

# EVALUACION DE MECANISMOS DE COOPERACION ENTRE GENERACIONES DE DESARROLLADORES EN EL PROCESO DE DESARROLLO DE SOFTWARE LIBRE

Fernando Castillo Zapata<sup>1</sup>, Jorge Andrick Parra Valencia<sup>2</sup>

<sup>1</sup>Ingeniero de sistemas, Universidad Autónoma del Caribe – Colombia, Maestría en software libre Convenio universidad abierta de Cataluña España – Universidad autónoma de Bucaramanga. Correo electrónico: ingfcastillo@gmail.com

<sup>2</sup>Ph.D., M.Sc., SE., Profesor titular programa de Ingeniería de sistemas, Investigador asociado Grupo de investigación en pensamiento sistémico. Presidente comunidad Colombiana de dinámica de sistemas 2011-2012, 2012-2013-2014. Universidad Autónoma de Bucaramanga, avenida 42 No. 48 – 11, Bucaramanga, Colombia. Correo electrónico: andrick.parra@gmail.com

## RESUMEN

Las generaciones de desarrolladores de software libre, presentan relaciones complejas que determinan sus inclinaciones para iniciar o continuar proyectos de software libre. Estas relaciones intergeneracionales definen un nivel de cooperación fundamental para el crecimiento y el aumento de calidad del software libre. Actualmente no es claro como la cooperación entre generaciones de desarrolladores de software libre afecta el desarrollo de software libre [12]. En la búsqueda de propuesta de solución, se aplicaron los lineamientos metodológicos de la Dinámica de sistemas para entender como la cooperación basada en confianza, puede afectar la acción colectiva de las generaciones de desarrolladores de software libre. Los resultados obtenidos explican como la complejidad dinámica puede dominarse para gestionar la cooperación entre generaciones de desarrolladores de software de forma efectiva. Como conclusión final se obtuvo que los altos tiempos de redesarrollo de software afectan directamente la efectividad de la cooperación basada en confianza.

**Palabras Clave:** Cooperación, dinámica de sistemas, generaciones de desarrolladores, mecanismos de cooperación, software libre.

Recibido: 11 de Agosto de 2014.  
Received: August 11<sup>th</sup>, 2014.

Aceptado: 25 de Mayo de 2015.  
Accepted: May 25<sup>th</sup>, 2015.

## EVALUATION OF COOPERATION MECHANISMS BETWEEN GENERATIONS OF DEVELOPERS IN THE DEVELOPMENT PROCESS OF FREE SOFTWARE

### ABSTRACT

*Generations of free software developers have complex relationships that determine their inclinations to start or continue free software projects. These intergenerational relationships define a level of cooperation vital for growth and increased quality of free software. Currently, it is unclear how cooperation between generations of free software developers affects the development of the free software. Methodological guidelines for system dynamics to understand how cooperation based on trust can affect the collective action of the generations of free software developers were applied. The results explain how dynamic complexity can be mastered for managing cooperation between generations of software developers effectively. The final conclusion obtained was that high times of software redevelopment affects the effectiveness of cooperation based on trust.*

**Keywords:** Cooperation, cooperation mechanisms, free software, generations of developers, system dynamics.

## 1. INTRODUCCION.

La cooperación entre desarrolladores de nuevos proyectos de software libre y los desarrolladores de nuevas versiones de proyectos de software libre, que conforman las diferentes generaciones de software, es vital para el aumento de la cantidad y calidad del software libre.

Sin embargo es limitado el conocimiento sobre la cooperación entre generaciones de desarrolladores de software libre. Este artículo estudia el efecto de la promoción de la acción colectiva entre generación de desarrolladores de software en el desarrollo de software libre.

A continuación se presenta una referencia al problema de la cooperación desde la perspectiva de los dilemas sociales, luego se detallan algunos aspectos que supone la cooperación entre generaciones de desarrolladores de software libre, y finalmente se indica la brecha que será tratada en este artículo.

### 1.1. Dilemas Sociales

Los dilemas sociales son conflictos que los grupo de individuos enfrentan comúnmente, tomando decisiones entre beneficio individual o colectivo [1]. En dichos dilemas sociales todos los miembros del grupo tendrían mejores beneficios si cooperan.

La disponibilidad del recurso y la realimentación del estado del recurso juegan un papel fundamental en la decisión de cooperar. Además, los dilemas sociales contemplan dos aspectos: Deserción de individuos (no cooperar), Decisión propia y personal de contribuir y la medida de esa contribución. Estos aspectos generan toda una estructura de conflictos: Conflictos en el nivel, Conflictos en la localización, Conflictos en la distribución.

Los conflictos generados por los dilemas sociales pueden afectar el desempeño de los grupos pequeños para manejar los recursos compartidos [2] y la capacidad de sostenibilidad de los recursos de gran escala como la atmosfera [8]. La cooperación es factible para enfrentar dilemas sociales a pequeña escala [9] [10].

El retardo en la percepción de los efectos de la cooperación en el desarrollo de software libre afecta la cooperación basada en confianza. [13]

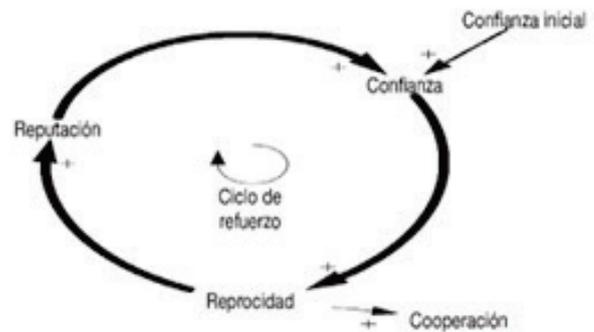


Fig. 1. Versión dinámica genérica del mecanismo de cooperación basado en confianza.

Una versión dinámica del mecanismo basado en confianza promueve reciprocidad. Después, la reputación es afectada por la reciprocidad.

Más reciprocidad produce más reputación e incrementa la cooperación. Finalmente, la reputación mejora la confianza. En términos dinámicos, las condiciones iniciales [3] para la confianza afectan el comportamiento de la cooperación porque estas variables básicas (confianza, reputación y cooperación) están unidas en un ciclo de retroalimentación de refuerzo, que refuerza cualquier condición inicial. Este es el caso con el mecanismo de cooperación basado en confianza que presenta una dependencia a las condiciones iniciales.

*Problemas de la cooperación basada en confianza.* La dependencia a las condiciones iniciales y los retardos en la información de los resultados de la cooperación, son los problemas identificados para la promoción de la acción colectiva efectiva entre generaciones de desarrolladores de software en el proceso de desarrollo de software libre. No hay una unificación de conceptos a cerca de la posibilidad de aplicar cooperación basada en confianza en situaciones sociales a gran escala [11].

La anterior circunstancia ha sido descrita por [6] para varias situaciones de cooperación alrededor del cambio climático y crisis de electricidad. Si las generaciones no disponen de información sobre los resultados de su cooperación, estos podrían dejar de cooperar. Esta situación ha sido estudiada para fenómenos de alta inercia como los que tienen que ver con la limitación de gases de alta inercia en la atmosfera tales como el CO<sub>2</sub> [7] y el ozono.

## 1.2. Gestión de la Cooperación

Este trabajo supone que la cooperación es posible pero su aparición supone una complejidad dinámica. Por lo tanto, se propone estudiar la dinámica que supone la promoción de una cooperación efectiva que supere los problemas estructurales y dinámicos que definen mecanismos de cooperación como el mecanismo basado en confianza [13]. La gestión de la cooperación supone la articulación de reglas, estrategias, normas y mecanismos que permitan mejorar las condiciones para que la dinámica de la cooperación produzca acción colectiva efectiva en la solución de dilemas sociales en grupos. Así, es posible gestionar la cooperación en dilemas sociales a través de mecanismos adicionales combinados con la cooperación basada en la confianza, para hacer frente a la dependencia de las condiciones iniciales. Para evaluar esta afirmación, se han desarrollado modelos de simulación para representar el proceso de desarrollo de software inter generacional integrando el mecanismo de cooperación basado en confianza.

## 1.3. Dilema de Confianza

El dilema de la confianza es un concepto central de la teoría de la acción colectiva. Este dilema explica el proceso de evaluar el riesgo que un individuo asume al confiar en que un actor realizará una acción de cierta manera [13].

## 1.4. Déficit de Investigación

Luego de la revisión realizada, no se han encontrado trabajos que ofrezcan elementos para una gestión dinámica de la cooperación entre generaciones de desarrolladores de software libre. Este tipo de investigaciones es pertinente debido que la calidad de la cooperación entre generaciones de desarrolladores de software es clave para aumentar la cantidad y calidad del software libre.

## 2. MATERIALES Y METODO

Para el desarrollo de los modelos se aplicaron los lineamientos metodológicos de la Dinámica de Sistemas [4] [5]. Los pasos más importantes son:

-Hipótesis dinámica que explica como el mecanismo de cooperación desarrollado se integra al proceso de desarrollo de software libre entre generaciones de desarrolladores de software.

-Modelo matemático del proceso de desarrollo de software y del funcionamiento del mecanismo de cooperación integrado.

-Simulaciones para evaluar la efectividad del mecanismo de cooperación en la promoción y el sostenimiento de la cooperación.

Se aplicó la Dinámica de Sistemas para desarrollar hipótesis dinámicas que representan como la cooperación afecta el desarrollo de software libre y de esta forma evaluar la acción colectiva efectiva del mecanismo de cooperación entre las generaciones de desarrolladores de software libre. Los modelos se desarrollaron usando Vensim 6.1 para Windows.

## 3. RESULTADOS

Se presentan inicialmente la hipótesis dinámica, el modelo y la simulación correspondiente.

### 3.1. Hipótesis Dinámica

Se planteó una hipótesis dinámica que expone que a mayor confianza entre generaciones de desarrolladores de software menor tiempo promedio de re-desarrollo y así mismo mayores aplicaciones en re-desarrollo. El termino re-desarrollo hace referencia al proceso de generar nuevas versiones de un software libre determinado. Las aplicaciones en re-desarrollo son aquellas aplicaciones obsoletas que fueron seleccionadas para actualizarlas y/o adaptarlas. Las aplicaciones obsoletas son aplicaciones en uso que por un factor de uso y de tiempo quedan con una funcionalidad limitada con respecto a los requerimientos del mercado. Las aplicaciones en uso pueden ser derivadas de una aplicación nueva o una aplicación proveniente de re-desarrollo.

En este artículo se realizó una desagregación de la confianza desde el concepto de dilema de confianza. Un dilema de confianza supone el decidir desde la percepción de los costos entre la vulnerabilidad y oportunidad que supone el cooperar con base en la confianza.

Si la confianza no tiene la suficiente calidad en términos de su correspondencia con un comportamiento cooperativo esperado en su respuesta, se genera una vulnerabilidad. Si la confianza es de la suficiente calidad, se genera una oportunidad de cooperación. La versión inicial de la hipótesis dinámica:



Fig. 2a. Hipótesis dinámica. Propone la integración del mecanismo de cooperación basado en confianza con la cooperación entre generaciones de desarrolladores de software.

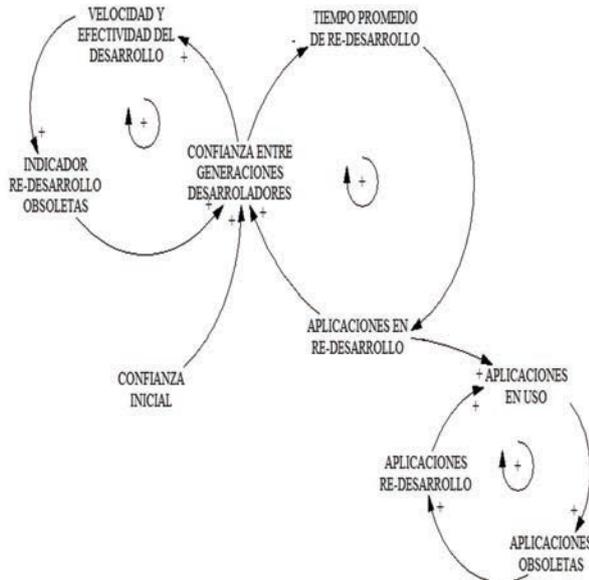


Fig. 2b. Hipótesis dinámica que propone la integración del mecanismo de cooperación basado en confianza con la cooperación entre generaciones de desarrolladores de software de una manera más ampliada.

### 3.2. Modelo de Simulación

El modelo de simulación presenta la confianza entre generaciones de desarrolladores de software como un nivel que se alimenta de la percepción retardada de las aplicaciones en re-desarrollo. La confianza degrada mediante una vida media. La confianza afecta el re-desarrollo que a su vez afecta las aplicaciones en uso. El ciclo se define como del tipo refuerzo.

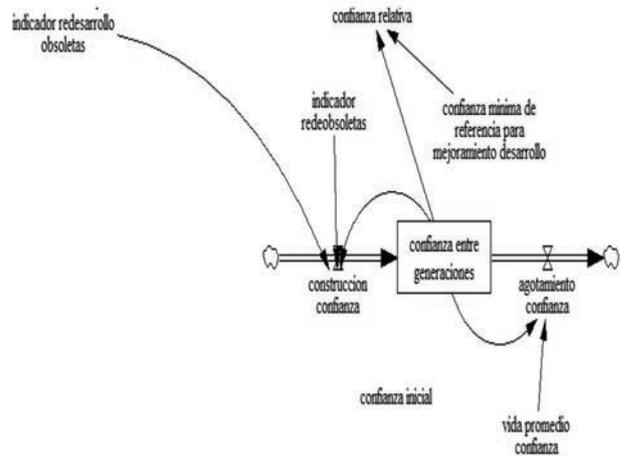


Fig. 3a. Parte 1 de 3. Modelo que propone la integración del mecanismo de cooperación basado en confianza con cooperación entre generaciones de desarrolladores de software libre.

Finalmente, se ilustra como las cantidades de aplicaciones obsoletas se aumentan a medida que aumentan las aplicaciones en uso. La cantidad de desarrolladores nuevos que se convierten a desarrolladores de re-desarrollos combinado con los desarrolladores de re-desarrollos, teniendo en cuenta la productividad de las dos generaciones aumentan el indicador de re-desarrollo de obsoletas que a su vez aumenta la construcción de la confianza.

Ecuación diferencial asociada a la variable de nivel.

$$\frac{d(\text{confianza entre generaciones})}{dt} = \text{construcción confianza}(t) - \text{agotamiento confianza}(t)$$

Para las variables de nivel, se presentan las siguientes ecuaciones diferenciales:

$$\frac{d(\text{desarrolladores activos nuevas versiones})}{dt} = \text{transito de RE a NUEVO}(t) - \text{desertores nuevos desarro}(t) - \text{transito de NUEVO a RE}$$

$$\frac{d(\text{desarrolladores de redesarrollos})}{dt} = \text{transito de NUEVO a RE}(t) - \text{desertores}(t) - \text{retiro normal}(t) - \text{transito de RE a nuevos}(t)$$

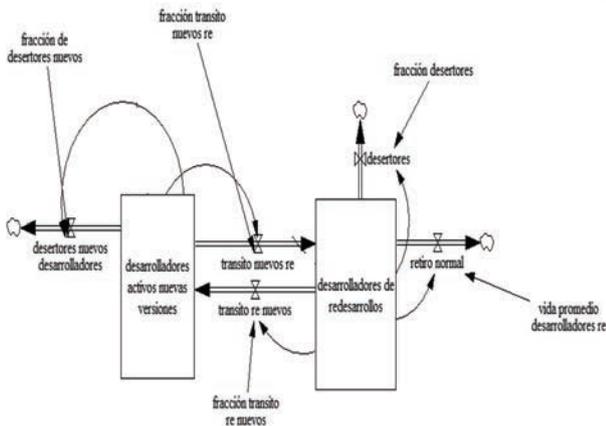
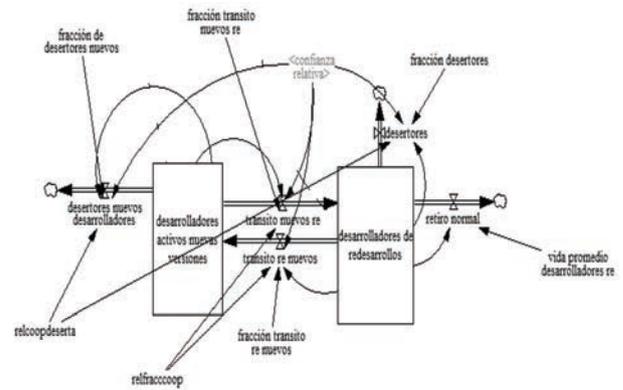


Fig. 3a. Parte 2 de 3. Modelo que propone la integración del mecanismo de cooperación basado en confianza con cooperación entre generaciones de desarrolladores de software libre. Generaciones de desarrolladores.



1. Confianza relativa
2. rel coop deserta
3. rel facc coop
4. Vida promedio desarrolladores de redesarrollo

Fig. 3b. Parte 1 de 2. Nuevo modelo que propone la integración del mecanismo de cooperación basado en confianza con cooperación entre generaciones de desarrolladores de software libre.

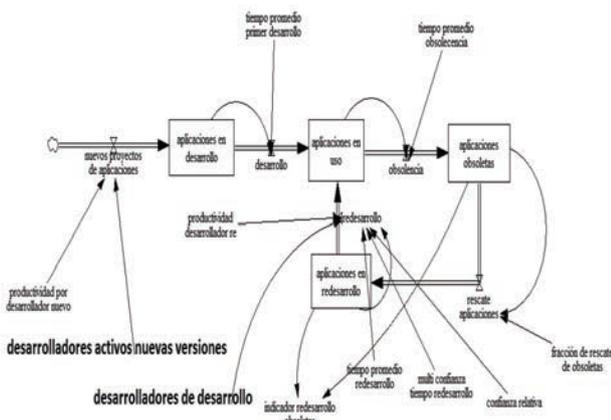


Fig. 3a. Parte 3 de 3. Modelo que propone la integración del mecanismo de cooperación basado en confianza con cooperación entre generaciones de desarrolladores de software libre. Aplicaciones.

Ecuación diferencial asociada a la variable de nivel.

$$\frac{d(\text{aplicaciones en desarrollo})}{dt} = \text{nuevos proyectos de aplicaciones}(t) - \text{desarrollo}(t)$$

$$\frac{d(\text{aplicaciones en uso})}{dt} = \text{desarrollo}(t) + \text{redesarrollo}(t) - \text{obsolescencia}(t)$$

$$\frac{d(\text{aplicaciones obsoletas})}{dt} = \text{obsolescencia}(t) - \text{rescate aplicaciones}(t)$$

$$\frac{d(\text{aplicaciones en redesarrollo})}{dt} = \text{rescate aplicaciones}(t) - \text{redesarrollo}(t)$$

Ecuación diferencial asociada a la variable de nivel.

$$\frac{d(\text{desarrolladores activos nuevas versiones})}{dt} = \text{transito de RE a NUEVO}(t) - \text{desertores nuevos desarro}(t) - \text{transito de NUEVO a RE}$$

$$\frac{d(\text{desarrolladores de redesarrollos})}{dt} = \text{transito de NUEVO a RE}(t) - \text{desertores}(t) - \text{retiro normal}(t) - \text{transito de RE a nuevos}(t)$$

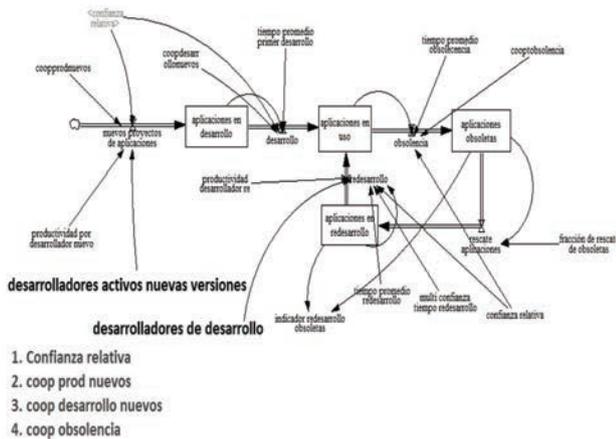


Fig. 3b. Parte 2 de 2. Nuevo modelo que propone la integración del mecanismo de cooperación basado en confianza con cooperación entre generaciones de desarrolladores de software libre. Aplicaciones.

Para las variables de nivel, se presentan las siguientes ecuaciones diferenciales:

$$\frac{d(\text{aplicaciones en desarrollo})}{dt} = \text{nuevos proyectos de aplicaciones} - \text{desarrollo}(t)$$

$$\frac{d(\text{aplicaciones en uso})}{dt} = \text{desarrollo}(t) + \text{redesarrollo}(t) - \text{obsolescencia}(t)$$

$$\frac{d(\text{aplicaciones obsoletas})}{dt} = \text{obsolescencia}(t) - \text{rescate aplicaciones}(t)$$

$$\frac{d(\text{aplicaciones en redesarrollo})}{dt} = \text{rescate aplicaciones}(t) - \text{redesarrollo}(t)$$

### 3.3. Resultado de la Simulación

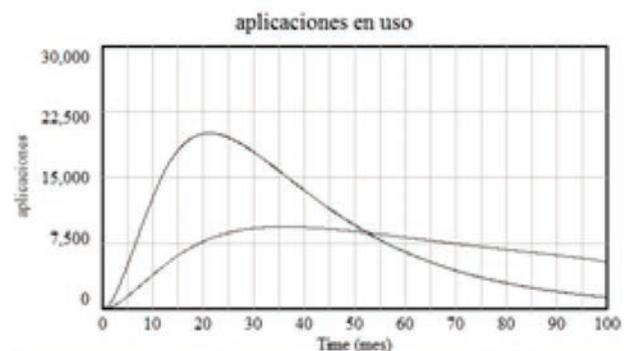
Durante el proceso de re-desarrollo de aplicaciones se va aumentando la confianza entre generaciones de desarrolladores de software. La confianza relativa y teniendo en cuenta las generaciones de desarrolladores de re-desarrollo, la generaciones de desarrolladores de desarrollo que hacen tránsito a desarrollar re-desarrollo y la productividad de cada generación de desarrolladores, aumentan las aplicaciones en uso.

Una mayor cantidad aplicaciones en uso genera una mayor cantidad de aplicaciones obsoletas, desde la cual se pueden obtener un alto índice de

aplicaciones que entran en etapa de re-desarrollo, de esta manera aumenta la confianza entre generaciones de desarrolladores de software.

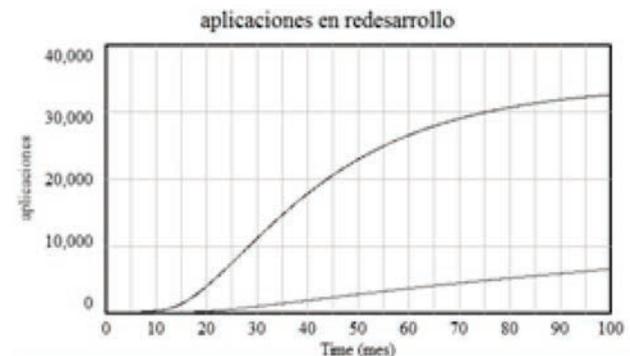
$$\frac{d(\text{aplicaciones en uso})}{dt} = \text{desarrollo}(t) + \text{redesarrollo}(t) - \text{obsolescencia}(t)$$

$$\frac{d(\text{aplicaciones en redesarrollo})}{dt} = \text{rescate aplicaciones}(t) - \text{redesarrollo}(t)$$



Valor de 1 mes  
 Valor de 50 meses

Figura 4a. Resultados para niveles altos y bajo de confianza inicial en referencia a las aplicaciones en uso. Conclusión: a mayor confianza inicial mayor cantidad de aplicaciones en uso.



Valor de 1 mes  
 Valor de 50 meses

Figura 4b. Resultados para niveles altos y bajo de confianza inicial en referencia a las aplicaciones en re-desarrollo. Conclusión: a mayor confianza inicial mayor cantidad de aplicaciones en re-desarrollo.

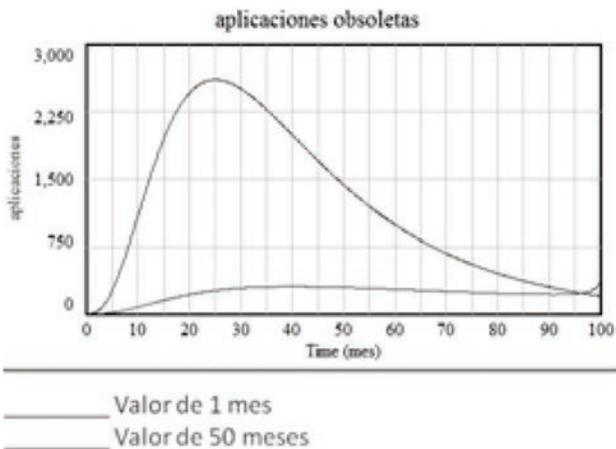


Fig. 4c. Resultados para niveles altos y bajo de confianza inicial en referencia a las aplicaciones obsoletas. Conclusión: a mayor confianza inicial mayor cantidad de aplicaciones obsoletas

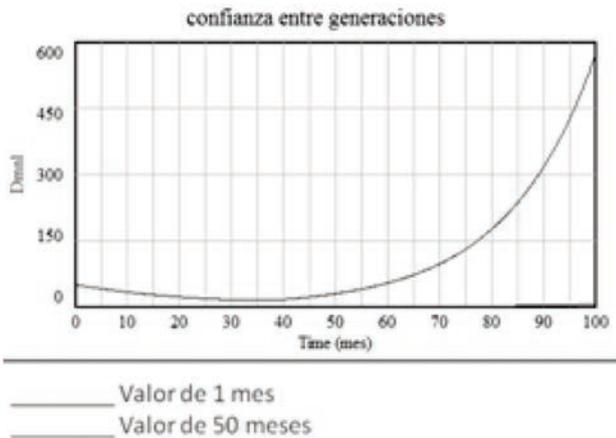


Fig. 4d Resultados para niveles altos y bajos de confianza inicial en referencia a la confianza entre generaciones.

$$\frac{d(\text{aplicaciones obsoletas})}{dt} = \text{obsolescencia}(t) - \text{rescate aplicaciones}(t)$$

$$\frac{d(\text{confianza entre generaciones})}{dt} = \text{construcción confianza}(t) - \text{agotamiento confianza}(t)$$

### 3.4. Análisis de Sensibilidad

Se realizó un análisis de sensibilidad consistente en 200 simulaciones que variaron de manera uniforme la confianza inicial. Las siguientes figuras presentan los espacios de probabilidad para el parámetro confianza inicial para diferentes variables descritas.

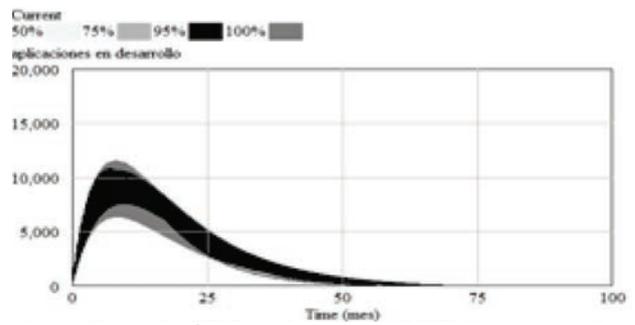


Fig. 5a. Análisis de sensibilidad para las aplicaciones en desarrollo para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

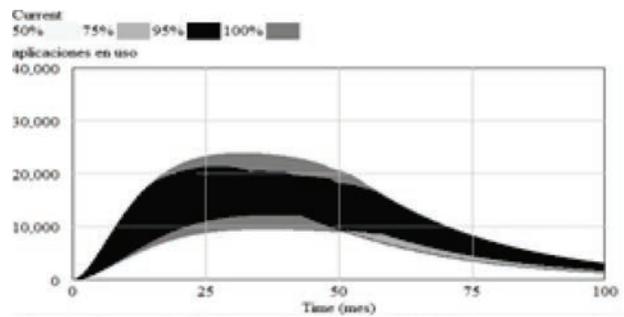


Fig. 5b. Análisis de sensibilidad para las aplicaciones en uso para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

El análisis de sensibilidad demuestra que la confianza depende de las condiciones iniciales [3] de la confianza de cooperación lo cual afecta la confianza entre generaciones. A mayor confianza inicial mayor confianza entre generaciones, debido al mejoramiento de la acción colectiva entre generaciones de desarrolladores de software.

$$\frac{d(\text{desarrolladores de redesarrollos})}{dt} = \text{transito de NUEVO a RE}(t) - \text{desertores}(t) - \text{retiro normal}(t) - \text{transito de RE a nuevos}(t)$$

$$\frac{d(\text{aplicaciones en uso})}{dt} = \text{desarrollo}(t) + \text{redesarrollo}(t) - \text{obsolescencia}(t)$$

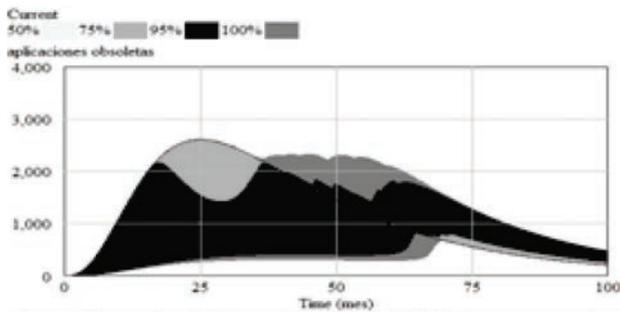


Fig. 5c. Análisis de sensibilidad para las aplicaciones obsoletas para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

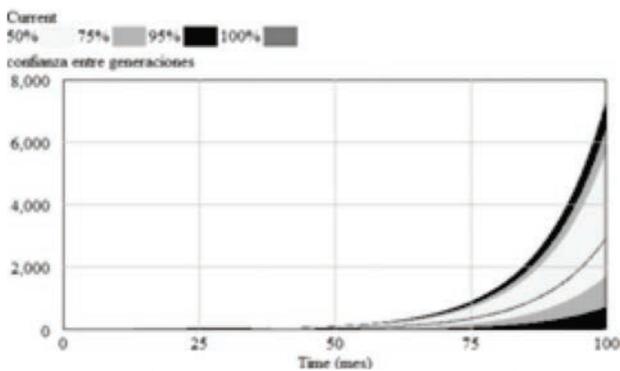


Fig. 5d. Análisis de sensibilidad para la Confianza entre generaciones para el Mecanismo de cooperación basado en confianza en el proceso de cooperación entre Generaciones de desarrolladores de software Libre.

$$\frac{d(\text{aplicaciones obsoletas})}{dt} = \text{obsolescencia}(t) - \text{rescate aplicaciones}(t)$$

$$\frac{d(\text{confianza entre generaciones})}{dt} = \text{construcción confianza}(t) - \text{agotamiento confianza}(t)$$

#### 4. DISCUSION

La cantidad de aplicaciones en re desarrollo genera dinamismo en el ciclo de vida de los desarrollos de software libre, permitiendo que aplicaciones en uso rápidamente se conviertan en obsoletas y de este último grupo se rescatan muchos desarrollos para generar nuevas aplicaciones basadas en

desarrollos conocidos. Este ciclo genera confianza entre las generaciones de desarrolladores, con lo cual se da la cooperación. Cuando aumenta la cooperación aumentan las aplicaciones en desarrollo, aumentan las aplicaciones obsoletas y aumentan las aplicaciones en re desarrollo.

Los análisis previos nos indican la forma en que la cooperación nos permite que desarrollemos más software y obtengamos más aplicaciones obsoletas. El anterior resultado fue alcanzado aplicando los lineamientos metodológicos de la dinámica de sistemas, con los cuales se demuestra que la gestión dinámica de la cooperación inter generacional basada en el mecanismo de la confianza puede ser efectiva y sostenible.

Ostrom [2] muestra el mecanismo de cooperación basado en confianza, pero no menciona el efecto sobre la cooperación entre generaciones de desarrolladores de software libre. Castillo menciona el problema de la dependencia a las condiciones iniciales [3] del mecanismo de cooperación basado en confianza, pero no evalúa el mejoramiento de la acción efectiva entre generaciones de desarrolladores de software.

Parra y Dyer han reportado el problema en dilemas sociales de gran escala tales como la acumulación de CO2 y el deterioro del OZONO estratosférico [7], pero no en el desarrollo de software libre. Lo anterior supone que este hallazgo es un aporte al área.

#### 5. CONCLUSIÓN

El retardo en los tiempos de re-desarrollo afecta la cantidad de aplicaciones en re-desarrollo lo cual a su vez afecta la efectividad de la cooperación entre generaciones de desarrolladores de software.

La aplicación del mecanismo de cooperación basado en confianza permite el aumento de la confianza entre generaciones de desarrolladores de software, por el aumento de la velocidad de desarrollo de aplicaciones que por su uso y por un factor de tiempo pasaron de aplicaciones en uso a aplicaciones obsoletas.

Incentivar la aplicaciones en uso, incrementa las aplicaciones obsoletas, desde las cuales obtenemos un incremento de las aplicaciones en re desarrollo, este incremento nos afecta de manera

positiva la confianza entre generaciones de desarrolladores.

## 6. AGRADECIMIENTOS

Constantemente agradeciéndole a DIOS por la claridad y sabiduría durante la maestría.  
A mi familia por el apoyo y la comprensión.  
Especial agradecimiento al Dr. Pascual Matera Lajud por su apoyo constante e incondicional.

## 7. REFERENCIAS

- [1] Kollock, P., Social dilemmas: The anatomy of cooperation, *Annual Review of Sociology*, vol. 24, no. 1, 183–214, 1998.
- [2] Ostrom, E., A behavioral approach to the rational choice theory of collective action, in *Polycentric games and institutions: readings from the Workshop in Political Theory and Policy Analysis*. University of Michigan Press, 472, 2000.
- [3] Castillo, D. y Saysel, A., Simulation of common pool resource field experiments: a behavioral model of collective action, *Ecological Economics*, vol. 55, no. 3, 420–436, 2005.
- [4] Sterman, J., *Business dynamics: Systems thinking and modeling for a complex world with CD-ROM*. Irwin/McGraw-Hill, 2000.
- [5] Forrester, J., *Industrial Dynamics*. MIT press Cambridge, MA, 1961.
- [6] Parra, J., Mecanismo de cooperación en dilemas sociales de recurso agotable de gran escala, Ph.D. dissertation, Universidad Nacional de Colombia. Doctorado en Ingeniería Area Sistemas, 2010.
- [7] Parra, J., y Dyner, I., Cooperation mechanism for large-scale social dilemmas involving resource depletion, in *Proceedings International System Dynamics Conference*. System Dynamics Society, 72, 2010.
- [8] Buck, S., *The global commons: an introduction*. Island Press, 1998.
- [9] Ostrom, E., y Walker, J., *Trust and reciprocity: Interdisciplinary lessons from experimental research*. Russell Sage Foundation Publications, 2005.
- [10] Ostrom, E. y Dietz, T. y Dolsak, N. y Stern P. y Stonich, S., *The drama of the commons*. National Research Council, 2002.
- [11] McGinnis, M. y Ostrom, E., Will Lessons from Small-Scale Social Dilemmas Scale Up? New issues and paradigms in research on social dilemmas, 189–211, 2008.
- [12] Salcedo, J. *Dilemas de acción colectiva, instituciones y cooperación en organizaciones tecnológicas. Redes para la innovación*. Barcelona, Cataluña: Universidad autónoma de Barcelona, 2009.
- [13] Parra, J. y Villamizar L., *Efectos del retardo de la información en la gestión de la cooperación en el desarrollo de software libre*, 2013.