

**EVALUACION DE MECANISMOS DE COOPERACION
APLICADOS EN LA PROMOCION DE UNA ACCION
COLECTIVA EFECTIVA ENTRE GENERACIONES DE
DESARROLLADORES EN EL PROCESO DE DESARROLLO DE
SOFTWARE LIBRE
Un enfoque sistémico**

**INVESTIGADOR
FERNANDO CASTILLO ZAPATA
Ingeniero de sistemas
UNIVERSIDAD AUTONOMA DEL CARIBE – COLOMBIA
Estudiante maestría en software libre
Convenio universidad abierta de Cataluña España – Universidad
autónoma de Bucaramanga**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA
PROGRAMA DE INGENIERIA DE SISTEMAS
MAESTRIA EN SOFTWARE LIBRE
BUCARAMANGA-COLOMBIA
2014**

**EVALUACION DE MECANISMOS DE COOPERACION
APLICADOS EN LA PROMOCION DE UNA ACCION
COLECTIVA EFECTIVA ENTRE GENERACIONES DE
DESARROLLADORES EN EL PROCESO DE DESARROLLO DE
SOFTWARE LIBRE
Un enfoque sistémico**

**DIRECTOR DEL PROYECTO
JORGE ANDRICK PARRA VALENCIA,
Ph.D., M.Sc., SE.**

**Profesor titular programa de Ingeniería de sistemas
Investigador asociado Grupo de investigación en pensamiento sistémico.
UNIVERSIDAD AUTONOMA DE BUCARAMANGA-COLOMBIA
Presidente comunidad Colombiana de dinámica de sistemas 2011-2012,2012-
2013**

**INVESTIGADOR
FERNANDO CASTILLO ZAPATA
Ingeniero de sistemas
UNIVERSIDAD AUTONOMA DEL CARIBE – COLOMBIA
Estudiante maestría en software libre
Convenio universidad abierta de Cataluña España – Universidad
autónoma de Bucaramanga**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA
PROGRAMA DE INGENIERIA DE SISTEMAS
MAESTRIA EN SOFTWARE LIBRE
BUCARAMANGA-COLOMBIA
2014**

INDICE GENERAL.	Pág.
1. Introducción.	1
2. Identificación del problema	4
2.1. Título	4
2.2. Grupo de investigación	4
2.3. Línea de investigación	4
2.4. Tipo de investigación	4
2.5. Pregunta de investigación	5
2.6. Criterios de evaluación.	7
2.7. Hipótesis	8
3. Planteamiento del problema	9
3.1. Descripción	9
4. Justificación	11
5. Marco teórico	12
5.1. Conceptos fundamentales	12
5.1.1. Desarrollo de software libre	12
5.1.2. Cooperación	12
5.1.3. Mecanismos de cooperación	13
5.1.4. Comunidades de desarrollo	14
5.1.5. Comportamiento individual	14
5.1.6. Racionalidad colectiva	14
5.2. Mapa conceptual	15
5.3. Matriz (categoría vs libros)	16
5.4. Teorías	18
5.4.1. Teoría de juegos	18
5.4.2. Modelo bazar	18
5.4.3. Dilema del prisionero	19
5.4.4. Teoría de juegos. Estrategia “Tit for tat”	19
5.4.5. Palabras claves utilizadas en la búsqueda	23
5.4.6. Motores de búsqueda	24
5.4.7. Criterios de búsqueda para incluir	24
5.4.8. Criterios de búsqueda para excluir	24
6. Objetivos (generales y específicos).	25
6.1. Objetivo general	25
6.2. Objetivos específicos	25
7. Método de investigación	27
8. Resultados de la investigación	30
9. Conclusiones	54
10. Recomendaciones y trabajos futuros	55
11. Bibliografía	58

1. Introducción.

Este capítulo ofrece los antecedentes sobre software libre y cooperación.

La presente propuesta de investigación se concibe en el contexto del área de desarrollo de software libre. Y surge como un proyecto de investigación explicativa que tiene como finalidad la evaluación de mecanismos de cooperación aplicados en la promoción de una acción colectiva efectiva entre generaciones de desarrolladores en el proceso de desarrollo de software libre.

Los inicios de los procesos de cooperación en el desarrollo de software libre están fuertemente relacionados al modelo bazar

El cual propone tres reglas 1) liberar código fuente rápido y a menudo 2) distribuir responsabilidades y tareas todo lo posible. 3) ser abierto hasta la promiscuidad. Muy a pesar de ser este modelo el líder de este proceso, este presenta problemas para producir cooperación cuando se empiezan proyectos desde cero o cuando se practica en grupos reducidos demasiado heterogéneos o con mucho desnivel de desconocimiento.

Gestionar, promover y convertir a la cooperación en un recurso auto sostenible implica un problema de complejidad dinámica, debido a

que en este tipo de sistemas la relación causa y efecto es distante en el tiempo.

Los dilemas sociales son conflictos que los grupo de individuos enfrentan comúnmente, tomando decisiones entre beneficio individual o colectivo. 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, cuya decisión esta frecuentemente ligada al cuando más que al contribuir o no.

Además, los dilemas sociales contemplan dos aspectos:

- a. Deserción de individuos (no cooperar).
- b. Decisión propia y personal de contribuir y la medida de esa contribución

Estos aspectos generan toda una estructura de conflictos:

- a. Conflictos en el nivel
- b. Conflictos en la localización
- c. Conflictos en la distribución

PRINCIPALES CARACTERISTICAS DE LOS DILEMAS SOCIALES

<p>Conflicto entre la racionalidad individual y el bienestar colectivo cuando se usa un recurso finito y no excluible.</p>
<p>EL dilema ocurre en recursos comunes que pueden ser asumidos como agotables con probabilidad de congestión.</p>
<p>El dilema surge porque los elementos de los cuales un individuo se apodera ya no están disponibles para los demás.</p>
<p>Los individuos conforman un área extensa. Por lo cual no es fácil el contacto cara a cara.</p>
<p>Los individuos tienen intereses diversos.</p>
<p>Las consecuencias y/o efectos de la cooperación de los demás se manifiesta mucho tiempo después.</p>
<p>Los individuos tienen una percepción de los efectos directos que causa el desertar sobre la disponibilidad del recurso agotable.</p>

2. Identificación del problema.

2.1. Título.

Evaluación de mecanismos de cooperación aplicados en la promoción de una acción colectiva efectiva entre generaciones de desarrolladores en el proceso de desarrollo de software libre

2.2. Grupo de investigación.

Pensamiento Sistémico

2.3. Línea de investigación.

Estudios institucionales.

2.4. Tipo de investigación.

Investigación explicativa.

2.5. Pregunta de investigación.

El proceso de cooperación se da y es auto sostenible cuando se permite la comunicación cara a cara y si los integrantes del grupo tiene retroalimentación instantánea del efecto de las acciones del resto del grupo y del estado de dicho recurso. Todo lo anterior NO se presenta en los dilemas sociales de recurso de gran escala.

Las características anteriormente descritas se presentan en los dilemas sociales de pequeña escala, en cuyos dilemas los mecanismos de cooperación basados en la teoría de cooperación han demostrado ser efectivos.

En la evaluación de la efectividad de mecanismos de cooperación en dilemas sociales, se tienen en cuenta los siguientes elementos:

Alto grado de disponibilidad de la información relacionada con el estado del bien y el efecto de las acciones de los demás individuos.
--

Medidas para enfrentar el deterioro de la cooperación debido a la complejidad dinámica producía por:
--

- | |
|--|
| <ul style="list-style-type: none">• Retardos en medir el estado del recurso.• Retardos en reconocer el estado del recurso.• Acciones y efectos de las acciones de los individuos sobre |
|--|

la disponibilidad del recurso.

- Retardos inherentes a la dinámica del recurso.
- Retardos inherentes a la apropiación del recurso.
- Retardos inherentes a la percepción (tiempo requerido por el grupo para asimilar cambios en las variables monitoreadas.)

Alto grado de heterogeneidad y dispersión geográfica de los individuos.

Condiciones iniciales no favorables de confianza.

Los siguientes mecanismos permiten abordar los anteriores elementos:

Mecanismo de cooperación por confianza.

Mecanismo de cooperación por percepción del daño: permite enfrentar las condiciones iniciales desfavorables.

Mecanismo de cooperación como norma: permite enfrentar la volatilidad de la cooperación basada en confianza y hacer sostenible dicha cooperación.

Consideración a tener en cuenta por parte de los mecanismos:

Reemplazar la comunicación cara a cara con la difusión efectiva del estado de los recursos y el efecto de las acciones de los

demás.
Considerar los efectos sobre la calidad e la información debido a los retardos, la realimentación imperfecta y la inercia.
No disponibilidad del recurso por escaladas de desconfianza y condiciones iniciales insuficientes de confianza.
Inducir la cooperación como norma a largo plazo, luego de acentuar la cooperación por confianza y la cooperación por percepción del daño.

La pregunta de investigación propuesta es la siguiente:

¿Pueden y como pueden reglas y normas basadas en la teoría de la cooperación promover la cooperación inter generacional entre desarrolladores?

2.6. Criterios de evaluación

La efectividad de los mecanismos de cooperación será medida según la capacidad de promover y sostener la cooperación.

¿Permiten solventar condiciones iniciales desfavorables?

¿Generan cooperación auto sostenible?

¿Generan hábito de norma para garantizar que no sea una norma pasajera?

¿Generan contingencias ante escaladas de deserción?

2.7. Hipótesis

Teniendo en cuenta la argumentación presentada se propone la siguiente hipótesis:

Es posible aplicar un mecanismo de cooperación que permita mejorar la cooperación inter generacional entre desarrolladores en el proceso de desarrollo de software libre.

3. Planteamiento del problema.

3.1. Descripción.

La diversificación y amplitud de los avances en comunicaciones han permitido que más personas trabajen a distancia y pertenezcan a grupos de desarrollo de software libre a nivel mundial. Este incremento exponencial y las amplias demandas del mercado que ven en el software libre, nuevos y valiosos modelos de negocio, promueven a la búsqueda de métodos y estrategias que garanticen la implementación de buenas prácticas de cooperación en el desarrollo cooperativo.

Los desarrollos de software libre que empiezan desde cero con integrantes heterogéneos y con diversos niveles de conocimiento presentan el mayor índice de falta de cooperación. Esto último muchas veces ligado a falta de mecanismos y estrategias de comunicación efectiva para la promoción de la cooperación.

Aproximadamente el 50% no recibe retribución material directa, ver estudio de David, Waterman y Arora, 2003

El desarrollo de software libre hace uso de mecanismos cibernéticos para la comunicación y el intercambio de ideas, pero es la calidad en

la cooperación la que merece una nueva propuesta basada en métodos y estrategias para el mejoramiento de la cooperación..

Es necesario:

Estudiar la cooperación CORE-DEVELOPERS y nuevas generaciones.

Analizar Estrategias de resolución de conflictos.

Evaluar mecanismos de cooperación que aplicado en la promoción de una acción colectiva efectiva entre generaciones de desarrolladores de software.

4. Justificación

El desarrollo de software libre promueve el avance tecnológico con base en libertades, que invitan a personas, de todos los niveles, a crear nuevas y suficientes soluciones tecnológicas para el mundo de hoy.

Los nuevos modelos de negocio que propone el DESARROLLO DE SOFTWARE LIBRE están marcando pautas a nivel mundial tanto para el sector propietario como para el sector libre. Es inevitable el crecimiento de este movimiento, pero también es inevitable la necesidad del mejoramiento de la cooperación, cuyo mejoramiento debe ser adaptable y acorde a los perfiles de las generaciones que integran y marcan pautas en el mundo del desarrollo del software libre.

Teniendo en cuenta el recurso detallado anteriormente, y asimilando estos términos a los dilemas sociales de gran escala se requieren evaluar mecanismos de cooperación en búsqueda de una acción colectiva efectiva entre generaciones de desarrolladores de software libre, lo cual contribuye al desarrollo de más cantidad de software con mayor calidad.

5. Marco teórico.

A continuación se detallan los conceptos y teorías fundamentales relacionadas con el estudio del tema en cuestión.

5.1. Conceptos fundamentales.

5.1.1. Desarrollo de software libre

Actividad liderada por la comunidad de software libre, la cual está encaminada a producir el código fuente que a través de integración y cooperación se convierte en software para usar.

5.1.2. Cooperación

Trabajo en común llevado a cabo por un grupo de trabajo, de personas o entidades mayores hacia un objetivo compartido, generalmente usando métodos también comunes, en lugar de trabajar de forma separada en competencia.

5.1.3. Mecanismos de cooperación

Cooperación mediante castigo

Para Schelling la amenaza del castigo implícito o explícito puede persuadir a los otros para cumplir acuerdos.

Cabe la posibilidad, según la teoría, de que surja o se sostenga la cooperación entre enemigos o amigos con un bajo nivel de confianza, gracias al efecto de persuasión que los individuos le asignan a su percepción sobre la posibilidad y capacidad de retaliación de los otros individuos. Para que la retaliación sea creíble y logre el efecto de persuasión, debe existir una percepción de alta capacidad y voluntad para APLICAR EL CASTIGO.

Cooperación Basada en confianza.

A mayor disponibilidad del recurso, mayor confianza de cooperación lo que a su vez permitirá el incremento de la acción cooperativa, lo que a su vez mejorará la disponibilidad del recurso.

Cooperación y aprendizaje social

Luego de repetidos encuentros de cooperación, los individuos que adopten la cooperación como norma no requieren confianza de

cooperación para cooperar. A mayores experiencias pasadas de cooperación, mayor aprendizaje de cooperación como norma, lo que a su vez permite la acción cooperativa que mejora la disponibilidad del recurso.

5.1.4. Comunidades de desarrollo

Colectivo sin fines de lucro, que promueve el uso, estudio, investigación y desarrollo de Tecnologías Libres para las áreas de educación, seguridad, industria, comercio, etc... Con el fin de que todos los usuarios puedan participar para compartir sus conocimientos y aprender de otros.

5.1.5. Comportamiento individual

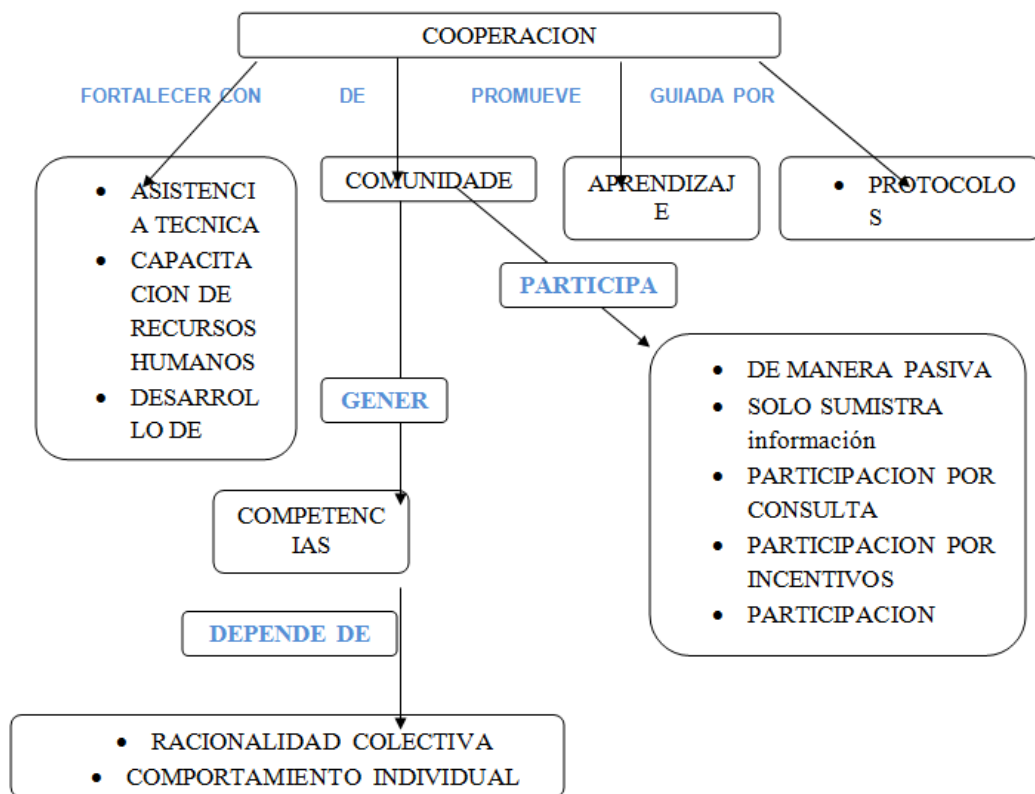
Proceder individual relacionado a su entorno y a sus estímulos.

5.1.6. Racionalidad colectiva

Capacidad colectiva que permite pensar, evaluar, entender y actuar de acuerdo a ciertos principios de consistencia. Para satisfacer algún objetivo o finalidad. Cualquier construcción

mental llevada a cabo mediante procedimientos racionales tiene por tanto una estructura lógico-mecánica distinguible.

5.2. Mapa conceptual



5.3. Matriz (categorías vs libros)

	COOPERACION PARA EL DESARROLLO	APRENDIZAJE COLABORATIVO	COMPETENCIAS TRANSVERSALES	RACIONALIDAD COLECTIVA / COMPORTAMIENTO INDIVIDUAL	COOPERACION INTERGENERACIONAL EN DESARROLLO SOFTWARE LIBRE
80 HERRAMIENTAS PARA EL DESARROLLO PARTICIPATIVO	*				
LAS COMUNIDADES VIRTUALES COMO POTENCIAL PEDAGÓGICO PARA EL APRENDIZAJE COLABORATIVO A TRAVÉS DE LAS TIC		*	*		
“ GUIDELINES FOR PERFORMING SYSTEMATIC LITERATURE REVIEWS IN SOFTWARE ENGINEERING				*	
COOPERACIÓN PARA EL MEJORAMIENTO	*			*	

TO DE LA CALIDAD DEL AIRE					
EVALUACIÓN DE LA COOPERACIÓN EN DILEMAS SOCIALES DE GRAN ESCALA	*			*	

5.4. Teorías

5.4.1. Teoría de juegos.

Área de la matemática aplicada que utiliza modelos para estudiar interacciones en estructuras formalizadas de incentivos y llevar a cabo procesos de decisión.

Esta teoría nos define una situación en la cual existe una interacción de varios individuos con participación consciente buscando el máximo provecho, con lo cual se incurre en un costo.

Mediante turnos se participa de una manera concurrente sin saber de antemano la opción elegida por el rival.

5.4.2. Modelo bazar.

Modelo de desarrollo de software que busca mayor calidad, mayor eficacia.

Este modelo implica una autoridad sin mando pero con autoridad conferida.

Características:

- Sin carácter competitivo.
- Desarrollo compartido.

- Defensa de lo público.
- Información descentralizada.
- Inteligencia compartida.
- Creación colectiva.
- No es necesario un control exhaustivo sobre el desarrollo.

5.4.3. Dilema del prisionero

El dilema del prisionero es un problema fundamental de la teoría de juegos que muestra que dos personas pueden no cooperar incluso si en ello va el interés de ambas.

Para el caso del software libre es una partida de juego de todos contra todos y la estrategia adoptada es la colaboración.

5.4.4. Teoría de juegos. Estrategia: "tit for tat"

Es la estrategia ganadora en la Teoría de juegos. “Solo se coopera si el adversario coopera”. En la primera jugada el primer jugador realiza una acción de cooperar y después una vez vuelto el turno este mismo jugador copia los movimientos del contrincante. Algunos opinan que esta estrategia ganadora, TIT FOR TAT,

tienen unos rasgos que la hacen más sofisticadas que otras con aptitud menos cooperantes. Estos rasgos son:

- Amable. Son aquellas estrategias que nunca es la primera en ser egoísta. Ser amable no implica que perdone.

- Clemente. Se venga inmediatamente del traidor o egoísta pero que al final tiende a olvidar los antiguos perjuicios sufridos.

Tit for Tat no desea más premio que los demás, además se siente bien cuando el otro obtiene el mismo premio. En Tit for Tat nunca gana el juego como mucho empata. Esto en SWL significa que todo el mundo debe disponer de las mismas capacidades para ser libre que las de uno mismo. Al final siempre acaban ganando las tácticas amables y clementes frente a las sucias aunque en la naturaleza tiende a predominar las últimas (cooperación egoísta).

En la teoría de juegos los usuarios que utilizan la técnica del Tit for Tat, van cooperando en pocos equipos locales incluso tienen la fuerza de poder prosperar hacia entornos muchos más amplios, incluso pueden acaparar a cierto número de individuos no cooperantes que van totalmente a lo suyo, toman una actitud egoísta.

La cooperación causa una realimentación en la que obliga a que todo el mundo que se beneficie del SWL tenga la obligación de ir divulgando su uso. Esto hace que el modelo perdure y sea potente, además hace que haya gran cantidad de usuarios colaborando en el desarrollo. El SWL no sigue la técnica del Tit for Tat completamente ya que lo usan gente que no coopera. A parte, hay un tipo de derecho, el Copyleft que permite que la gente se beneficie del SWL pero no puede apropiarse ni crear aplicaciones propietarias, esto hace que se aproxime a nuestra estrategia.

En ocasiones llevar una estrategia amable y clemente puede apoderarse de estrategias que no cooperen no siendo necesario expulsarlas, a diferencia del Tit for Tat. Aunque algunos casos suponen un peligro debido a que grandes empresas adopten claramente acciones egoístas y no cooperantes infringiendo por ejemplo licencias GPL (General Public License).

La comunidad de desarrollo de software libre corresponde a las formas básicas y humanas en cuanto a la participación y colaboración se refiere. Son las experiencias individuales y los grados de compromiso los que demarcan el contexto para el logro de objetivos.

En el campo tecnológico la tendencia a ser autodidactas marca un avance en términos de aprendizaje-sostenido, pero demarca unas fronteras ideológicas las cuales tienen impacto negativo en la coordinación de tareas encaminadas a respaldar la evolución en el aprendizaje de los demás.

La colaboración y la participación activa, fluida y proactiva están ligadas a la flexibilidad y disponibilidad de los grupos integrantes de la comunidad del software libre. Estos aspectos son evolutivos y no fijos en el tiempo.

Los entornos de desarrollo de software se caracterizan por el pensamiento analítico, sistemático y racional en todas las etapas del ciclo de desarrollo, con lo cual se obtienen los objetivos previstos. Lo anterior sumado al concepto de iterativo-evolutivo de las metodologías ágiles de desarrollo de software completan los requerimientos para la realización de proyectos exitoso con un alto grado de participación.

Se hace necesario perfeccionar y acelerar la evolución de los modelos de cooperación que pueden ser aplicados para los desarrollo de software libre.

Los primeros paso deben estar encaminados a la confrontación de las teorías aplicadas actualmente contra herramientas que permitan concretizar un enfoque participativo. Esto puede

lograrse dando respuesta a los siguientes interrogantes. ¿Las herramientas actuales tienen un efecto grupal medible?, ¿Permiten la aplicación a grupos interdisciplinarios?, ¿Se obtiene un alto grado de conocimiento recíproco?

Todo proceso de desarrollo tiene agentes participativos en diferentes niveles y tienen líderes, estos últimos deben agregar al proceso un alto grado de espíritu facilitador cuyo perfil debe abarcar características como:

Depositar confianza en la gente para potencializar capacidades.

Capacidad de escuchar.

Balance entre las ideas propias y las ideas colectivas.

Adaptabilidad de la metodología de trabajo según los contextos.

Creatividad, flexibilidad y sensibilidad en el trato con los integrantes.

Sintetizar y analizar para poder terminar situaciones adversas entre participantes.

5.4.5. Palabras claves utilizadas en la búsqueda

La búsqueda se orientó a los trabajos relacionados con desarrollo de software libre, desarrollo colaborativo y aprendizaje colaborativo.

5.4.6. Motores de búsqueda

<http://scholar.google.com/>

<http://www.researchgate.net>

5.4.7. Criterios de búsqueda para incluir

- Desarrollo de software colaborativo
- Desarrollo de proyectos participativos
- Proyectos de cooperación
- Desarrollo software libre
- Redes sociales

5.4.8. Criterios de búsqueda para excluir

- Filosofía
- Gerencia
- Redes de computadores
- Desarrollo software privativo

6. Objetivos.

6.1. Objetivo general

Evaluar mecanismos de cooperación aplicados en la promoción de una acción colectiva efectiva entre generaciones de desarrolladores en el proceso de desarrollo de software libre

6.2. Objetivos específicos

- Proponer un sistema de reglas y estrategias orientadas a promover la cooperación intergeneracional con base en la revisión de la literatura realizada.
- Diseñar un modelo de simulación en Dinámica de Sistemas que represente el proceso de desarrollo de software libre y cómo las reglas y estrategias propuestas afectan la cooperación en dichos procesos de desarrollo.

- Evaluar la efectividad del sistema de reglas y estrategias propuesto para la promoción efectiva de la cooperación en el proceso de desarrollo de software libre.

7. Método de investigación

La dinámica de sistema nos da un enfoque para el análisis de políticas y el diseño asistido por computador que aplica a problemas dinámicos en sistemas sociales, administrativos, económicos, ecológicos complejos caracterizados por su interdependencia, la interacción mutua, la realimentación de información y la causalidad circular.

Los modelos en dinámica de sistemas combinan niveles y decisiones articulados para explicar un sistema. Normalmente, los modelos en dinámica de sistemas buscan responder a una pregunta o propósito específico.

Los pasos fundamentales del enfoque de Dinámica de Sistemas son:

- Articular el problema.
- Desarrollar una hipótesis dinámica que explique la causa del problema.
- Construir un modelo de simulación por computador.
- Evaluar si el modelo reproduce el comportamiento observado en el mundo real.
- Diseñar y evaluar políticas para aliviar el problema.

- Implementar la solución.

Una parte importante de la metodología de investigación de este proyecto es identificar los criterios de evaluación de la efectividad a tener en cuenta. Los cuales son:

- Cooperación
- Aprendizaje
- Confianza
- Complejidad dinámica

La evaluación de la efectividad de la cooperación en la promoción de una acción efectiva entre generaciones de desarrolladores en el proceso de desarrollo de software libre requiere realizar una estimación para identificar si:

1. la cooperación alcanzada es sostenible.
2. El mecanismo es capaz de promover la cooperación a pesar de condiciones de confianza inicial de cooperación no favorables, si es suficiente y oportuna la realimentación del estado del recurso.

En el proceso de desarrollo de software libre se pueden distinguir el siguiente mecanismo de cooperación: **Cooperación por confianza.**

Si la confianza tiene una suficiente calidad se generan oportunidades de confianza.

A mayor disponibilidad del recurso → mayor confianza de cooperación.

A mayor confianza de cooperación → mayor acción cooperativa.

A mayor acción cooperativa → mayor disponibilidad del recurso.

Evaluación de la efectividad de los mecanismos de cooperación.

Mecanismo cooperación basado en confianza: ciclo de refuerzo: cooperación.

A mayor confianza de cooperación mayor cantidad de software en re-desarrollo, labor que se facilita por el grado de confianza relativa y el tránsito entre desarrolladores de aplicaciones nuevas y desarrolladores de re-desarrollos.

8. Resultados de la investigación

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

8.1. Hipótesis Dinámica

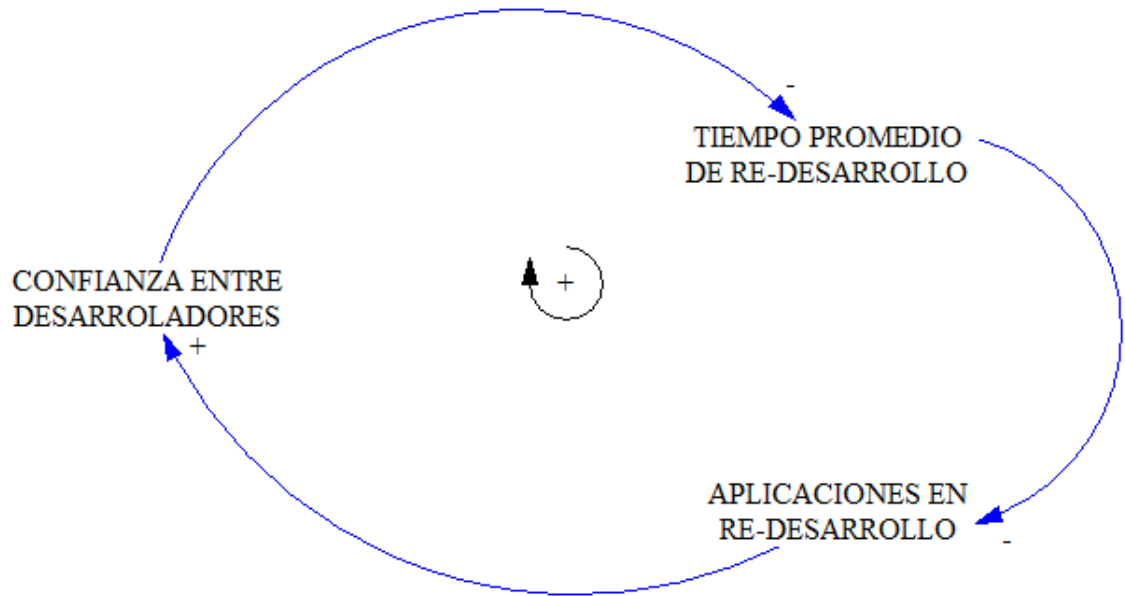
Se planteo una hipótesis dinámica que expone que a mayor confianza entre generaciones de desarrolladores de software menor tiempo promedio de re-desarrollo y así mismo mayores aplicaciones en re-desarrollo.

En este artículo se realizo 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.

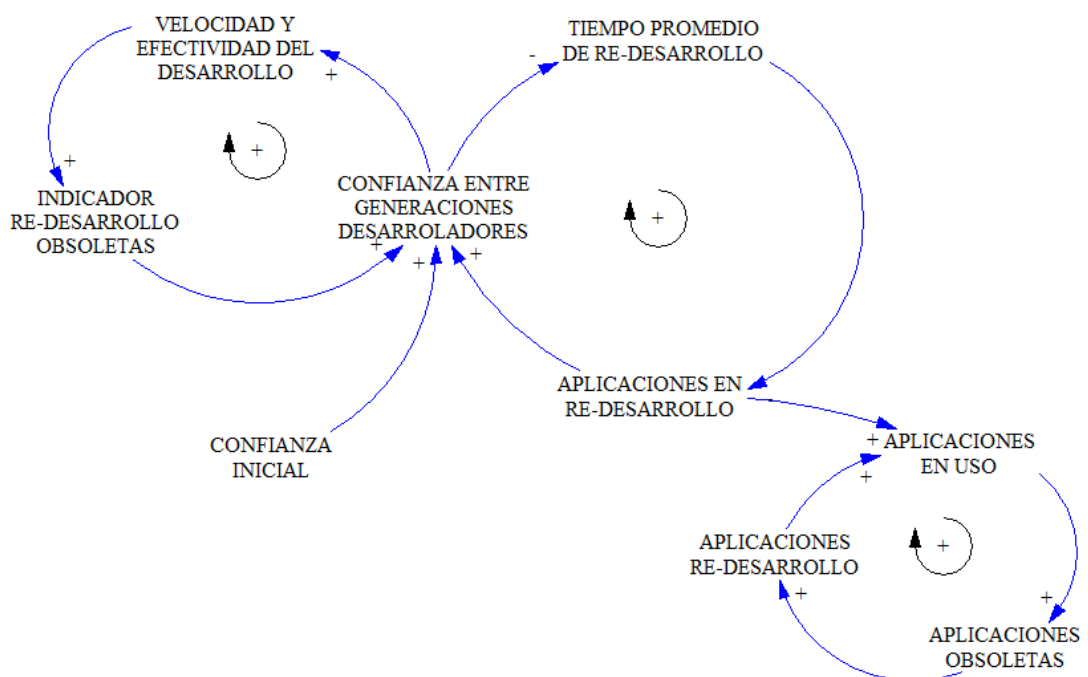
Si la confianza es de la suficiente calidad, se genera una oportunidad de cooperación.

Hipótesis dinámica básica.



Hipótesis dinámica final.

Integrando a la hipótesis dinámica inicial el ciclo de las aplicaciones y la velocidad y efectividad del desarrollo de software, obtenemos esta nueva versión.



8.2. Modelo De Simulación

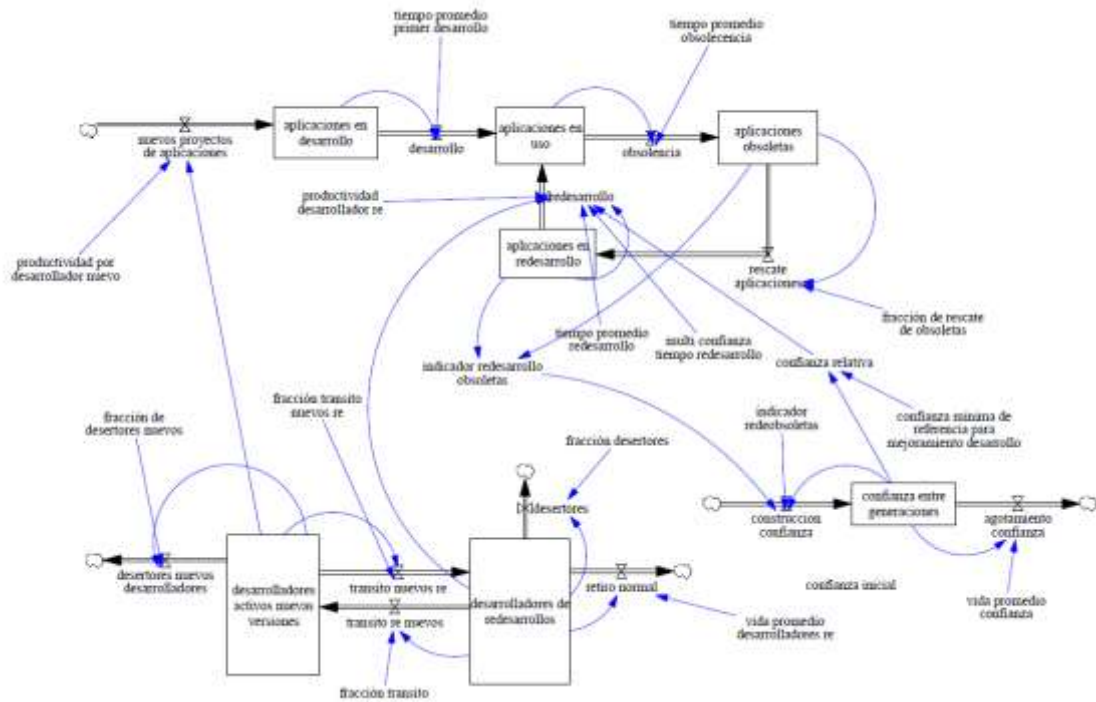
El modelo de simulación presenta como la confianza entre generaciones de desarrolladores de software se representa como un nivel que se alimenta de la percepción retardada de las aplicaciones en re-desarrollo.

La confianza degrada mediante una vida media.

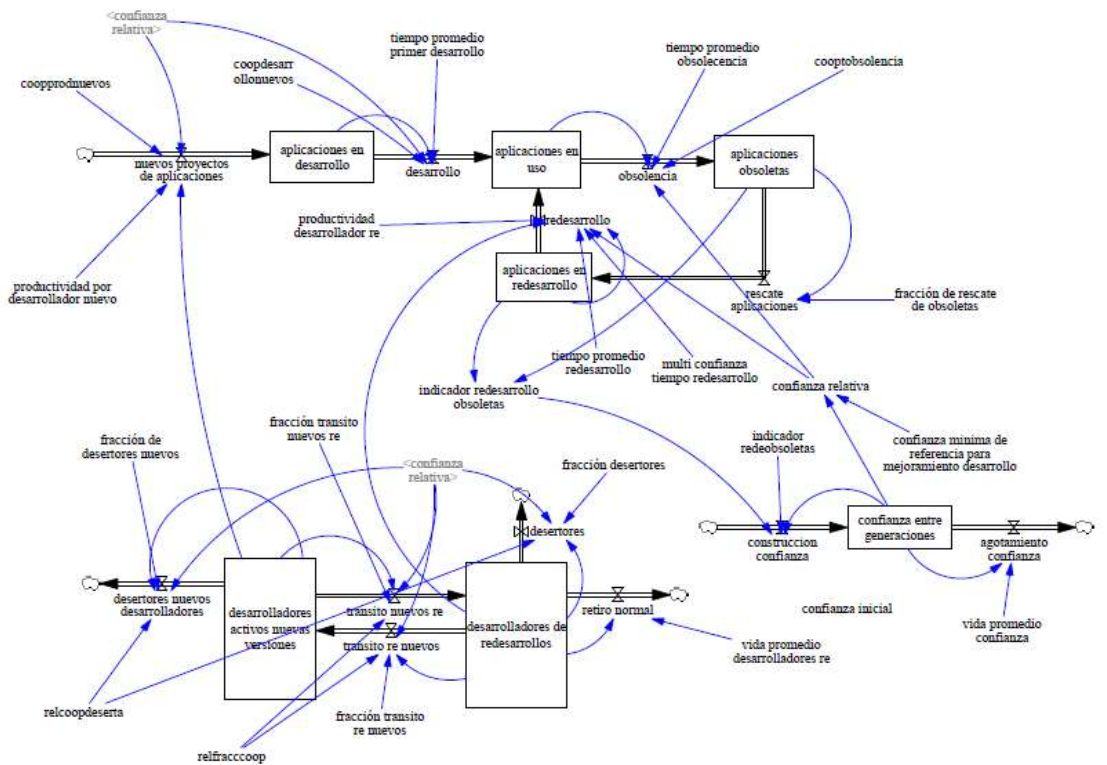
La confianza afecta el re-desarrollo que a su vez afecta las aplicaciones en uso. El ciclo se define como del tipo refuerzo.

Finalmente se ilustra como las cantidades de aplicaciones obsoletas se aumentan a medida que bajan las aplicaciones en uso.

La cantidad de desarrolladores nuevos que se convierten a desarrolladores de re-desarrollos combinado con los desarrolladores de re-desarrollo, 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.



Versión 1.



Versión 2: confianza aplicada los nuevos proyectos de aplicaciones, al desarrollo, a la obsolescencia, a los desarrolladores activos de nuevas versiones, y a los desarrolladores de re-desarrollo.

8.3. 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 transito a desarrollar re-desarrollo y la productividad de cada generación de desarrolladores, aumentan las aplicaciones en uso.

Un menor tiempo de obsolescencia 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 que aumenta la confianza entre generaciones de desarrolladores de software.

Aumentando el tiempo promedio de desarrollo obtenemos un aumento en las aplicaciones en desarrollo y a su vez una disminución de las aplicaciones en uso y de las aplicaciones obsoletas, lo cual NO incentiva el ciclo de re-desarrollo que afecta la confianza entre generaciones de desarrolladores de software.

Aumentado la vida promedio de la confianza obtenemos un aumento de la confianza entre generaciones de desarrolladores de software, por la disminución del agotamiento de la confianza.

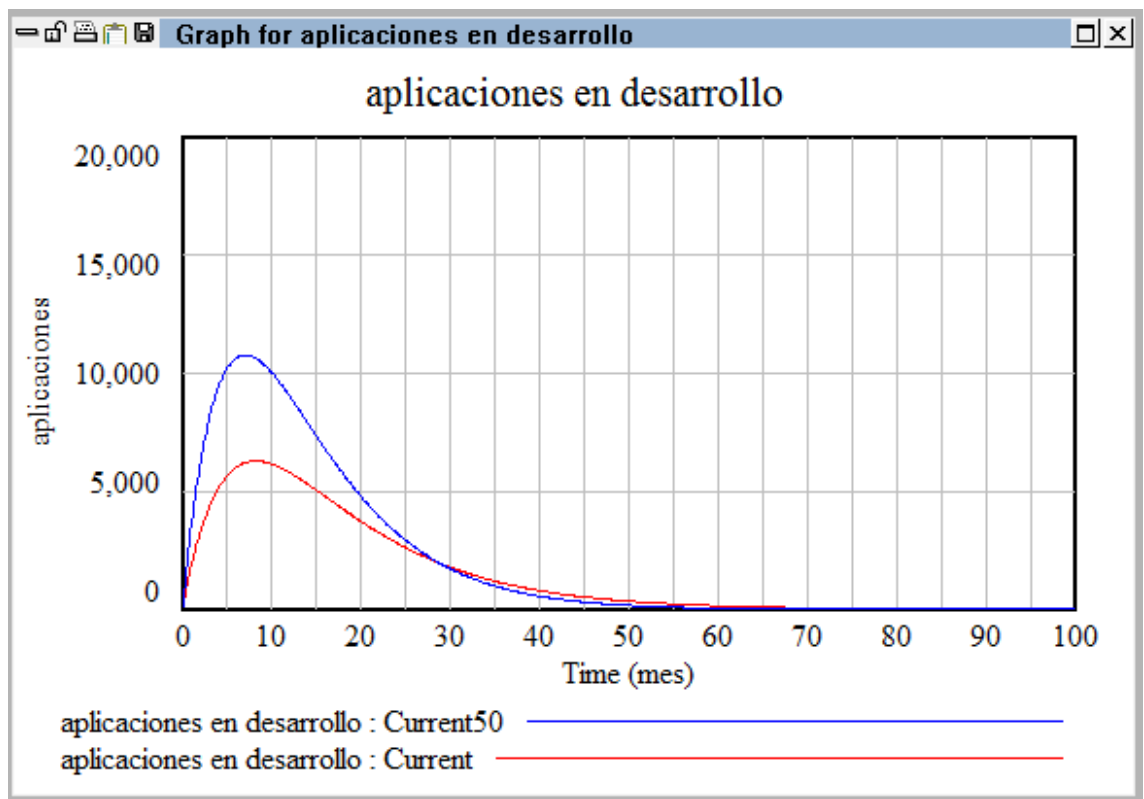
Aumentando la confianza relativa se disminuye la velocidad de re-desarrollo y se disminuye un poco las aplicaciones en uso.

Aumentando la fracción de aplicaciones obsoletas, aumentan las aplicaciones en re-desarrollo, disminuyen las aplicaciones obsoletas, aumentando el indicador de re-desarrollo de obsoletas lo cual afecta positivamente la confianza entre generaciones de desarrolladores de software.

Aumentando el tiempo de re-desarrollo se disminuyen las aplicaciones en re-desarrollo lo cual afecta negativamente la confianza entre generaciones de desarrolladores de software.

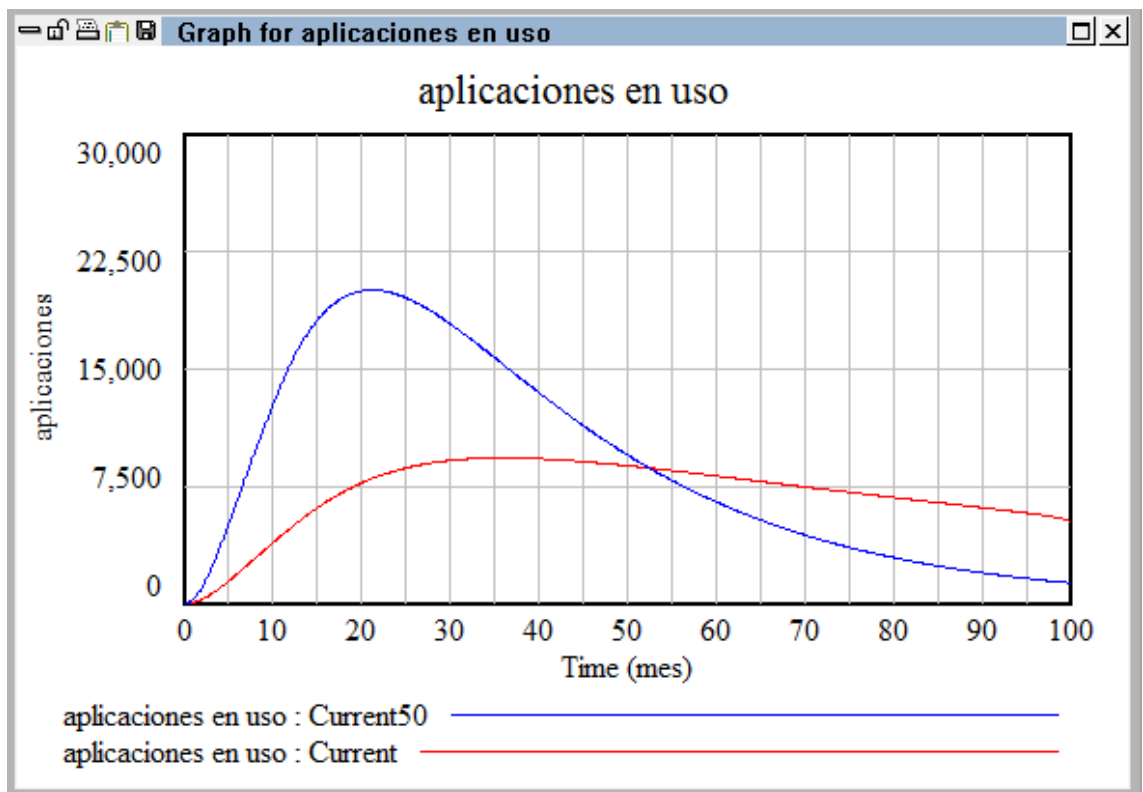
Graficas:

Para valores de confianza inicial: valores (1,50)



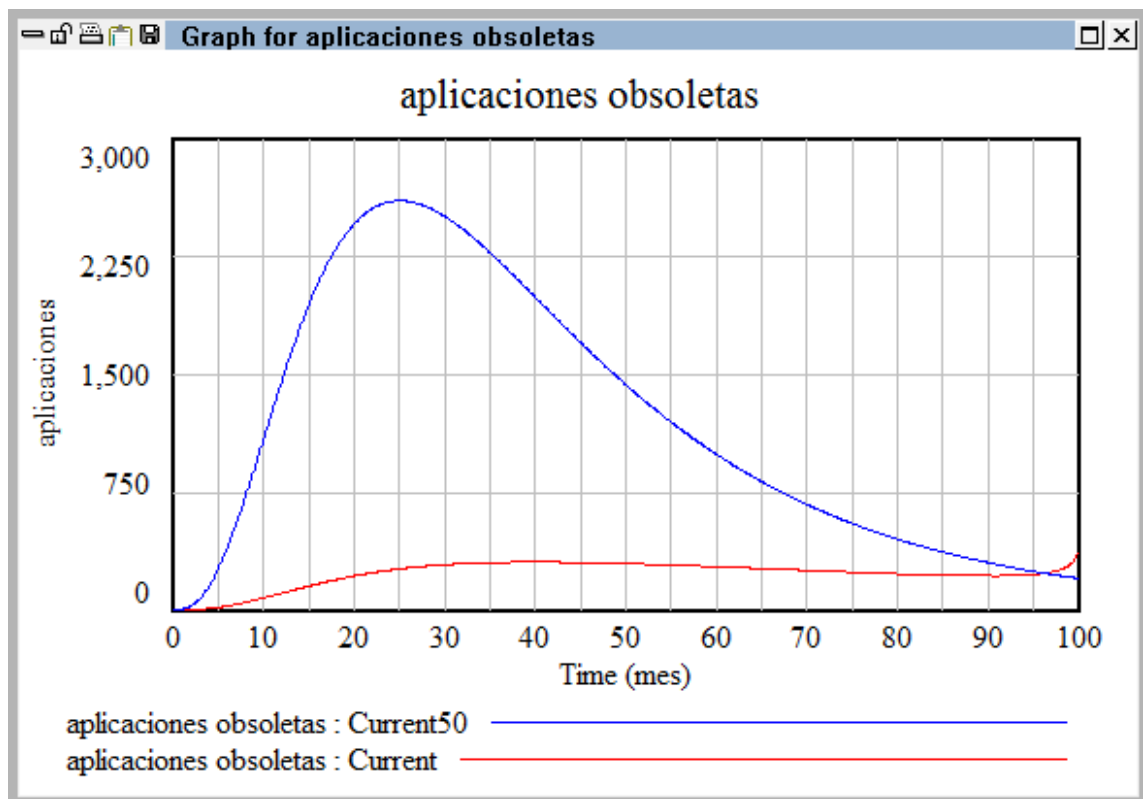
Conclusión: a mayor confianza inicial mayor cantidad de aplicaciones en desarrollo

Para valores de confianza inicial: valores (1,50)



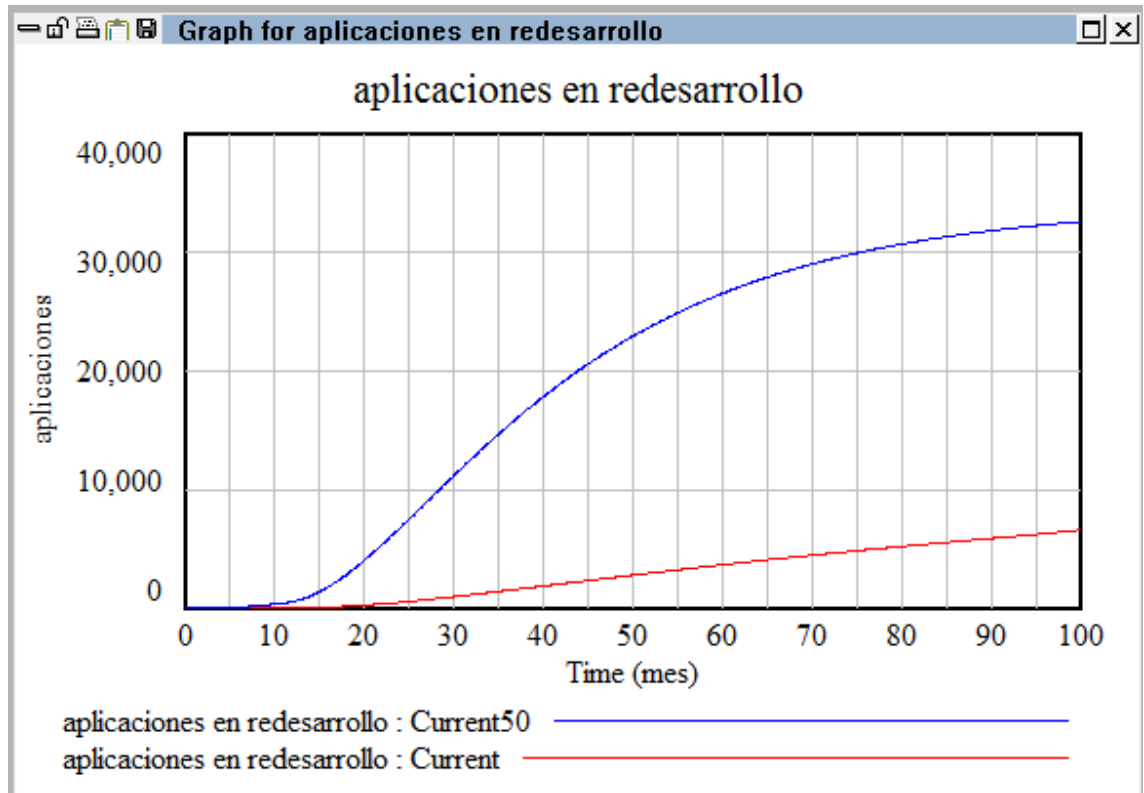
A mayor confianza inicial mayor cantidad de aplicaciones en uso

Para valores de confianza inicial: valores (1,50)



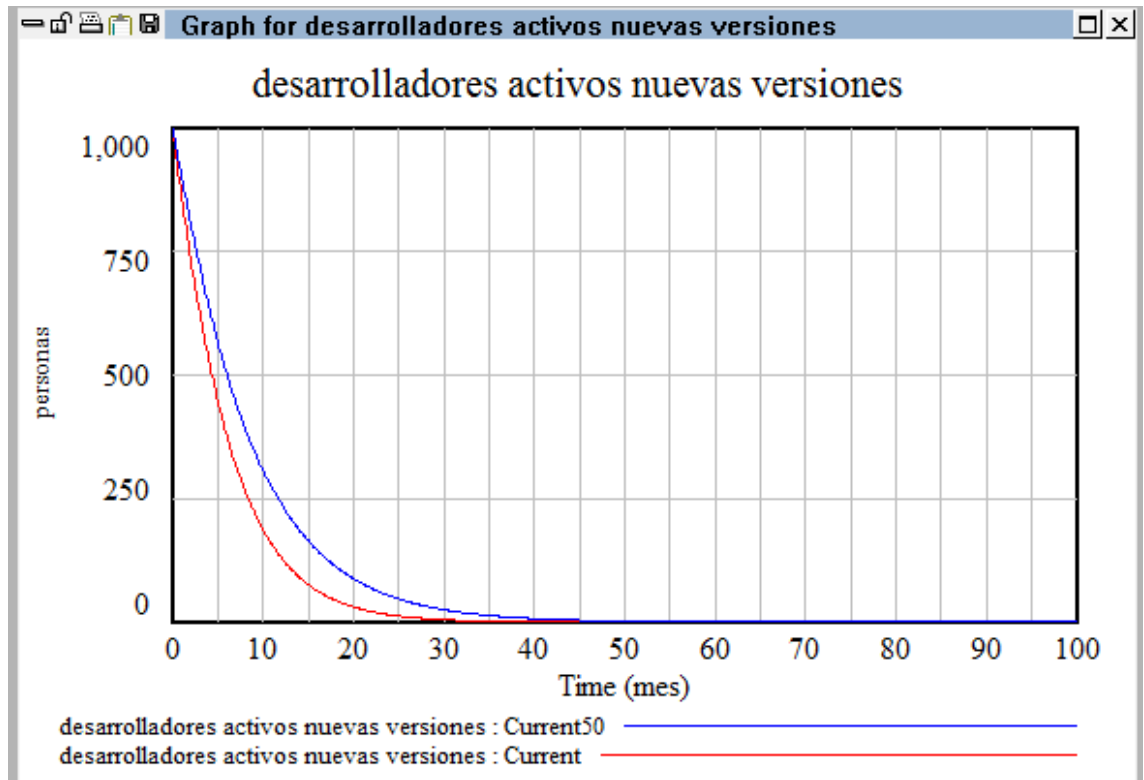
A mayor confianza inicial mayor cantidad de aplicaciones obsoletas.

Para valores de confianza inicial: valores (1,50)



A mayor confianza inicial mayor cantidad de aplicaciones en re-desarrollo

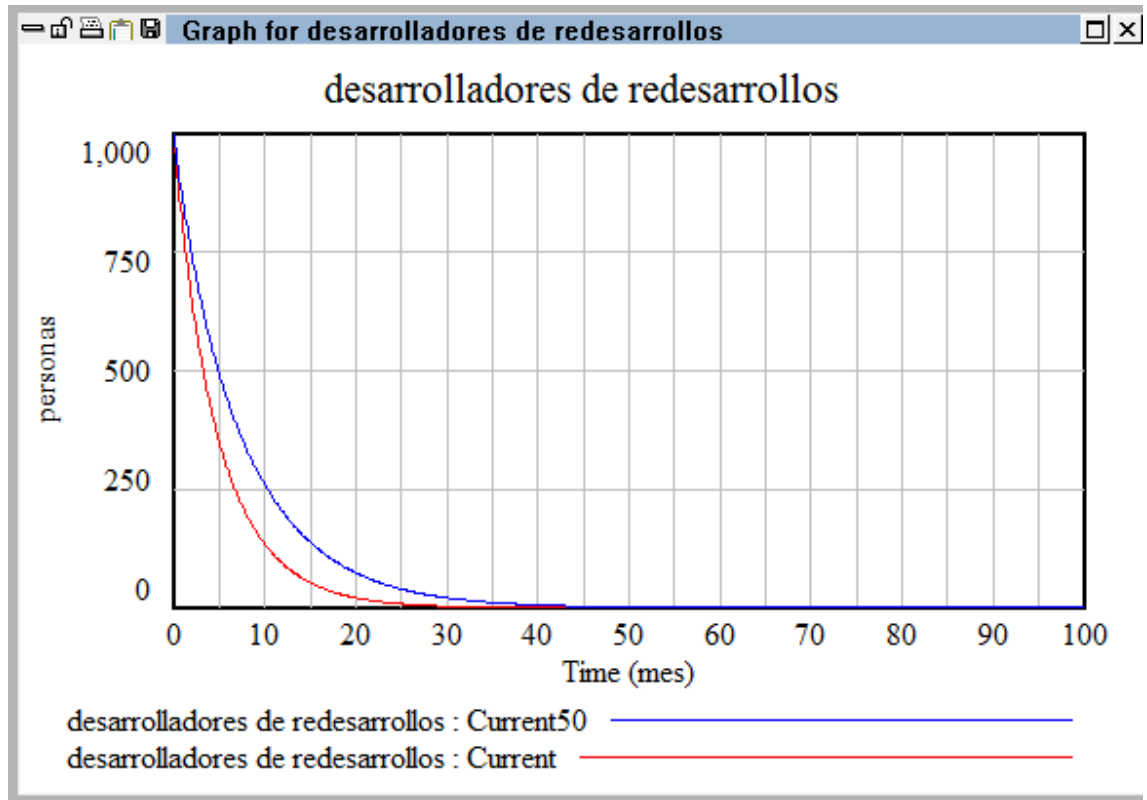
Para valores de confianza inicial: valores (1,50)



Conclusión: para valores altos de la vida promedio de la confianza hay leve incremento de los desarrolladores activos de nuevas versiones.

Propuesta para trabajos fututos.

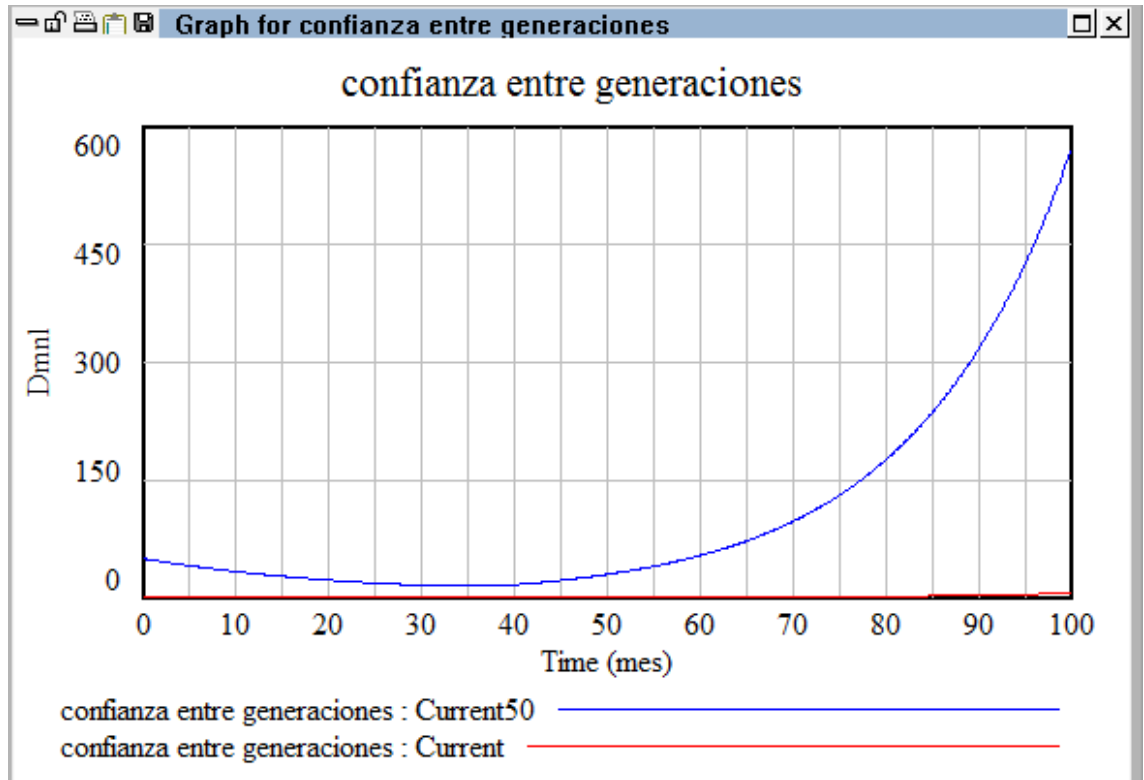
Para valores de confianza inicial: valores (1,50)



Conclusión: para valores altos de la vida promedio de la confianza hay leve incremento de los desarrolladores activos de nuevas versiones.

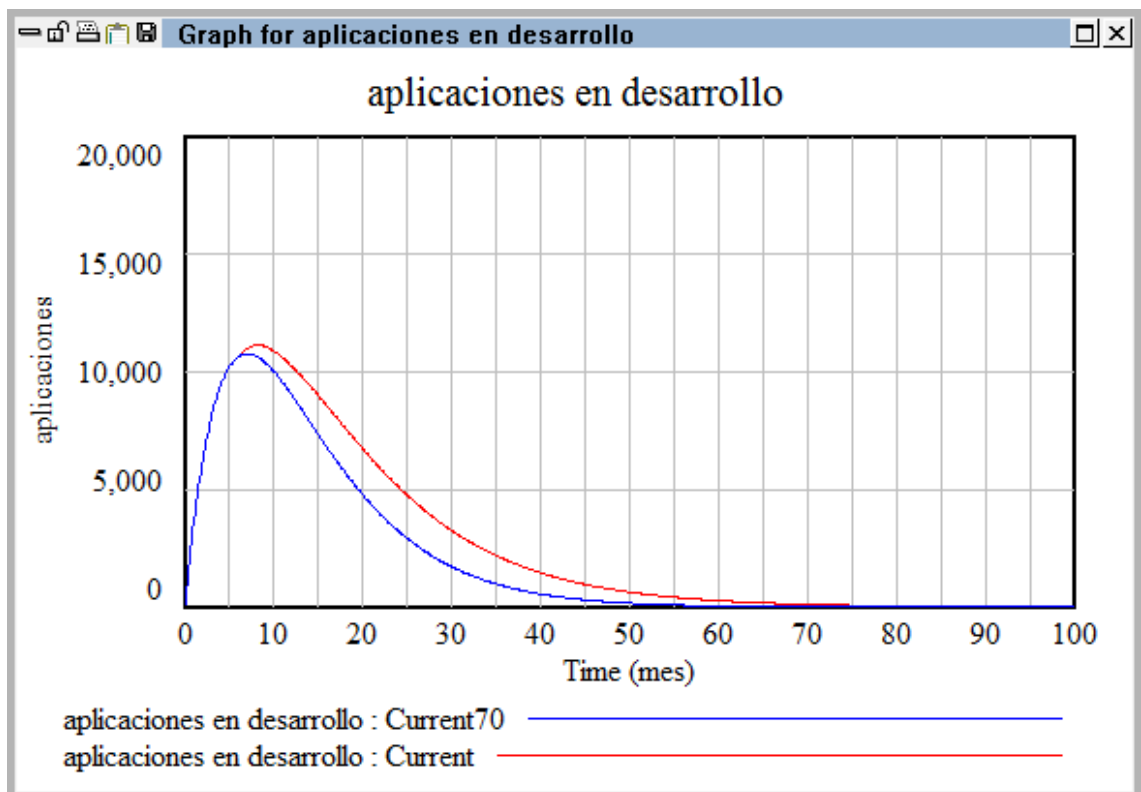
Propuesta para trabajos fututos.

Para valores de confianza inicial: valores (1,50)



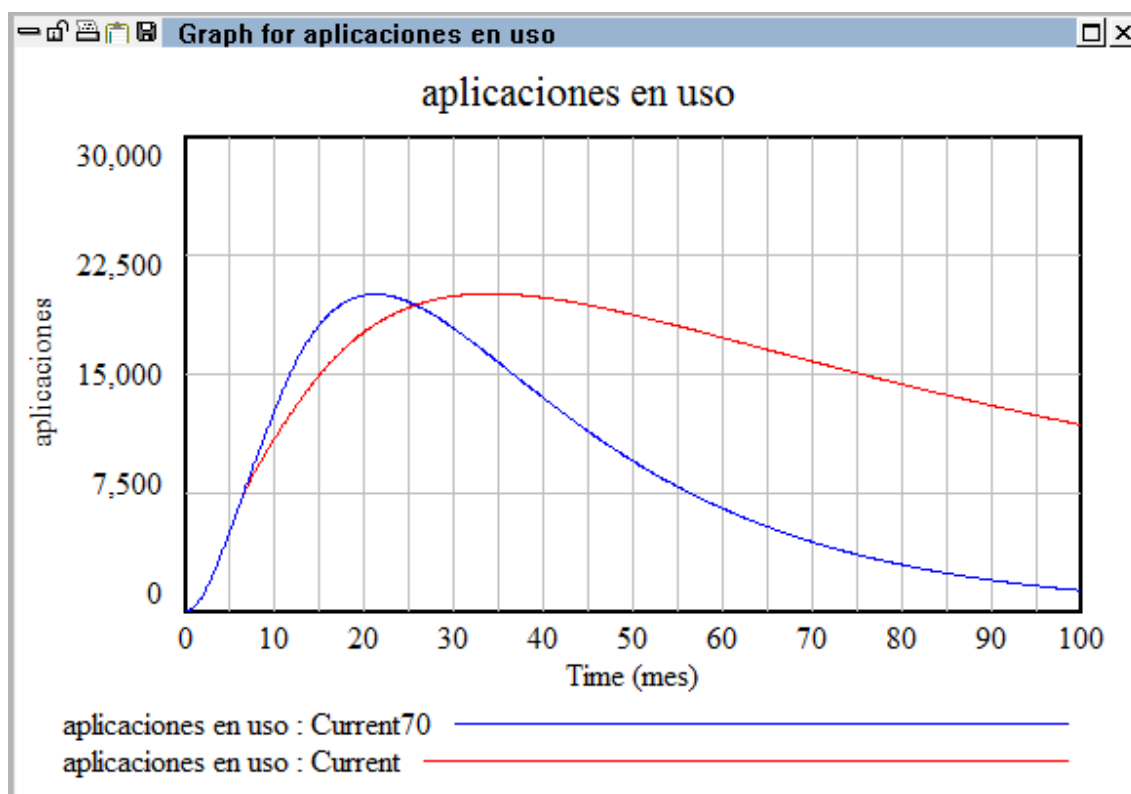
Conclusión: a mayor confianza inicial existe un incremento significativo de confianza entre generaciones.

Para Vida promedio de la confianza: valores (5,70)



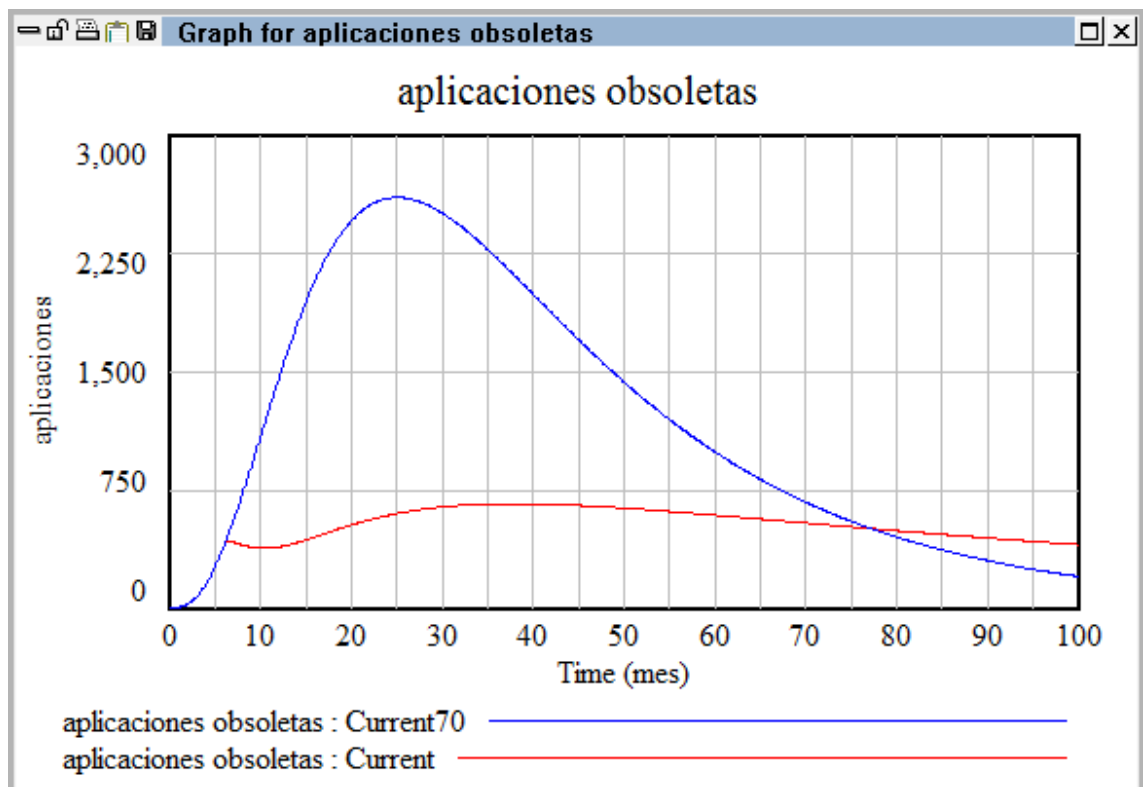
Conclusión: para valores altos de la vida promedio de la confianza menor cantidad de aplicaciones en desarrollos iniciales.

Para Vida promedio de la confianza: valores (5,70)



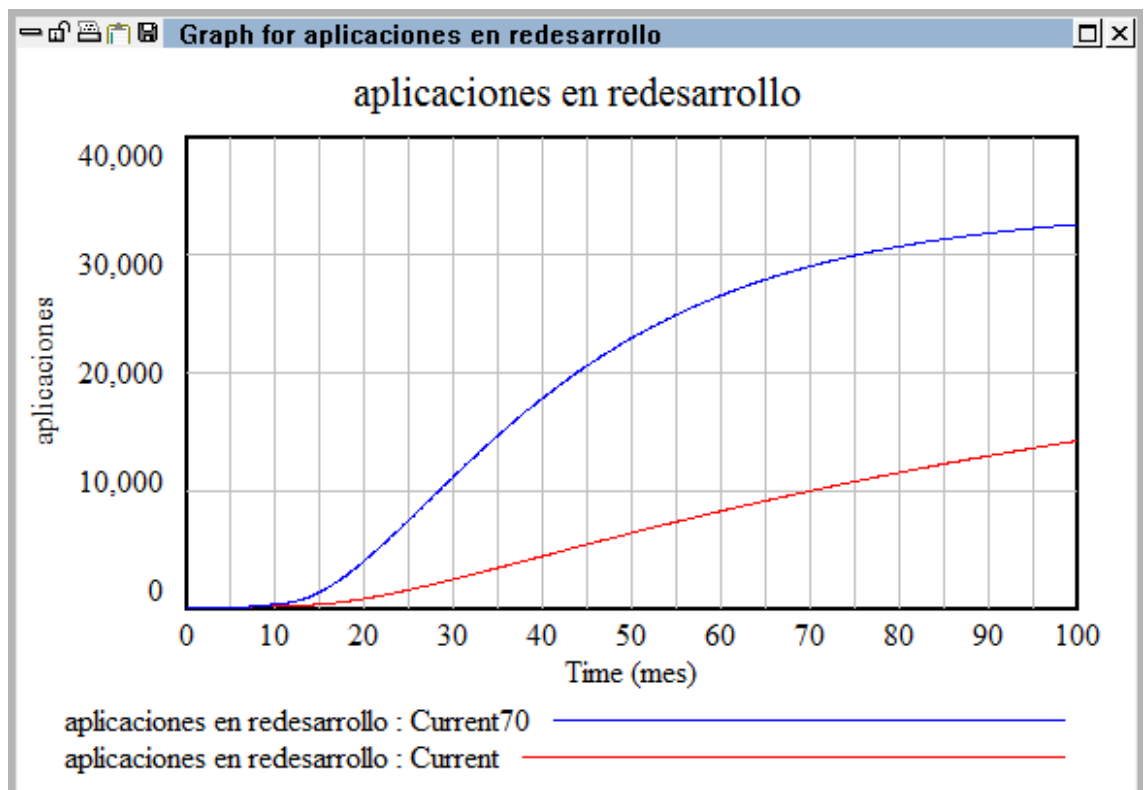
Conclusión: para valores altos de la vida promedio de la confianza hay un incremento de las aplicaciones en uso.

Para Vida promedio de la confianza: valores (5,70)



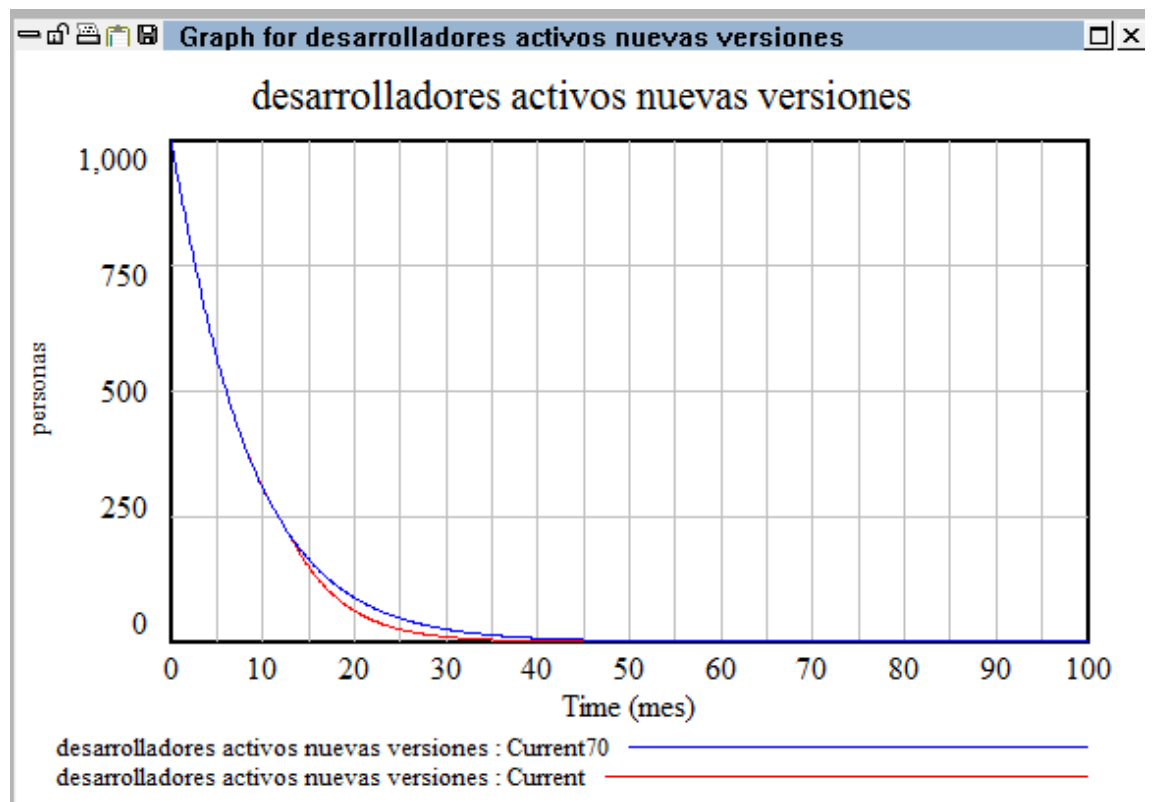
Conclusión: para valores altos de la vida promedio de la confianza hay un incremento significativo de las aplicaciones obsoletas.

Para Vida promedio de la confianza: valores (5,70)



Conclusión: para valores altos de la vida promedio de la confianza hay un incremento de las aplicaciones en re-desarrollo.

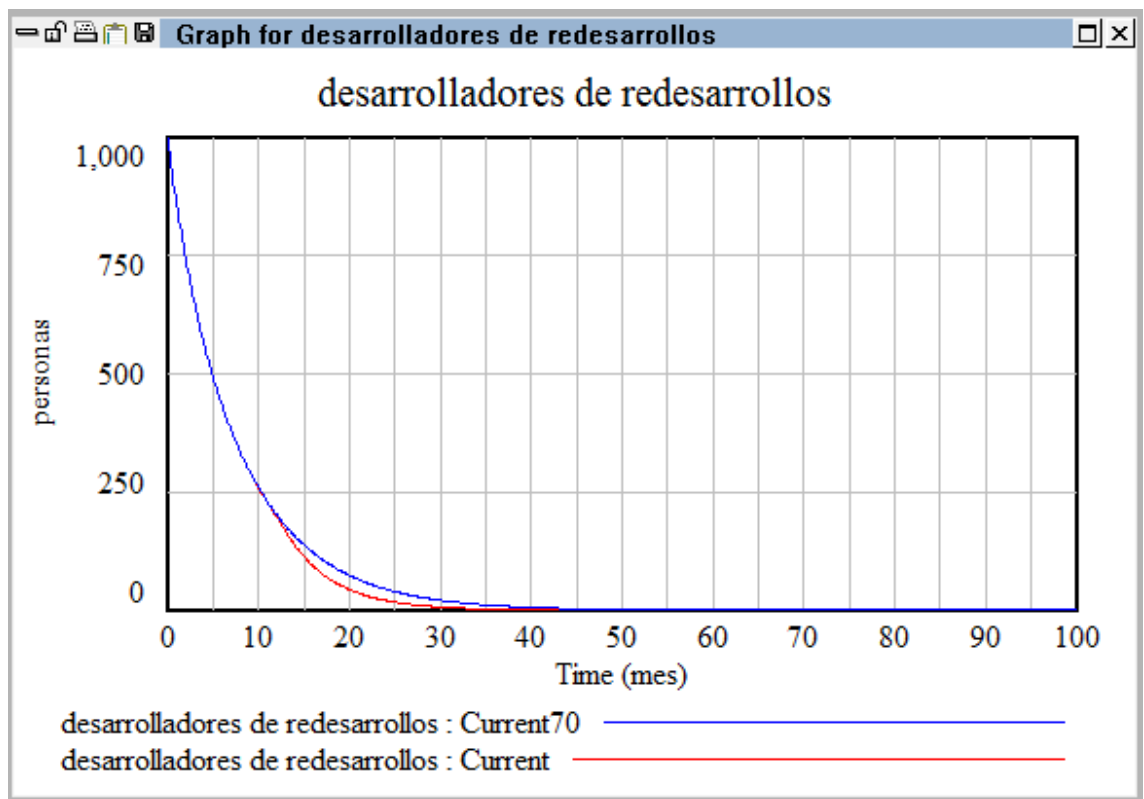
Para Vida promedio de la confianza: valores (5,70)



Conclusión: para valores altos de la vida promedio de la confianza hay leve incremento de los desarrolladores activos de nuevas versiones.

Propuesta para trabajos fututos.

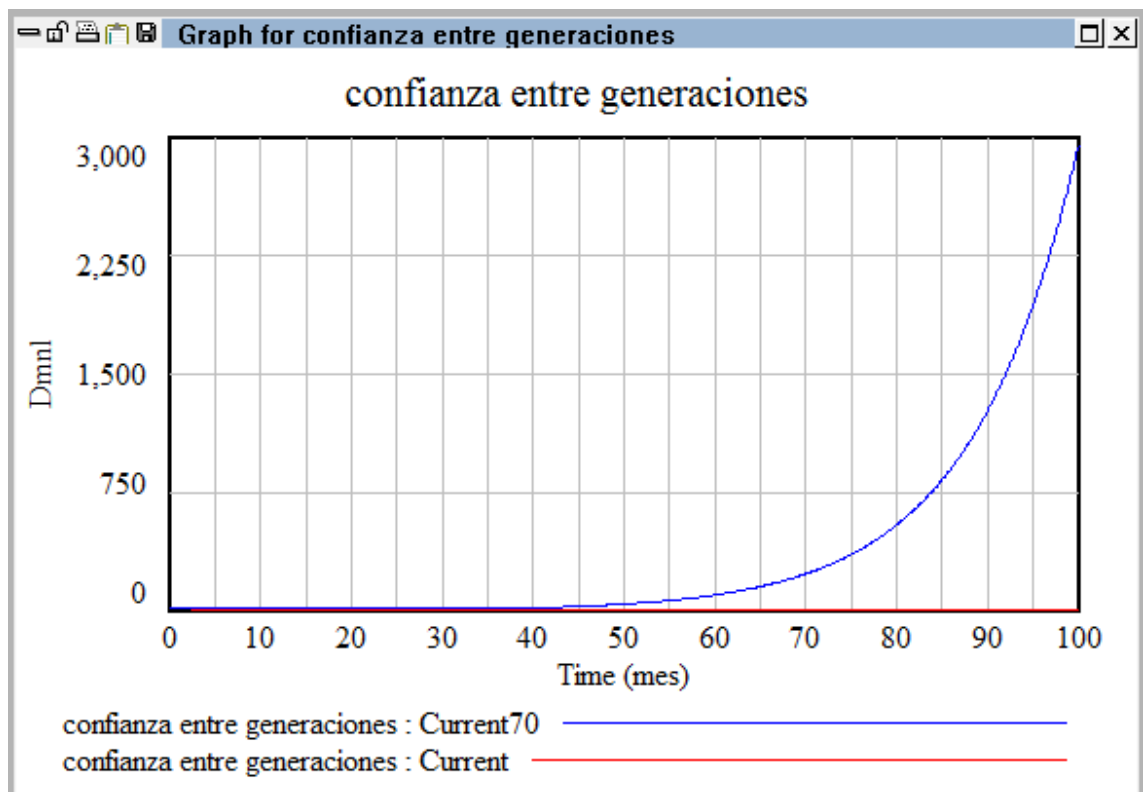
Para Vida promedio de la confianza: valores (5,70)



Conclusión: para valores altos de la vida promedio de la confianza hay leve incremento de los desarrolladores activos de nuevas versiones.

Propuesta para trabajos fututos.

Para Vida promedio de la confianza: valores (5,70)



Conclusión: para valores altos de la vida promedio de la confianza hay un incremento significativo de la confianza entre generaciones.

8.4 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.

El análisis de sensibilidad demuestra que la confianza depende de las condiciones iniciales 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.

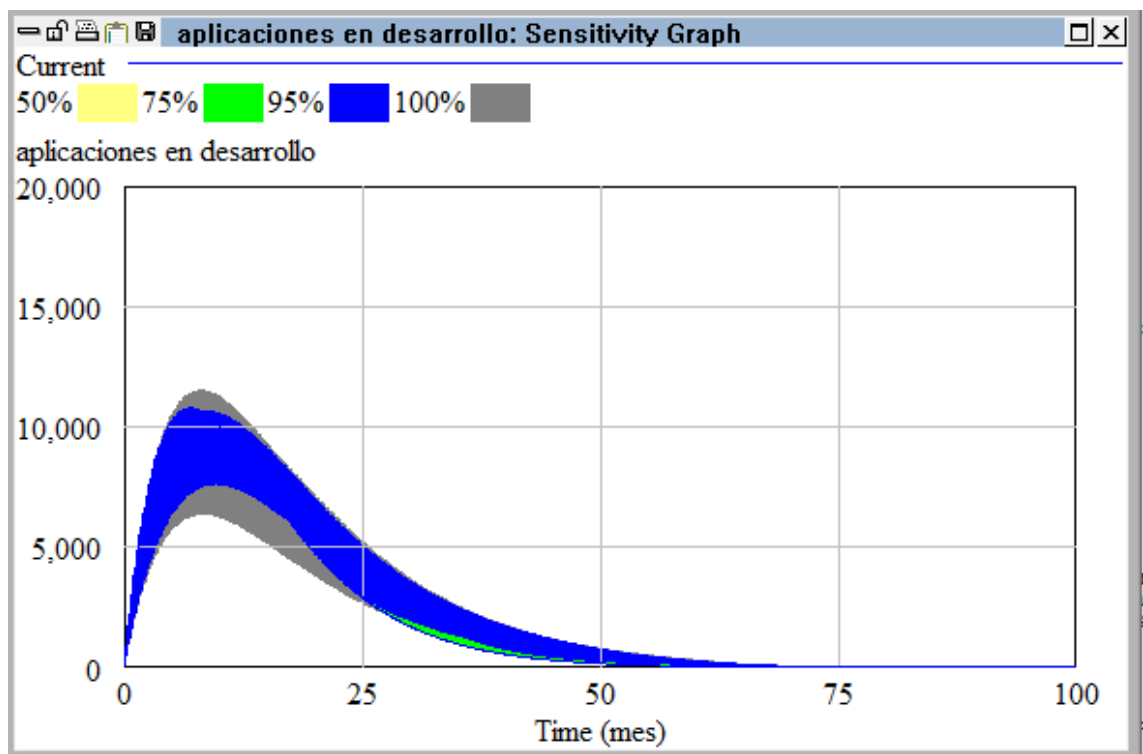


Figura 5ª. Análisis de sensibilidad para las aplicaciones en desarrollo para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

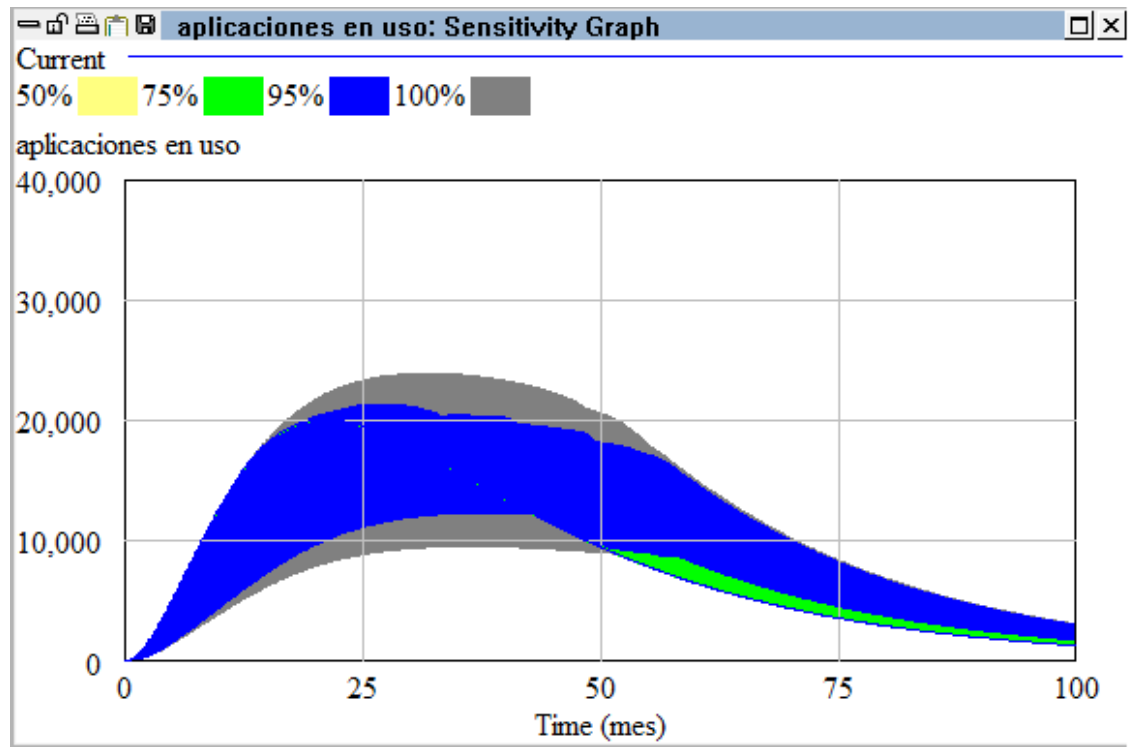


Figura 5b. Análisis de sensibilidad para las aplicaciones en uso para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

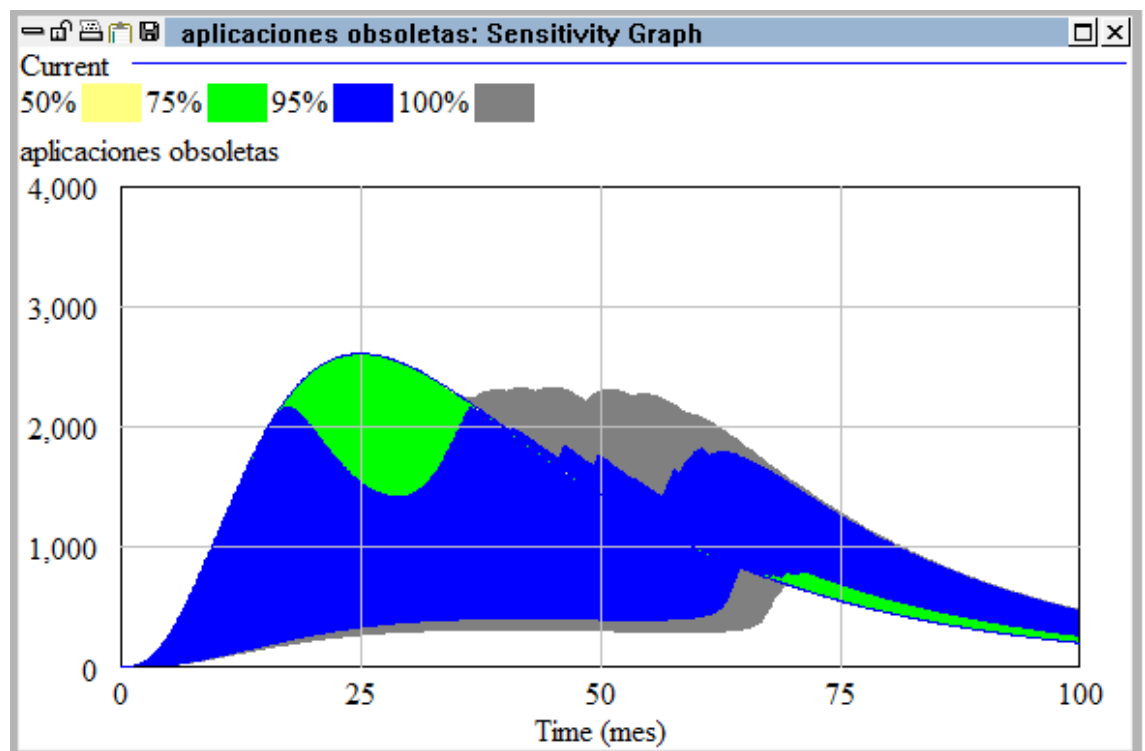


Figura 5c. Análisis de sensibilidad para las aplicaciones obsoletas para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

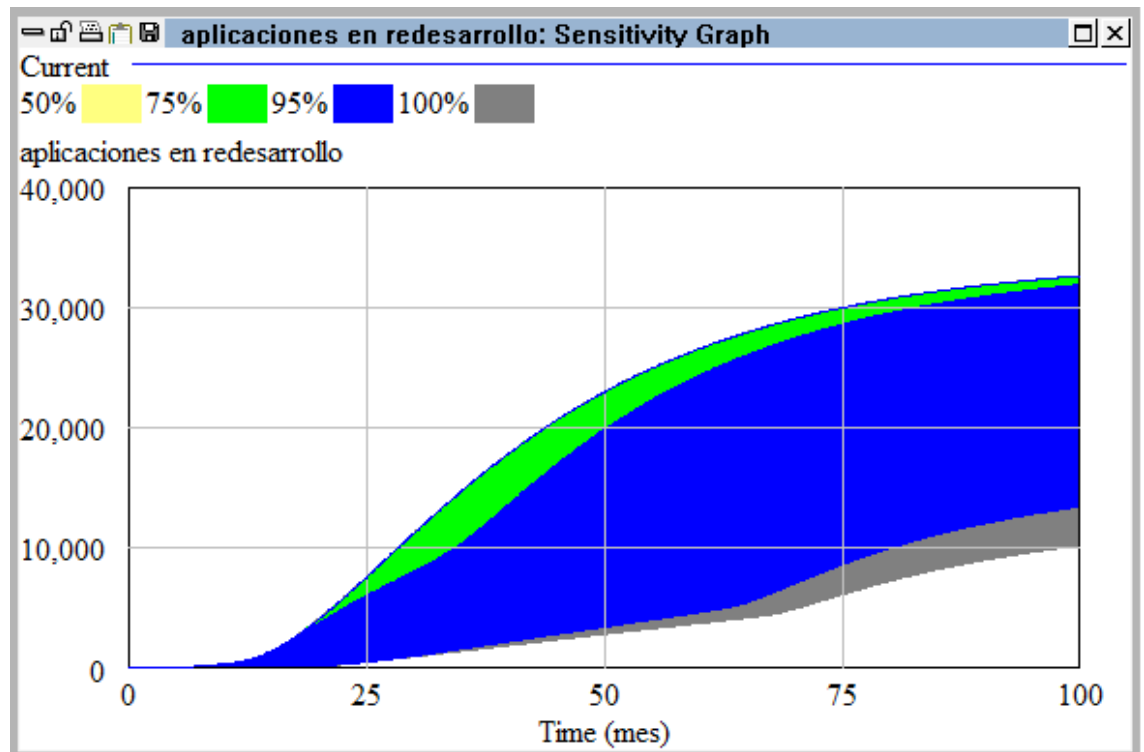


Figura 5d. Análisis de sensibilidad para las aplicaciones en re desarrollo para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

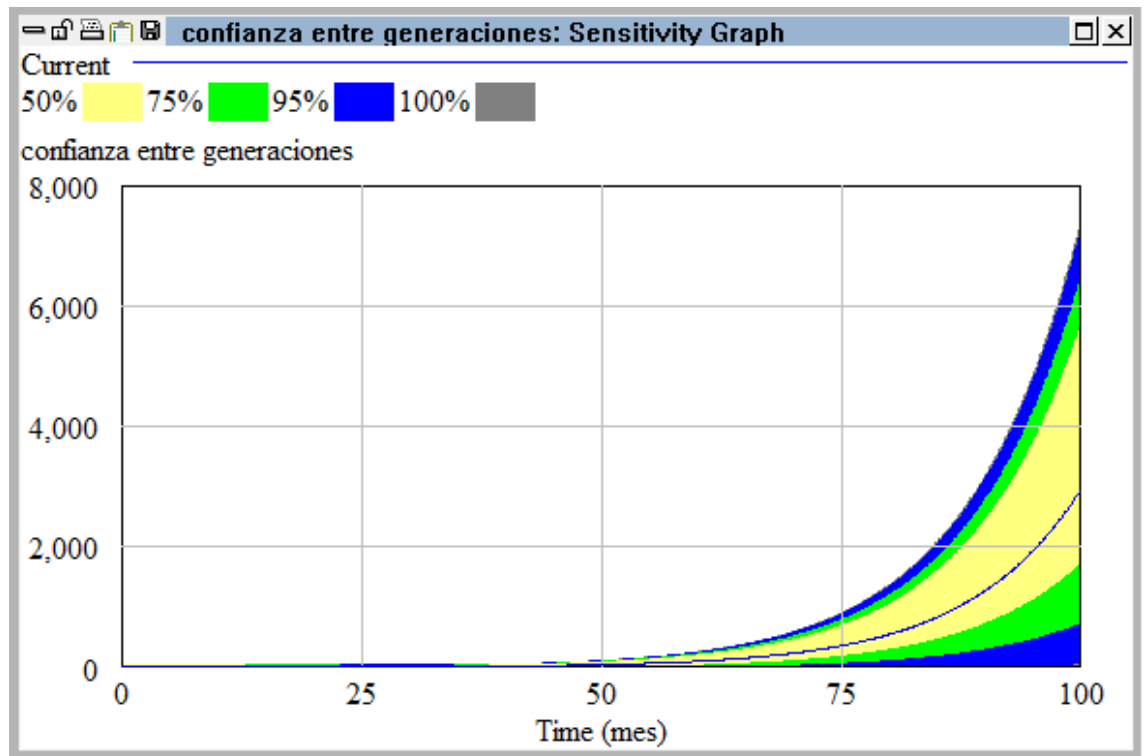


Figura 5^a. Análisis de sensibilidad para la confianza entre generaciones para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

9. Conclusiones

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.

Bajos tiempo de re-desarrollo y a mayor cantidad de desarrolladores de re-desarrollo y desarrolladores de aplicaciones nuevas que hacen transito a desarrolladores de re-desarrollo mantienen la cooperación entre generaciones de desarrolladores de software.

10. Recomendaciones y trabajos futuros

La presente investigación integra el mecanismo de cooperación basado en confianza al proceso de desarrollo de software libre.

A continuación se hace una síntesis de las oportunidades de trabajo futuro que abre esta investigación en cuanto a los diferentes mecanismos de cooperación que podrían ser implementados en función de crear una acción colectiva en el proceso de desarrollo de software libre.

MECANISMO DE COOPERACION	Cooperación por percepción de daño	Cooperación como norma
DESCRIPCION	En grupos que carecen de historia de cooperación previa, la percepción de daño puede ser usada para promover la cooperación. Se da la cooperación, como respuesta a una amenaza de daño por deterioro del recurso.	La cooperación como norma puede aprenderse en el largo plazo luego de un periodo de restricción. Luego de una fuerte restricción y fase de crisis, las personas pueden adoptar norma de cooperación y sostenerlas, gracias

		al desarrollo de la confianza y la cooperación en el corto plazo.
APLICABILIDAD A LA COOPERACIÓN ENTRE GENERACIONES DE DESARROLLADORES DE SOFTWARE	El anuncio de la posibilidad de reducción de cantidad y calidad del software libre, podría ocasionar un efecto dinámico. Esto podría dar inicio a una cooperación auto sostenible que disminuye la expectativa de menos software libre.	Podría implementarse como complemento a otro mecanismo que inicio el ciclo de cooperación. Las generaciones de desarrolladores podrían cooperar luego de un tiempo de restricción al bien común.
EVALUACIÓN SUBJETIVA DE LA EFICIENCIA Y EFECTIVIDAD TEÓRICA DEL MECANISMO	Se generaría una mejora en la cooperación, antes de iniciar una eventual situación de deterioro o falta de disponibilidad de software libre.	Por poco tiempo podría garantizarse la cooperación. Pero en el largo plazo no podría mantenerse

La tabla anterior relaciona dos mecanismos de cooperación y el pre-análisis del investigador, los cuales se proponen como elementos conceptuales y metodológicos para futuros trabajos sobre cooperación en el proceso desarrollo de software libre.

También se propones los siguientes temas para trabajos futuros:

- Desarrolladores activos de nuevas versiones
- Desarrolladores activos de re desarrollos.

11. Bibliografía

Delgado Gutierrez, J.A. (2002) "Análisis Sistémico: Su aplicación a las comunidades humanas": Cie Dossat 2000 Madrid.

Aracil, J. Gordillo F. (1997) "Dinámica de Sistemas": Alianza Universidad Textos. Madrid.

Aracil, J. (1992) "Introducción a la Dinámica de Sistemas": Ed. Alianza editorial AU.Textos. Madrid

Martín, G. j. (2010) "TEORÍA DE EJERCICIOS PRÁCTICOS DE DINAMICA DE SISTEMAS": Barcelona, España

Parra, V. J. & Rezinzew, D. I. (2008) "Cooperación, expectativa y racionamiento en dilemas sociales de recurso de gran escala": Revista de dinámica de sistemas Vol. 4 Núm. 2 .Bucaramanga, Colombia

Parra, V. J. & Rezinzew, D. I. (2013) "Cooperación para el Mejoramiento de la Calidad del Aire": Editorial académica española.

Parra, V. J. & Rezinzew, D. I. (2012) "Evaluación de la Cooperación en Dilemas Sociales de Gran Escala: Un enfoque desde la Dinámica de Sistemas para la evaluación de mecanismos de promoción de la acción colectiva": Editorial académica española.

Muñoz, H. S. & Perez, R. D. & Martinez, M. J. (2008). "E-Learning aplicado a la cooperación como herramienta educativa en la motivación de profesores y alumnos". Madrid:

Barajas, A. & Joost H. & Huergo E (2009). "la cooperación tecnológica en el programa marco de i+d de la unión europea: evidencia empírica para el caso de la empresa española". Madrid:

Instituto de Análisis Industrial y Financiero. Universidad Complutense de Madrid .

Cervera, B, Jaime (2008). "Cooperación Universitaria al Desarrollo: Actualidad y futuro a partir de dos experiencias recientes.". Valladolid, España:

Díaz, A. J. E. & Olaya, F. J. F. (2009) “ConstruColectiva: Guía metodológica para la gestión de proyectos de software basados en metodologías ágiles, utilizando ambientes de desarrollo colaborativo”. Bogotá, Cundinamarca, Colombia:.

Armendáriz, D. (2008) “Colaboración para el crecimiento del SFA. El grupo de PYMES adopta un modelo de trabajo Open Source, amplía su intervención en proyectos y recibe la confianza de instituciones y empresas.”. País Vasco – España: .

Software Engineering Group School of Computer Science and Mathematics Keele University Keele, Staffs ST5 5BG, UK & Department of Computer Science University of Durham Durham,UK
(2007) “ Guidelines for performing Systematic Literature Reviews in Software Engineering“

Geilfus, F. (2009) “80 herramientas para el desarrollo participativo: diagnóstico, planificación, monitoreo, evaluación”. San José, Costa Rica: Instituto Interamericano de Cooperación para la Agricultura (IICA)

Rebollo, C. M. A. & Perez, G. R. & Garcia, B. O. & Barrangan S. R. (2012). “Las comunidades virtuales como potencial pedagógico para el aprendizaje colaborativo a través de las TIC”. : Enseñanza & Teaching: Revista interuniversitaria de didáctica

López Zozal, Carlos & Marticorena Sánchez, Raúl & Santos, Pablo & Maudes, Jesús M. (2010). “Uso de herramientas para la adquisición de competencias transversales asociadas al desarrollo y mantenimiento de software”.

Universidade de Santiago de Compostela. Escola Técnica Superior d'Enxeñaría

: España