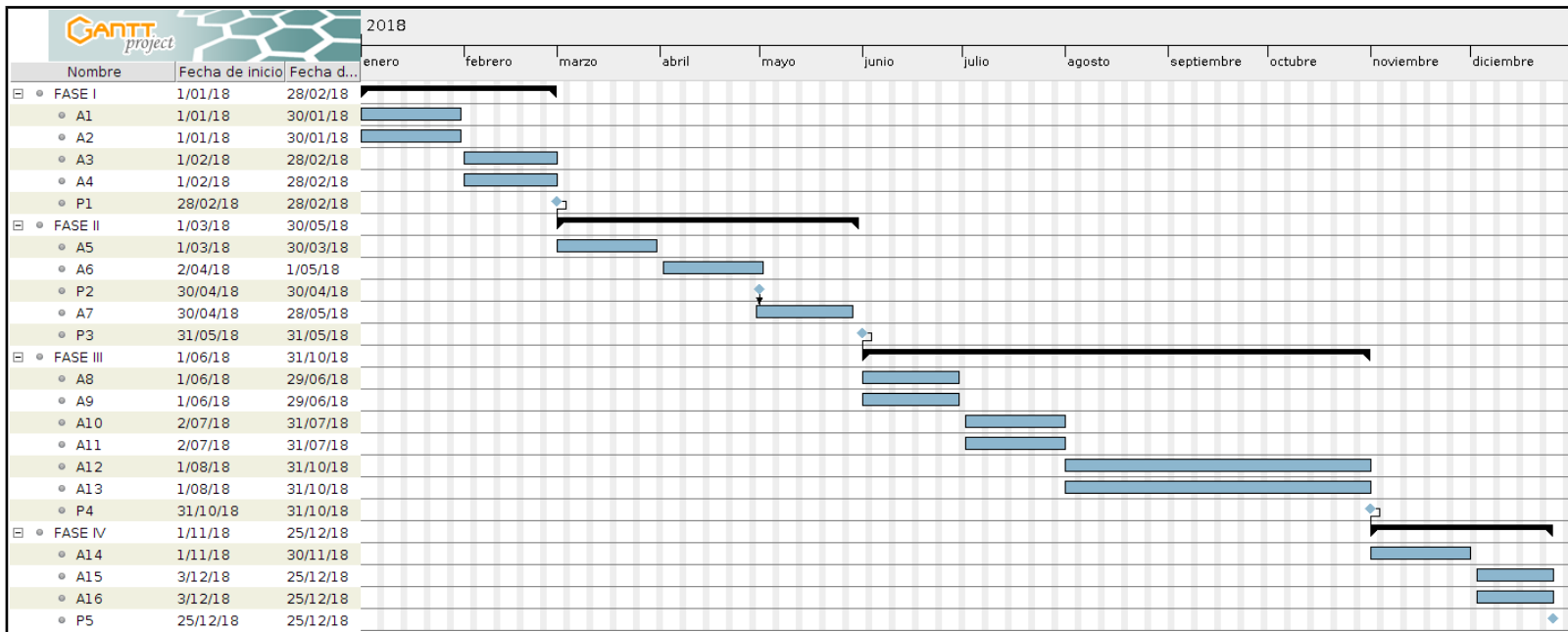
Fase I: Construcción del estado del arte	
OBJETIVO	<i>Investigar el estado del arte de trabajos relacionados metodologías para la visualización de Software usando metáforas visuales y que analicen a su vez la evolución del código fuente</i>
ACTIVIDADES	A1. Definir un marco teórico mínimo para las temáticas de visualización de software. A2. Realizar un filtrado inicial de referencias recientes en el campo de la visualización de software. A3. Realizar un segundo filtrado de referencias para seleccionar los proyectos relacionados con metáforas del código fuente. A4. Elaborar las conclusiones del estado del arte.
PRODUCTOS	P1. Reporte técnico del marco teórico y estado del arte del proyecto

Fase II: Definición de la metáfora visual	
OBJETIVO	<i>Definir la metáfora visual a implementar y el conjunto de características a extraer de los artefactos de código fuente.</i>
ACTIVIDADES	A5. Escoger y clasificar un conjunto inicial de características relacionadas con la evolución del software de acuerdo a las conclusiones del estado del arte. A6. Proponer y desarrollar varias propuestas de metáforas visuales para la representación de las características anteriormente seleccionadas. A7. Diseñar ilustraciones de ejemplo sobre las visualizaciones propuestas aplicadas a características extraídas de algunos artefactos de código fuente.
PRODUCTOS	P2. Reporte técnico de la descripción de la metáfora a implementar y de las otras posibles propuestas descartadas. P3. Reporte técnico de la propuesta de la metáfora principal, utilidad y aplicaciones.

Fase III: Implementación de la metodología	
OBJETIVO	<i>Desarrollar una aplicación de software que implemente la metodología propuesta y que tomando como entrada un conjunto de artefactos de código fuente produzca una visualización que represente distintas características a analizar de dichos elementos.</i>
ACTIVIDADES	<p>A8. Documentar los casos de uso y diagramas de flujo de la aplicación a desarrollar.</p> <p>A9. Escoger la tecnología a implementar en la propuesta de visualización y conexiones con los repositorios de código fuente.</p> <p>A10. Diseñar la arquitectura y el modelado de información para la generación de las visualizaciones.</p> <p>A11. Implementar el diseño (desarrollo de la aplicación).</p> <p>A12. Realización de iteraciones semanales que generen productos mínimos viables para la generación de las visualizaciones deseadas.</p> <p>A13. Realización de controles semanales de lo que se ha hecho y lo que no, por cada versión del prototipo liberada.</p>
PRODUCTOS	P4. Prototipo de software por cada iteración realizada.

Fase IV: Validación de la metodología	
OBJETIVO	<i>Validar la metodología desarrollada mediante la aplicación de la misma a algunos proyectos de código abierto que servirán de casos de estudio</i>
ACTIVIDADES	<p>A14. Definir un grupo delimitado artefactos de código fuente que pertenezcan a un mismo proyecto del tipo <i>open source</i>.</p> <p>A15. Generar por cada grupo de artefactos una visualización usando la metodología propuesta.</p> <p>A16. Generar un listado de conclusiones respecto a la visualización obtenida.</p>
PRODUCTOS	P5. Reporte técnico con la descripción del proceso de validación y de conclusiones respecto a posibles usos y aplicaciones de la metodología propuesta.

10. CRONOGRAMA:



11. BIBLIOGRAFÍA BÁSICA:

Beck, F., 2014. Software Feathers figurative visualization of software metrics, in: 2014 International Conference on Information Visualization Theory and Applications (IVAPP). Presented at the 2014 International Conference on Information Visualization Theory and Applications (IVAPP), pp. 5–16.

Diehl, S., 2007. Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software. Springer Science & Business Media.

Graham, H., Yang, H.Y., Berrigan, R., 2004. A Solar System Metaphor for 3D Visualisation of Object Oriented Software Metrics, in: Proceedings of the 2004 Australasian Symposium on Information Visualisation - Volume 35, APVis '04. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 53–59.

Salameh, H.B., Ahmad, A., Aljammal, A., 2016. Software evolution visualization techniques and methods - a systematic review, in: 2016 7th International Conference on Computer Science and Information Technology (CSIT). Presented at the 2016 7th International Conference on Computer Science and Information Technology (CSIT), pp. 1–6. doi:10.1109/CSIT.2016.7549475

Schwaber, K., Beedle, M., 2001. Agile Software Development With Scrum. Prentice Hall, Upper Saddle River, NJ.

Wettel, R., Lanza, M., 2007. Visualizing Software Systems as Cities, in: 2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis. Presented at the 2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis, pp. 92–99. doi:10.1109/VISSOF.2007.4290706

12. RECURSOS FÍSICOS:

- Equipo de cómputo para desarrollo, redacción y acceso a la Internet (Propio del investigador)
- Equipo de cómputo para el desarrollo de la etapa de experimentación (Propio del investigador)
- Fotocopias y material de papelería.
- Artículos y libros de referencia (Facilitados por la división de bibliotecas de la Universidad Nacional de Colombia)

13. COSTOS DEL TRABAJO Y FUENTES DE FINANCIACIÓN:

La principal fuente de financiamiento para el proyecto propuesto sera mixta, es decir que estará compuesta de recursos propios del autor y de parte de los recursos que provee la Universidad Nacional de Colombia en el convenio que como estudiante de la Maestría de Ingeniería de Sistemas y Computación el autor es beneficiario. La siguiente descripción de gastos se aplica para un cálculo anual, es decir durante la duración de los doce meses de ejecución del proyecto. La estimación de costos de personal resulta de un estimado en la experiencia del costo de un desarrollador junior (para el caso del investigador) y de estimados en los gastos de profesores. Otros rubros como el gasto de Internet y equipos son también estimados con base en la experiencia y tasas actuales de prestadores de servicios públicos.

CONCEPTO	FUENTE	TOTAL
GASTOS DE PERSONAL		
Salario investigador	Investigador	\$26.400.000,00
Salario Director	Universidad Nacional de Colombia	\$18.000.000,00
EQUIPOS Y SOFTWARE		
Computador de escritorio	Investigador	\$2.300.000,00
MATERIALES DE SUMINISTRO Y BIBLIOGRAFÍA		
Conexión a Internet	Investigador	\$960.000,00
Bibliografía y acceso a artículos académicos	Universidad Nacional de Colombia	\$1.000.000,00
TOTAL		\$48.660.000,00

14. COMENTARIO CON VISTO BUENO DEL DIRECTOR:

15. FIRMA DEL PROPONENTE

Joan Sebastian Lopez Riaño

16. FIRMA DEL DIRECTOR (ASESORES)

Felipe Restrepo Calle PhD

17. FECHA