

**PROTOTIPO DE SISTEMA DE RESTRICCIONES
PARA DISPOSITIVOS MÓVILES**

**JAMES JEFERSON MUÑOZ CORONEL
HAROLD ANDRÉS MUÑOZ MUÑOZ
CAMILO HERNÁN VILLOTA IBARRA**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2014**

**PROTOTIPO DE SISTEMA DE RESTRICCIONES
PARA DISPOSITIVOS MÓVILES**

**JAMES JEFERSON MUÑOZ CORONEL
HAROLD ANDRÉS MUÑOZ MUÑOZ
CAMILO HERNÁN VILLOTA IBARRA**

**Proyecto de grado presentado como requisito para optar al título de
Ingeniero de Sistemas**

**Asesor:
GONZALO JOSÉ HERNÁNDEZ GARZÓN**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2014**

NOTA DE RESPONSABILIDAD

“La Universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”

Artículo 13° del Acuerdo No 005 de enero 26 de 2010, emanado del Honorable Consejo Directivo de la Universidad de Nariño.

Las ideas y conclusiones aportadas en el siguiente trabajo son responsabilidad exclusiva del autor.

Artículo 1ro del Acuerdo No. 324 de octubre 11 de 1966 emanado del Honorable Consejo Directivo de la Universidad de Nariño.

Nota de aceptación:

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

San Juan de Pasto, Junio de 2014

DEDICATORIA

Dedico este trabajo principalmente a Dios, por permitirme llegar hasta este momento tan importante de mi formación profesional. A mis padres, por ser el ejemplo más importante y por brindarme su apoyo incondicional en todo momento. A mi novia por siempre creer en mí y brindarme todo su cariño. A mis amigos, compañeros y profesores de la Universidad por vivir conmigo esta etapa.

MARCAS REGISTRADAS

Android es una marca registrada de Google Inc., Eclipse es una marca registrada de The Eclipse Foundation, Java es una marca de Oracle, Android Studio es una marca registrada de IntelliJ.

AGRADECIMIENTOS

Al Ingeniero Gonzalo Hernández por su valiosa colaboración en el desarrollo de este proyecto de investigación.

Al plantel de profesores del programa de Ingeniería de Sistemas de la Universidad de Nariño, por brindarnos su incondicional apoyo y orientarnos de la mejor manera en nuestro proceso de formación profesional.

A cada uno de nuestros compañeros y amigos, por los consejos recibidos, los buenos momentos compartidos, por el apoyo incondicional que nos brindaron en todo momento y los fuertes vínculos de amistad que nos mantienen unidos aun por fuera de la Universidad y que de seguro perduraran toda la vida.

A nuestras familias por el apoyo que nos brindan cada día permitiéndonos culminar una etapa más en la academia al igual que el apoyo emocional para salir adelante en el ámbito profesional y personal.

A Dios por permitirnos hacer su voluntad logrando todas las metas que se propone día tras día.

RESUMEN

Actualmente las necesidades de desarrollar aplicaciones más complejas y robustas para dispositivos móviles, debido a la importancia que está tomando el uso de éstos, nos lleva a buscar nuevas metodologías de programación que permitan solucionar problemas específicos, lo que conlleva a tomar a la “programación por restricciones” como una opción de implementación de soluciones para un nuevo mercado y los motores de restricciones como Choco y Gecode están orientados a computadores de escritorio. Esta investigación se realiza con el fin de poder usar este motor en dispositivos con sistemas operativos Android, permitiéndoles desarrollar aplicaciones que den solución a problemas complejos donde la mejor opción puede ser tomar un paradigma de programación por restricciones. En este documento se presentan los resultados del proyecto de investigación que permite el uso del motor de restricciones de Choco en dispositivos con sistema operativo Android.

Palabras clave: Programación por restricciones, aplicaciones Android, Android Studio, Choco, Gecode.

ABSTRACT

Currently the need to develop more complex and robust applications for mobile devices leads us to seek new programming methodologies that allow solving specific problems, and the importance that is taking the use of mobile devices in society for all kind of tasks, which emphasizes the development of applications for a new market. Research is done in order to use these engines on devices with Android operating systems. For the development of applications that provide solutions to complex problems is used the programming paradigm by restrictions. The results of the research project that allows using the engine Choco restrictions on devices with Android operating system are presented in this document.

Keywords: Constraint Programming, Android applications, Android studio, Choco, Gecode.

CONTENIDO

	Pág.
INTRODUCCIÓN	14
1. ESTADO DEL ARTE.....	18
1.1 PROBLEMAS N-P.....	18
1.2 PROGRAMACIÓN POR RESTRICCIONES	18
1.3 GECODE.....	19
1.4 CHOCO.....	21
1.5 ANDROID.....	21
2. ANTECEDENTES	22
3. METODOLOGIA.....	25
4. DESARROLLO DEL PROYECTO DE INVESTIGACIÓN.....	27
4.1 ETAPA I: CONCEPTUALIZACIÓN	27
4.2 ETAPA II: ACOPLAMIENTO DEL SISTEMA BASE	29
4.2.1. Desarrollo e Implementación del prototipo CONSTDROID.....	29
4.2.1.2. Requisitos funcionales del prototipo	30
4.2.2. Versión portable Android Studio with Choco	39
4.2.2. Acoplamiento con Gecode.	40
4.3. ETAPA III: CREACIÓN DE SOLUCIONES	41
4.3.1. Sudoku.	41
4.3.2. Magic Square.	42
4.4. ETAPA IV: PRUEBAS.....	44
4.5. ETAPA V: PUBLICACIÓN.....	49
5. ANÁLISIS DE RESULTADOS	50
6. CONCLUSIONES.....	51
BIBLIOGRAFÍA.....	53

LISTA DE CUADROS

	Pág.
Cuadro 1. Actividades planificadas	26
Cuadro 2. Contenido de la cátedra Programación por Restricciones	27
Cuadro 3. Contenido de la cátedra Aplicaciones Android	28
Cuadro 4. Dispositivos Android y características	44
Cuadro 5. Tiempos de Respuesta en Magic Square.....	45
Cuadro 6. Tiempos de Respuesta en Sudoku	45

LISTA DE IMÁGENES

	Pág.
Imagen 1. Arquitectura por Capas	30
Imagen 2. Modelo de análisis	31
Imagen 3. Diagrama de actividades Registrar Variables	32
Imagen 4. Diagrama de actividades Registrar Restricciones	33
Imagen 5. Diagrama de actividades Solucionar Problema	34
Imagen 6. Diagrama de secuencia para Registrar Variables	35
Imagen 7. Diagrama de secuencia para Registrar Restricciones	35
Imagen 8. Diagrama de secuencia para Solucionar Problema	36
Imagen 9. Interfaz de Usuario Registrar Variables	37
Imagen 10. Interfaz de Usuario Registrar Restricciones.....	38
Imagen 11. Interfaz de Usuario Solucionar Problema	38
Imagen 12. Interfaz de Usuario ver Solución del Problema	39
Imagen 13. Pantallazo Aplicación Sudoku.....	42
Imagen 14. Pantallazo Aplicación Magic Square	44
Imagen 15. Plantilla Sudoku	46
Imagen 16. Relación entre Dispositivos y tiempos de respuesta para Magic Square 3x3	46
Imagen 17. Relación entre Dispositivos y tiempos de respuesta para Magic Square 5x5	47
Imagen 18. Relación entre Dispositivos y tiempos de respuesta para Magic Square 7x7	47
Imagen 19. Relación entre Dispositivos y tiempos de respuesta para Sudoku Vacío	48
Imagen 20. Relación entre Dispositivos y tiempos de respuesta para Sudoku con Plantilla	48

GLOSARIO

Android: Es un sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles con pantalla táctil como teléfonos inteligentes o tabletas.

C++: Lenguaje de programación híbrido donde se pueden desarrollar aplicaciones orientadas a objetos como estructuradas.

Choco: Biblioteca en el lenguaje de programación Java para el desarrollo de sistemas y aplicaciones basados en restricciones.

Gecode: Biblioteca en el lenguaje de programación C++ para el desarrollo de sistemas y aplicaciones basados en restricciones.

Java: Lenguaje de programación orientado a desarrollar aplicaciones orientadas a objetos.

NDK: Es un conjunto de herramientas que permiten incrustar código máquina nativo compilado en lenguajes C y/o C++ para Android.

Programación: es el proceso de diseñar, codificar, depurar un conjunto de instrucciones ordenadas y con un sentido, denominadas como código fuente.

Paradigma: es una propuesta tecnológica que es adoptada por una comunidad de programadores el cual se orienta a tratar únicamente de resolver uno o varios problemas claramente delimitados.

SDK: es un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto, en este caso para Android.

INTRODUCCIÓN

En la actualidad el boom de la tecnología móvil avanza a pasos agigantados, la solución de problemas y el desarrollo de aplicaciones para dispositivos móviles se ha convertido en un eje central de desarrollo, todas las actividades que se realizan en el hogar, en las empresas y en la industria se quieren controlar y procesar desde estos dispositivos, pero hoy en día se presentan muchísimos problemas a los cuales no se les puede dar solución mediante estos dispositivos porque sus capacidades los limitan para poder correr procesos muy complejos, aun después de alcanzar importantes mejoras, la capacidad de memoria, la capacidad de procesamiento y la batería no son suficientemente potentes.

Este trabajo de investigación implementa un prototipo de sistema de restricciones en dispositivos móviles con sistema operativo Android para poder desarrollar aplicaciones que den solución a problemas N-P. La principal razón para desarrollar este prototipo es que los dispositivos móviles cuentan con limitados recursos hardware y con este prototipo se aprovechará y reducirá el uso de los recursos para dar solución a este tipo de problemas.

Se incluye el motor de restricciones bajo sistema operativo Android de tal manera que éste pueda utilizar sus bibliotecas y ver hasta qué punto un dispositivo móvil puede llegar a soportar el motor y que aplicaciones se pueden implementar sobre este tipo de dispositivos.

TEMA

TÍTULO

PROTOTIPO DE SISTEMA DE RESTRICCIONES PARA DISPOSITIVOS MÓVILES

LÍNEA DE INVESTIGACIÓN

Este proyecto corresponde a la línea de investigación de Lenguajes de Programación.

ALCANCE Y DELIMITACIÓN

Este proyecto pretende indagar acerca de las posibilidades de implementar un prototipo de sistema de restricciones para dispositivos móviles con sistema operativo Android que permite solucionar problemas N-P en estos dispositivos,

además se pretende establecer la factibilidad de montar o no totalmente el motor de restricciones en estos dispositivos. Para el desarrollo del prototipo se pretende trabajar con lenguajes de programación como C++ y Java y con el motor de restricciones Gecode (C++) y Choco (Java), además en cuanto a programación móvil se pretende trabajar con el kit DS-5 (C++) y el kit de herramientas SDK (Java).

MODALIDAD

Este proyecto corresponde a la modalidad trabajo de investigación.

DESCRIPCIÓN DEL PROBLEMA

Planteamiento del problema

Solucionar problemas cuya solución requiere un tiempo de ejecución no polinomial, ha sido un gran reto para los programadores debido al uso excesivo de los recursos hardware, además de la complejidad algorítmica requerida para plantear y codificar este tipo de problemas.

La programación por restricciones es usada especialmente para la resolución de problemas combinatorios particularmente difíciles. Mediante la especificación de un conjunto de restricciones (ecuaciones), las cuales deben ser satisfechas, en lugar de especificar los pasos para obtener dicha solución, el motor de restricciones trabaja bajo las restricciones planteadas y genera todas las posibles soluciones. Por lo general este paradigma se encarga de solucionar problemas especialmente en las áreas de planificación, programación de tareas, calendarización, inteligencia artificial, computación genética y juegos lógicos.

Teniendo en cuenta que el crecimiento en el mercado de los dispositivos móviles como aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red y con memoria limitada, toman cada día más importantes lugares en la industria y en la vida cotidiana, la inversión en los procesos de negocio se hacen ahora enfocados a tecnologías de este tipo. Entre estos dispositivos se encuentran: Smartphone, tabletas entre otros, diseñados específicamente para ofrecer la posibilidad de instalar programas para incrementar la capacidad para el procesamiento de datos y la conectividad.

Ya que hoy en día las aplicaciones no son únicamente para resolver problemas sencillos sino problemas de tipo N-P no se ha tenido éxito ya que se necesitan muchos recursos computacionales para poder solucionarlos¹, el diseño de un

¹ CORMEN, T., LEISERSON, C., RIVEST, R. & STEIN, C. (2009). Introduction to Algorithms, The MIT Press. USA

dispositivo móvil es quien genera las limitaciones de hardware principalmente en memoria RAM, procesador y disco duro porque desde sus inicios se quiso un dispositivo pequeño, portátil y que tenga pequeñas utilidades por lo cual realizar aplicaciones más robustas requieren hardware de gran capacidad y un aumento en el tamaño.

Actualmente las investigaciones en cuanto programación basado en restricciones bajo dispositivos móviles es escasa como también la utilización de motores de restricciones, debido a esto no se tiene conocimiento acerca de la viabilidad y antecedentes de la implementación bajo estos dispositivos por tal motivo se hace necesario realizar investigaciones acerca de la posibilidad de implementar un motor de búsqueda de soluciones basado en restricciones para dispositivos móviles y así poder dar solución a los problemas no polinomiales que han sido un verdadero motivo por el cual no se ahonda en la programación de aplicaciones de mayor complejidad.

FORMULACIÓN DEL PROBLEMA

¿Cómo implementar un prototipo de sistema de restricciones en dispositivos móviles bajo un sistema operativo Android para solución de juegos combinatorios?

SISTEMATIZACIÓN DEL PROBLEMA

- ¿Qué funcionalidades del motor de restricciones Gecode y Choco se pueden implementar en el prototipo de sistema de restricciones en dispositivos móviles que usen el sistema operativo Android?
- ¿Qué problemas no polinomiales se pueden solucionar mediante este prototipo de sistema de restricciones en los dispositivos móviles con sistema operativo Android?
- ¿Bajo cuáles dispositivos y con cuáles características se puede implementar el prototipo de sistema de restricciones?
- ¿Cómo socializar los resultados de la investigación?

OBJETIVOS

Objetivo general

Diseñar e implementar un prototipo de sistema de restricciones para dispositivos móviles bajo sistema operativo Android que permita solucionar juegos combinatorios.

Objetivos específicos

Examinar y comprender la arquitectura de implementación de un motor de programación por restricciones como Gecode bajo lenguaje C++ y Choco bajo lenguaje Java

Estudiar los fundamentos de la programación bajo sistema Android utilizando lenguaje C++ y Java.

Diseñar una solución basada en la arquitectura de un motor de programación por restricciones acoplada al sistema de programación bajo sistema operativo Android.

Preparar y socializar un artículo con los resultados de la investigación en un evento nacional.

JUSTIFICACIÓN

La gran demanda de dispositivos móviles y la capacidad de éstos para dar soluciones rápidas, versátiles y a la mano, hace más que pertinente la necesidad de incrementar el rango de problemas solucionables por medio de software que corra sobre estos dispositivos. Estos dispositivos aún son muy limitados en hardware para poder correr aplicaciones complejas, los cuales requieren una cantidad considerable de recursos como procesador y memoria RAM.

Aumentar la capacidad de solucionar problemas más complejos en un dispositivo móvil con las limitaciones en hardware que posee se convierte en una necesidad en el entorno en el que se encuentran, para darle mayor campo de acción a estos dispositivos y poder solucionar los problemas complejos (N-P) desde cualquier parte y en cualquier momento.

La implementación de un motor de restricciones en Android permitirá desarrollar aplicaciones para dispositivos Android basadas en este motor que permitirá reducir considerablemente el uso de recursos hardware en la solución de problemas y así lograr solucionar problemas que anteriormente no era posible debido a su complejidad. Por esto, un motor de restricciones funcionando en un dispositivo móvil brinda la posibilidad de crear aplicaciones que solucionen problemas N-P lo que conlleva a dar mayor versatilidad y funcionalidad a estos dispositivos.

1. ESTADO DEL ARTE

1.1 PROBLEMAS N-P

N-P es el acrónimo en inglés “Nondeterministic polynomial time” (“Tiempo polinomial no determinista. Es el conjunto de problemas que no pueden resolverse en un tiempo polinómico ya que aumentan exponencialmente con el tamaño del problema o el número de variables. Por lo tanto N-P es el conjunto de problemas de decisiones en el que se puede comprobar y responder rápidamente si se tiene la solución delante de nosotros pero el problema es difícil de resolver si no se posee un dato o información adicional para resolverlo ya que el número de combinaciones es demasiado grande. Por ejemplo, acomodar los números del 1 al 4 (24 maneras) puede llevar tal vez un minuto, pero acomodar los números del 1 al 8 (40320 maneras) llevará mucho más, a pesar de que 8 es apenas dos veces mayor que 4. En este tipo de problemas se llega al momento en que se consideran insolubles porque para resolverlos necesitarías más tiempo del que hay en el Universo.

1.2 PROGRAMACIÓN POR RESTRICCIONES

La programación de restricciones es una tecnología o paradigma de programación de software utilizada para la descripción y posterior resolución de problemas complejos, particularmente combinatorios, especialmente en las áreas de planificación y programación de tareas². Muchos de estos problemas pueden modelarse como problemas de satisfacción de restricciones y resolverse usando técnicas de programación de restricciones. Esto incluye problemas de áreas tales como inteligencia artificial, investigación operativa, bases de datos, sistemas expertos, etc.

Las soluciones se expresan en relaciones entre las variables y son expresadas en términos de restricciones. La especificación del conjunto de restricciones, deben ser satisfechas por cualquier solución del problema planteado, en lugar de especificar los pasos para obtener dicha solución. Así pues el objetivo de la programación por restricciones es resolver problemas que se puedan representar en función de variables y restricciones

² K. Apt, Principles of Constraint Programming, Ed. Cambridge University Press, USA, 2003.

La programación por restricciones se fundamenta en las siguientes teorías:

Resolución de Restricciones:

Trata problemas definidos sobre dominios infinitos a los cuales se les aplica una serie de restricciones que se deben cumplir para dar solución al problema.

Problema de Satisfacción de Restricciones (CSP):

Un conjunto de variables, un dominio finito de posibles valores, y un conjunto de restricciones que condicionan los valores que las variables puedan tener simultáneamente.

Solución a un CSP:

Es una asignación de un valor de su dominio a todas las variables, de manera que todas las restricciones son satisfechas en el mismo momento³.

Motores de Resolución (Solvers):

Se encargan de la implementación de algoritmos para resolver restricciones permitidas acorde con la teoría de restricciones. Para posteriormente colocarlas en un almacén de restricciones. A partir de esto se prueba la satisfacción de las mismas, se simplifican y si es posible se resuelven las restricciones.

1.3 GECODE

Gecode (del inglés Generic Constraint Development Environment) es un conjunto de herramientas para el desarrollo de sistemas y aplicaciones basados en restricciones en el lenguaje de programación C++⁴. Proporciona un solucionador de restricciones con un alto desempeño, ya que es modular y extensible, su eficiencia y portabilidad lo hacen ideal para el desarrollo de sistemas y aplicaciones basados en restricciones. Gecode es:

Abierto

Gecode es radicalmente abierto para la programación: puede ser fácilmente conectado con otros sistemas. Es compatible con la programación de nuevas restricciones, estrategias de ramificación y motores de búsqueda. También es posible crear nuevos dominios de variables que se pueden programar en el mismo nivel de eficiencia como las variables que están predefinidas en Gecode.

³ APT, K., (2003). Principles of Constraint Programming. Cambridge. UK.

⁴ SCHULTE, TACK, LAGERKVIST, Modeling and Programming with Gecode., 2003

Exhaustivo

Gecode tiene un conjunto completo de características: restricciones sobre los números enteros, booleanos, conjuntos y flotantes; C++ como capa de modelado; avanzada ramificación heurística, muchos motores de búsqueda, ruptura de la simetría automática (LDSB), soporte de Mini Zinc, y muchos más.

Eficiente

Gecode ofrece un excelente rendimiento con respecto a tiempo de ejecución y uso de memoria. Se ganó todas las medallas de oro en todas las categorías en los Desafíos MiniZinc 2008-2012

Documentado

Gecode viene con tutorial completo (más de 500 páginas), la documentación de referencia permite a los usuarios centrarse en diferentes tareas de modelado y programación.

Libre

Gecode se distribuye bajo la licencia MIT y está catalogado como software libre por la FSF. Todas las partes, incluyendo documentación de referencia, las implementaciones de restricciones globales, y los ejemplos están disponibles como código fuente para descargar libremente.

Portátil

Gecode está implementado en C++ que sigue atentamente el estándar C++. Puede ser compilado con modernos compiladores C++ y se puede ejecutar en una amplia gama de máquinas.

Paralelo

Gecode tiene la capacidad de aprovechar los múltiples núcleos de hardware que cuentan las máquinas actuales para la búsqueda en paralelo, proporcionando un sistema base ya eficiente con una ventaja adicional.

Probado

Gecode utiliza un conjunto de pruebas con casi 50.000 casos de prueba diferentes para llegar a una cobertura de prueba cerca de 100%.

1.4 CHOCO

Choco es una biblioteca de Java para problemas de satisfacción por restricciones (CSP) y programación por restricciones (CP)⁵. Está construida en un mecanismo de propagación basada en eventos con estructuras backtrackalbe. Choco es un software de código abierto, distribuido bajo licencia BSD y desarrollados principalmente por personas en École des Mines de Nantes (France) y es respaldada financieramente por Bouygues SA y Amadeus SA.

1.5 ANDROID

Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Android es un sistema operativo inicialmente pensado para teléfonos móviles, basado en GNU/LINUX, totalmente libre, para desarrollar aplicaciones sobre el o para realizar modificaciones a la plataforma.

El sistema permite programar aplicaciones en una variación de Java Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java.

⁵ LABURTHER, F., JUSSIEN, N. (2011). Choco Solver - Documentation. <http://Choco.mines-nantes.fr/>

2. ANTECEDENTES

Los siguientes proyectos usan Gecode para desarrollar sus funcionalidades:

- QeCode

Es un código abierto de satisfacción de problemas por restricciones cuantificados, basados en Gecode. QCSP son una extensión de CSP clásico en el que las variables pueden ser cuantificadas universalmente. Permite modelar la incertidumbre acerca de los datos o la presencia de un adversario. Las solicitudes se encuentran en las zonas de reproducción del juego, manejo de incertidumbre, planificación conformante, programación robusta, comprobación de modelos y pruebas.

- Tulipa

La Arquitectura de análisis lingüístico Tübingen utiliza Gecode como parte de un programa de análisis de variantes de árbol de Gramática contiguo.

- CPSP

El paquete CPSP-tools ofrece programas para resolver exactamente y completamente los problemas típicos de los eStudios que utilizan modelos de proteínas celosía 3D. Entre las tareas que se abordan son la predicción de estructuras óptimas a nivel mundial y / o sub-óptimo, así como el diseño y la secuencia de exploración de la red neutral.

- CP4IM

En lugar de crear a mano algoritmos imperativos, en la programación con restricciones mediante declaración se especifica un problema por medio de las limitaciones que necesita para satisfacer. Un solucionador genérico buscará entonces efectivamente las soluciones que satisfacen las restricciones. Es un tipo de Programación con restricciones de conjunto de elementos de Minería (CP4IM) es un enfoque declarativo a la restricción basada en la minería conjunto de elementos.

Los siguientes proyectos usan programación por restricciones para desarrollar sus funcionalidades:

- Metodología para crear aplicaciones de programación por restricciones como servicio web.

Como en programación por restricciones los desarrollos no son visibles debido a que se desarrollaron en una máquina específica con un software específico que de no tenerse instalado no se pueden correr.

- Minimización de costos en cadenas de suministro

Para satisfacer el problema de la cadena de suministros se debe cumplir con la satisfacción del requerimiento de un cliente, cumplir con estándares óptimos de calidad, ofrecer el producto a precios razonables, emplear tiempos de servicio óptimos y ejecutar todas las etapas de la cadena al menor costo posible.

- Problema de secuenciamiento de Aviones

El problema de secuenciamiento de aviones se presenta cuando a la entrada del rango del radar (horizonte del radar) del controlador de tráfico aéreo (ATC) en un aeropuerto, un avión requiere que el ATC le asigne un tiempo de aterrizaje, y si más de una pista está en uso asignarle la pista donde aterrizar.⁶

- Gestión del espectro radioeléctrico

El problema a solucionar es el de la gestión del espectro radioeléctrico que consiste en realizar el proceso de asignación de un grupo de frecuencias o canales a un operador o servicio en una región, satisfaciendo restricciones y buscando que la asignación maximice el número de frecuencias asignadas en una banda del espectro

Los siguientes proyectos usan Choco para desarrollar sus funcionalidades:

- Herramienta para el modelado y configuración de modelos de características.

El objetivo del software es crear la infraestructura adecuada para una rápida y fácil producción de sistemas software similar, destinado a un mismo segmento de mercado. Dado que estos productos son similares, comparten una serie de aspectos comunes pero también presentan ciertas variaciones entre ellos.

- Emparejamiento automático de servicios web usando programación con restricciones.

Las aplicaciones basadas en servicios web y las perspectivas sobre su crecimiento, han propiciado la aparición de nuevos desafíos en el desarrollo de este tipo de sistemas, el emparejamiento automático de servicios web es una de las áreas en las que aún se requieren esfuerzos de investigación.

⁶ AVISPA-UNIVERSIDAD DEL VALLE, En <http://avispa.univalle.edu.co/site/?q=node/22>

Aplicaciones resueltas con restricciones sobre Android:

Actualmente no se encuentran proyectos de aplicaciones que se desarrollen por medio de programación por restricciones sobre Android.

3. METODOLOGIA

La metodología para el desarrollo del proyecto va de la mano con al proceso de construcción de software, el cual se fundamenta en la teoría de Ingeniería de Software y en el modelo de programación colaborativo.

El proyecto consta de las siguientes etapas:

ETAPA I: Conceptualización

En esta fase se realiza la documentación y apropiación de los conceptos de programación por restricciones (Choco / Gecode) y sobre Android.

ETAPA II: Acoplamiento del sistema base

En esta fase se recompilará el código fuente de Gecode (disminuido) o se recrearán las bibliotecas de Choco para ser utilizado en el sistema operativo Android.

ETAPA III: Creación de soluciones

En esta fase se crearán las operaciones para poder programar con restricciones básicas usando el motor de restricciones ya acoplado.

ETAPA IV: Pruebas

En esta fase se evaluará el funcionamiento del prototipo de motor de restricciones para dispositivos móviles.

ETAPA V: Publicación

En esta etapa del proceso se sustentará y presentará el informe donde se darán a conocer los resultados obtenidos. (ver cuadro1)

Cuadro 1. Actividades planificadas

ETAPA	ACTIVIDADES LÓGICAS ANTERIORES	ACTIVIDADES PLANIFICADAS			ACTIVIDADES LÓGICAS POSTERIORES
		ORDEN	DETALLE	DURACIÓN	
I	-	A	Apropiación del concepto de programación por restricciones aplicando los ejercicios con Gecode.	4	B
	A	B	Aplicación de programación por restricciones con Choco.	4	D / E
	-	C	Apropiación de la programación bajo sistema operativo Android.	8	D / E
II	B C	D	Recompilar el código C++ de Gecode en el sistema operativo Android.	8	F
	B C	E	Reconstruir las bibliotecas en Java de Choco para utilizarlas en el sistema operativo Android.	8	F
III	D / E	F	Programación de propagadores aritméticos básicos sobre el motor de restricciones.	8	G
IV	F	G	Ejecución de pruebas evaluando el funcionamiento del motor de restricciones sobre S.O. Android.	4	H
V	F	H	Elaboración del informe final y Publicación de resultados	4	-

4. DESARROLLO DEL PROYECTO DE INVESTIGACIÓN

El proyecto consta de las siguientes etapas:

4.1 ETAPA I: CONCEPTUALIZACIÓN

En esta fase se realizó la apropiación de los conceptos de programación por restricciones con Gecode y Choco en el curso de programación Avanzada, cátedra impartida en la Universidad de Nariño, el curso se programó para 6 meses con una intensidad semanal de 4 horas y distribuido en cuatro unidades cuyas temáticas se muestran en el cuadro 3.

Cuadro 2. Contenido de la cátedra Programación por Restricciones

No. Unidad	Unidad	Temas
1	Análisis matemático	Introducción a problemas No Polinómicos Complejidad de Algoritmos, Casos posibles Mejor (Ω) Peor (O) y Medio(θ) Análisis de soluciones por medio de algoritmos convencionales
2	Modelamiento CSP	Modelamiento de un CSP Implementación de un CSP en utilizando un motor de restricciones Implementación de aplicaciones bajo restricciones utilizando ejercicios conocidos. (SendMoreMoney, Grocery)
3	Motores de Restricciones	Codificación de aplicaciones con Gecode Codificación de aplicaciones con Choco
4	Propagación y Distribución	Arboles de Búsqueda Heurísticas de Distribución Análisis de propagadores

Además se realizó la apropiación de los conceptos de programación en dispositivos Android, cátedra impartida en la Universidad de Nariño, el curso se programó para 6 meses con una intensidad semanal de 4 horas y distribuido en cinco unidades cuyas temáticas se muestran en el cuadro 4.

Cuadro 3. Contenido de la cátedra Aplicaciones Android

No. Unidad	Unidad	Temas
1	Introducción a Android y entorno de desarrollo	Instalación de Android IDE de desarrollo en Android
2	Diseño del interfaz del usuario - Vistas y Layout	Layout (LinearLayout) Layout (TableLayout) Layout (RelativeLayout) Layout (FrameLayout) Layout (ScrollView LinearLayout)
3	Controles, actividades, Barra de Acciones y Preferencias	Controles y capturas de Eventos
4	Ciclo de vida de una Actividad	Control de ciclo de vida de una actividad
5	Bases de datos	Almacenamiento de datos mediante la clase SharedPreferences Almacenamiento de datos en un archivo de texto en la memoria interna. Almacenamiento de datos en un archivo de texto localizado en una tarjeta SD Almacenamiento en una base de datos SQLite

4.2 ETAPA II: ACOPLAMIENTO DEL SISTEMA BASE

4.2.1. Desarrollo e Implementación del prototipo CONSTDROID. Para el desarrollo del prototipo CONSTDROID se siguió la metodología de proceso Unificado Racional (Rational Unified Process) es una metodología para desarrollo de software que permite ajustar la metodología a las necesidades del proyecto, equilibrando los esfuerzos, talento humano y tiempos para llevar a cabo en los mejores términos dicho proyecto resultando apropiada para cumplir el objetivo de esta investigación.

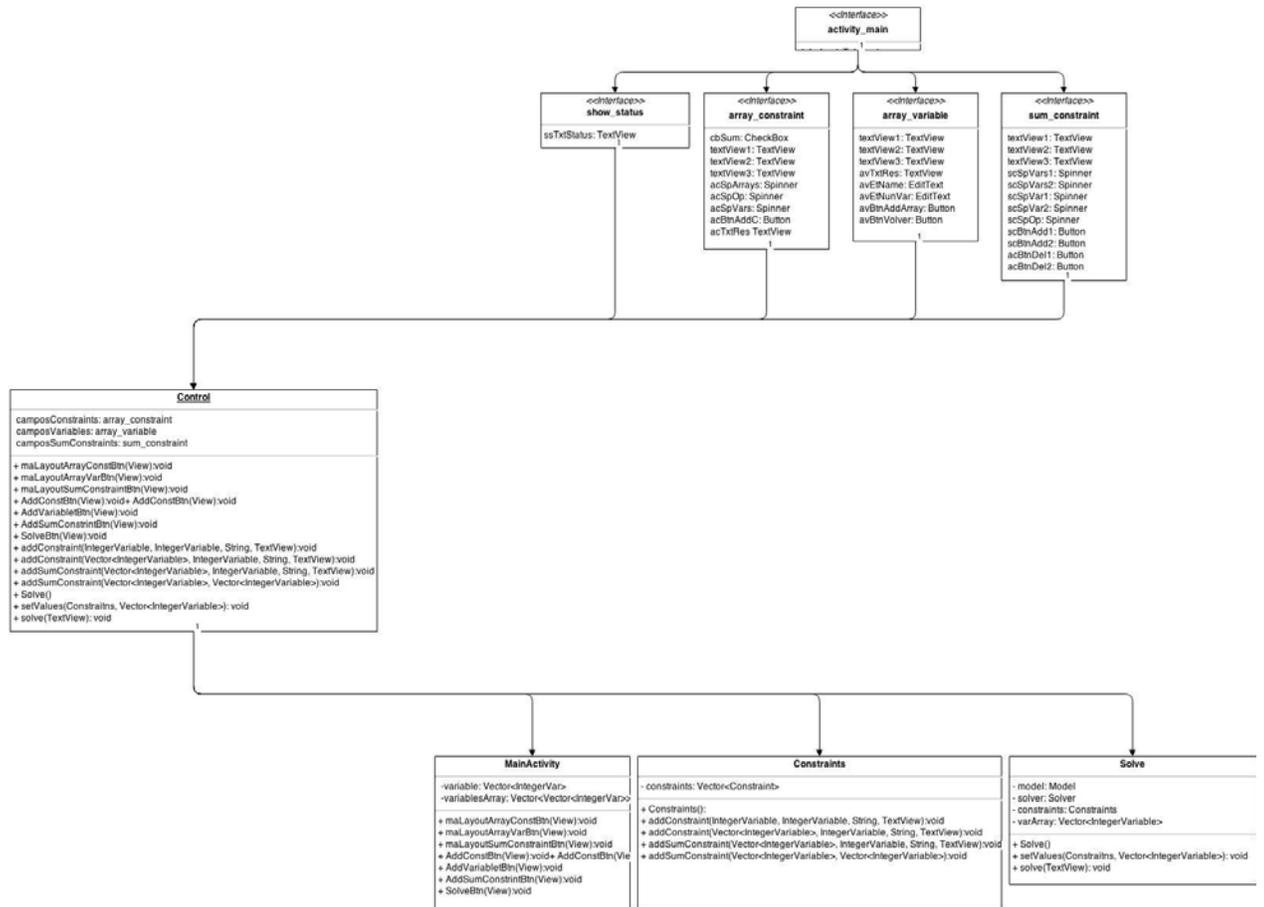
Un aspecto importante que permite llevar a cabo con satisfacción la implementación de la metodología RUP es el cumplimiento de las fases, aplicando la creación de la arquitectura del prototipo, definición de requerimientos, proceso de análisis y diseño con sus diagramas bajo el lenguaje de modelado UML (diagramas de casos de uso, diagrama de actividades y diagramas de secuencia), además de la implementación y despliegue⁷ y que se describe a continuación: (ver imagen 1)

a. Fase Modelo de negocios

b. Arquitectura del prototipo

⁷ En: <http://www.utvm.edu.mx/OrganoInformativo/orgJul07/RUP.htm>

Imagen 1. Arquitectura por Capas



4.2.1.2. Requisitos funcionales del prototipo

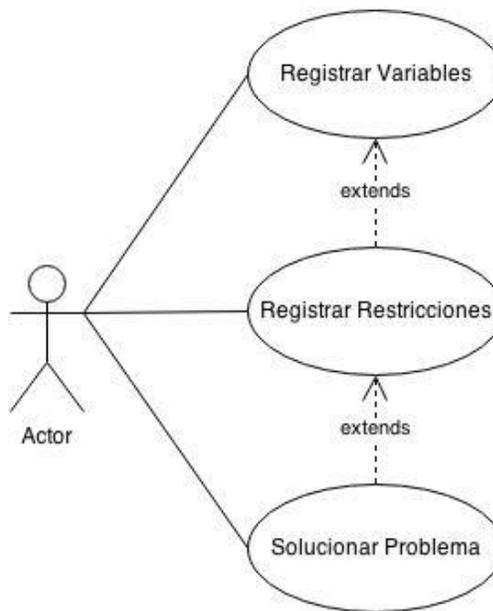
- Registrar variables y arreglos de variables de valores a procesar: El prototipo debe permitir al usuario registrar las variables que desee incluir en el problema de restricciones.
- Registrar restricciones sobre las variables: El prototipo debe permitir al usuario registrar restricciones de tipo menor que (<), mayor que (>), igual que (=), diferente que (!=) y todas las variables diferentes.
- Solucionar por medio del motor de búsqueda de restricciones: El prototipo debe solucionar el problema de restricciones según las variables y restricciones ingresadas.

- Generar todas las soluciones posibles: El prototipo debe generar todas las soluciones del problema y mostrarlas conforme el usuario las pida. (ver imágenes 2-12)

a. Fase Análisis y diseño

b. Modelo de análisis

Imagen 2. Modelo de análisis

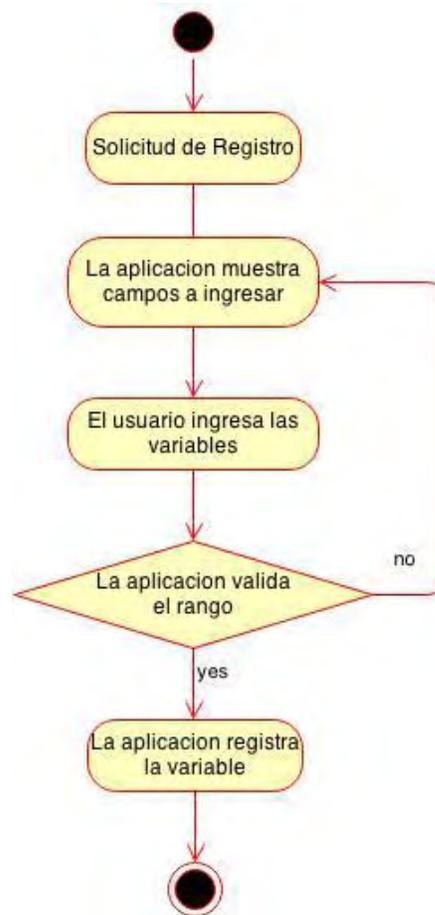


c. Modelo de diseño

d. Diagrama de actividades

e. Diagrama de actividades para Registrar Variables

Imagen 3. Diagrama de actividades Registrar Variables



f. Diagrama de actividades para Registrar Restricciones

Imagen 4. Diagrama de actividades Registrar Restricciones



g. Diagrama de actividades para Solucionar Problema

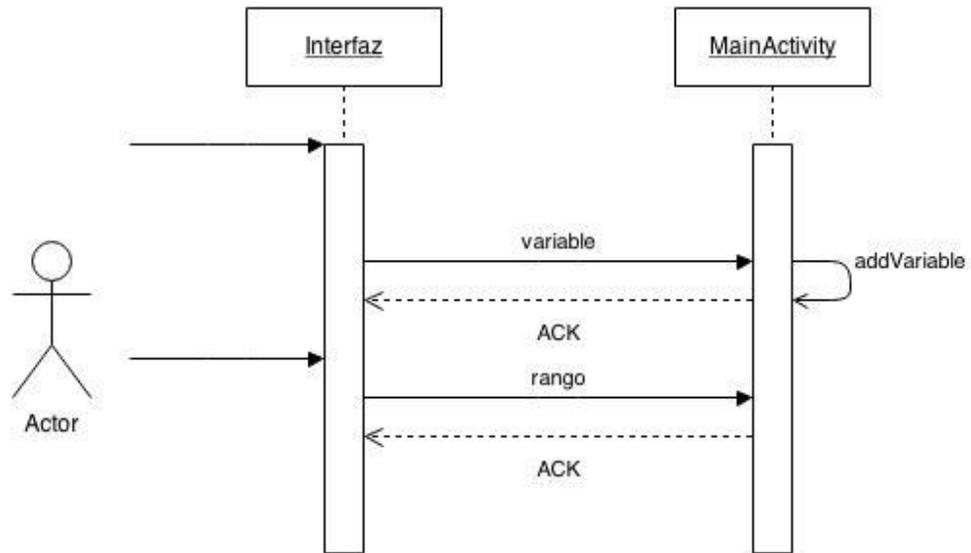
Imagen 5. Diagrama de actividades Solucionar Problema



h. Diagramas de secuencia

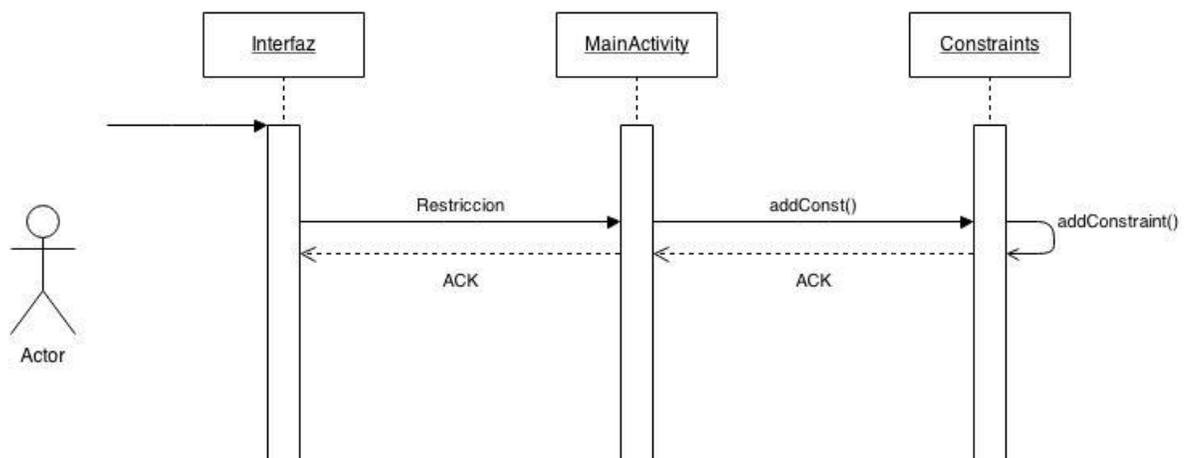
i. Diagramas de secuencia Registrar Variables

Imagen 6. Diagrama de secuencia para Registrar Variables



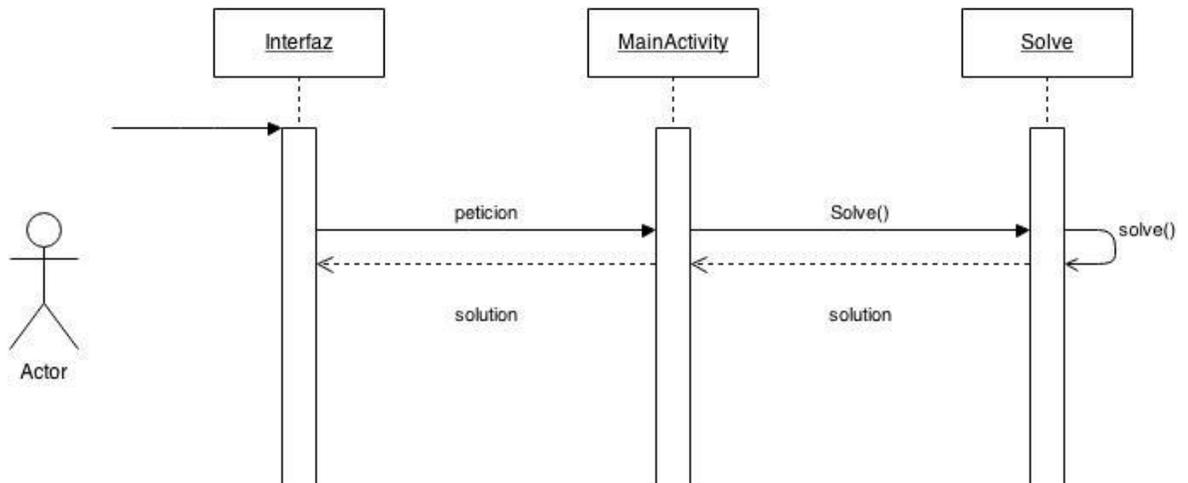
j. Diagramas de secuencia Registrar Restricciones

Imagen 7. Diagrama de secuencia para Registrar Restricciones



k. Diagramas de secuencia para Solucionar Problema

Imagen 8. Diagrama de secuencia para Solucionar Problema



I. Fase de implementación y despliegue

En esta fase se realizó la implementación del prototipo en donde los resultados son los pantallazos de la interfaz gráfica de usuario presentados a continuación:

Imagen 9. Interfaz de Usuario Registrar Variables

The image shows a user interface for registering variables. It has a dark header with three tabs: "Variables", "Restricciones", and "Solucion". The "Variables" tab is currently selected. Below the header, there are three input fields: "Nombre de la variable" (which is highlighted with a blue border), "Menor Valor", and "Mayor valor". At the bottom of the form, there are three buttons: "Add variable", "Add Array Variables", and "Show Status".

Imagen 10. Interfaz de Usuario Registrar Restricciones

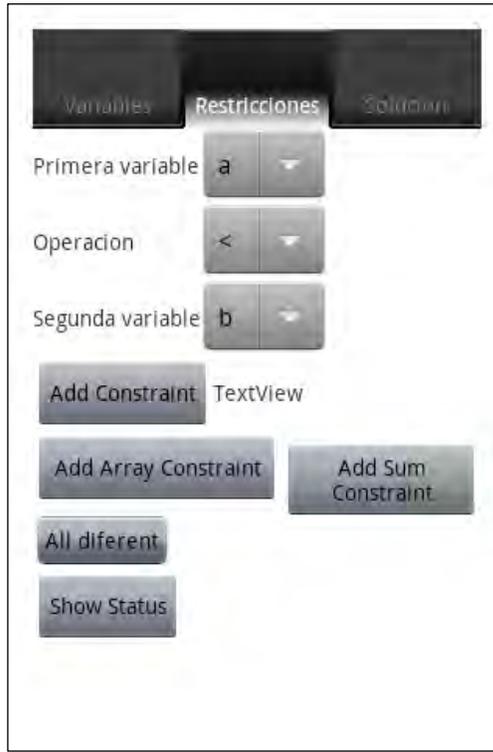
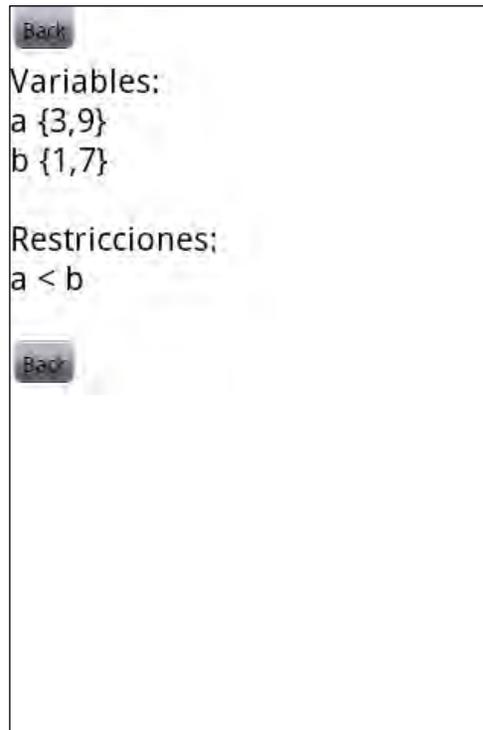


Imagen 11. Interfaz de Usuario Solucionar Problema



Imagen 12. Interfaz de Usuario ver Solución del Problema



El código fuente de la versión final del prototipo CONSTDROID se encuentra alojado en svn checkout <http://constdroid.googlecode.com/svn/trunk/constdroid-read-only> y puede ser trabajado con Subversion (<http://tortoisesvn.net/>).

El instalador para Android(.apk) se encuentra en el sitio web de la investigación <http://sonar.udenar.edu.co/Android-constraint-engine/>.

4.2.2. Versión portable Android Studio with Choco. En esta fase de la investigación se usa el entorno de programación Android Studio, un nuevo entorno de desarrollo integrado para el sistema operativo Android lanzado por Google, diseñado para ofrecer nuevas herramientas para el desarrollo de aplicaciones y alternativa al entorno Eclipse, hasta ahora el IDE más utilizado⁸.

Al crear un nuevo proyecto en Android Studio, la estructura del proyecto aparece con la mayoría de los archivos dentro del directorio SRC, posee un cambio al sistema de generación Gradle que proporcionará una mayor flexibilidad para el proceso de construcción, en el proyecto realizado se incorporaron las bibliotecas necesarias para programar con un sistema de restricciones de tal manera que se

⁸ En: <http://zeroturnaround.com/rebellabs/using-eclipse-for-java-development/1/>

generó una estructura general de los proyectos para que no sea necesaria implementarlas en futuros desarrollos de proyectos que necesiten restricciones, posteriormente se genera una versión portable del entorno para facilitar a los programadores el desarrollo de soluciones que usen un motor de búsqueda por restricciones.

La versión portable se denominó “Android Studio with Choco” publicada en el sitio Web de la investigación: <http://sonar.udenar.edu.co/Android-constraint-engine/?lang=es>, la versión base de Android Studio es *AndroidStudioBundle-133.970939-windows.exe* presente en el sitio web de los desarrolladores: <http://developer.Android.com/sdk/installing/Studio.html> además de la biblioteca principal de Choco: *Choco-solver-2.1.5.jar* presente igualmente en el sitio web de los desarrolladores: <http://www.emn.fr/z-info/Choco-solver/index.php?page=Choco-2>.

4.2.2. Acoplamiento con Gecode. En esta etapa de la investigación se usaron los archivos fuente presentes en el sitio web de Gecode: <http://www.Gecode.org/download/Gecode-4.2.1.tar.gz> , posteriormente se procede a compilar el archivo descargado para luego generar las bibliotecas necesarias para incluir en el NDK (Kit de herramientas para desarrollar aplicaciones en C/C++ sobre Android), el entorno desarrollo utilizado fue Eclipse con el plugin que permite el desarrollo de aplicaciones Android en su versión: *Android-ndk-r9d-windows* presente el sitio web: <http://dl.google.com/Android/ndk/Android-ndk-r9d-windows-x86.zip> .

Para la implementación de Gecode en el desarrollo de aplicaciones se encontraron varios inconvenientes que dificultaron el uso de la correcta inclusión de las bibliotecas ya que este motor de restricciones no es integrable con el sistema operativo Android como lo afirma en su publicación Christine Bouzant en el artículo del sitio de:

<http://article.gmane.org/gmane.comp.lib.Gecode.user/4129/match=ndk> al igual que Guido Tack (equipo de Gecode: <http://www.csse.monash.edu/~guidot/>) en donde coinciden en que la integración no se puede hacer ya que los hilos de Android (pthread API) son limitados en comparación con los hilos de un equipo tradicional de escritorio POSIX.

A pesar de ello se realizaron modificaciones en el código fuente de Gecode la cual permitía desactivar el manejo de hilos. Esta modificación se encontró en el foro de la página oficial de Gecode donde el tema central era la implementación de aplicaciones Android usando Gecode:

<http://article.gmane.org/gmane.comp.lib.Gecode.user/4129/match=ndk> esta se instaló en una distribución de Linux Debian, pero se presentan de igual manera los conflictos importar las bibliotecas al NDK.

4.3. ETAPA III: CREACIÓN DE SOLUCIONES

4.3.1. Sudoku. El Sudoku es un rompecabezas matemático de colocación que tiene como objetivo rellenar una cuadrícula de 9×9 celdas dividida en subcuadrículas de 3×3 con las cifras del 1 al 9 partiendo de algunos números ya dispuestos en algunas de las celdas. No se debe repetir ninguna cifra en una misma fila, columna o subcuadrículas.

Lo primero es llevar las consideraciones del sudoku a un modelo de problema de satisfacción por restricciones posteriormente se determina un sistema base para representar las variables que servirán para modelar el sudoku, en este caso será una matriz.

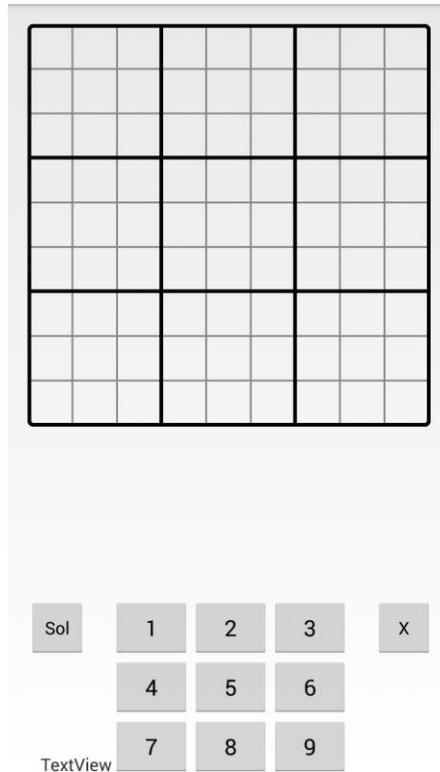
El siguiente paso es definir un dominio de valores que pueden tomar los campos del cuadro mágico, en este caso serían valores entre 1 y 9.

A continuación se debe definir que restricciones que rigen el modelo matemático en el sudoku, de esta manera para modelar el sudoku fueron definidas las siguientes:

- Los valores que pertenecen a una fila deben ser diferentes y estar entre 1 y 9, se debe cumplir para todas y cada una de las filas que pertenecen a la matriz del sudoku.
- Los valores que pertenecen a una columna deben ser diferentes y estar entre 1 y 9, se debe cumplir para todas y cada una de las columnas que pertenecen a la matriz del sudoku.
- Los valores que pertenecen a una subcuadrícula de 3×3 deben ser diferentes y estar entre 1 y 9, se debe cumplir para todas y cada una de las subcuadrícula que pertenecen a la matriz del sudoku.

Esta aplicación se desarrolló para dispositivos Android con 2 funcionalidades básicas. La primera es que te permite solucionar un Sudoku cuando están todas las casillas vacías, la segunda es agregar ciertos valores a algunas casillas del sudoku y solucionarlo a partir de este. La aplicación está en la capacidad de solucionar todos los posibles sudokus, además puede dar diferentes soluciones dependiendo de los valores base que se le ingresen o si se encuentra vacío, ya que toma una semilla aleatoria para general la solución. (ver imagen 13)

Imagen 13. Pantallazo Aplicación Sudoku



4.3.2. Magic Square. Es una tabla de grado n en la cual se dispone de una serie de números enteros en una matriz de forma tal que la suma de los números por columnas, filas y diagonales principales sea la misma. Además los números empleados para rellenar las casillas no deben de repetirse, de 1 a n^2 , siendo n el número de columnas y filas del cuadrado mágico.

Lo primero es llevar las consideraciones del cuadro mágico a un modelo de problema de satisfacción por restricciones:

Determinar un sistema base para representar las variables que servirán para modelar el cuadro mágico, en este caso será una matriz.

La primera es como obtener la suma de cada fila, columna, y diagonales principales y además garantizar que cada suma debe ser un valor específico de acuerdo al tamaño de la matriz.

M es la constante de la suma de cada fila, columna y diagonal se denomina la constante de magia o suma. Cada cuadrado mágico normal tiene una constante

única determinado únicamente por el valor de n y donde n es el tamaño de la matriz.

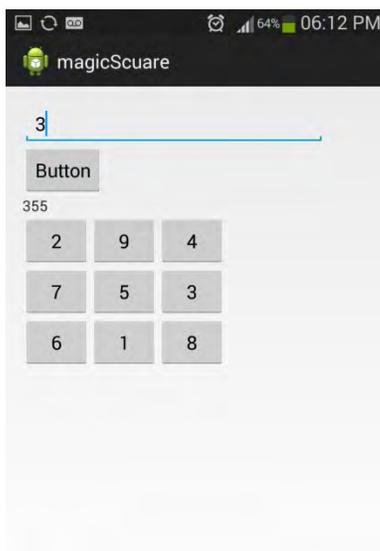
El siguiente paso es definir un dominio de valores que pueden tomar los campos del cuadro mágico, en este caso serían valores entre 1 y $n*n$.

A continuación se debe definir que restricciones rigen el modelo matemático en el cuadro mágico. Así pues para modelar el cuadro mágico se definieron las siguientes restricciones:

- Las sumas de los valores que pertenecen a una fila deben ser iguales a M para todas y cada una de las filas que pertenecen a la matriz del cuadro mágico.
- Las sumas de los valores que pertenecen a una columna deben ser iguales a M para todas y cada una de las columnas que pertenecen a la matriz del cuadro mágico.
- Las sumas de los valores que pertenecen a la diagonal principal de la matriz del cuadro mágico deben ser iguales a M .
- Las sumas de los valores que pertenecen a la diagonal secundaria de la matriz del cuadro mágico deben ser iguales a M .
- Todos los valores que pertenecen a la matriz del cuadro mágico deben ser diferentes.

Esta aplicación se desarrolló para dispositivos Android con una funcionalidad básica que permite ingresar un valor numérico entero, el cual será el tamaño de la matriz, y permite dar solución al cuadro mágico de acuerdo al tamaño de la matriz, en las pruebas a la aplicación se determinaron dos limitantes, la primera es que si solamente se coloca las restricciones para obtener la solución, se pueden solucionar cuadros mágicos de hasta 7×7 (filas x columnas). Si además de las restricciones se agregó una semilla aleatoria para poder solucionar cuadros mágicos de hasta 18×18 (filas x columnas). (ver imagen 14)

Imagen 14. Pantallazo Aplicación Magic Square



4.4. ETAPA IV: PRUEBAS

Se realizaron las pruebas de rendimiento en donde se capturaba el tiempo de respuesta en cada solución y bajo diferentes dispositivos Android con distintas características de hardware y versión del Sistema Operativo. (ver cuadro 5-7)

Cuadro 4. Dispositivos Android y características

No. Unidad	Solución	Características
1	Samsung Galaxy S3	Android 4.3, Procesador ARM Quad Core 1.4 Ghz, RAM 831Mb
2	Sony XPeria Mini Pro	Android 2.3., Procesador Dual Core 1.3 Mhz, RAM 256 Mb
3	Huawey Y511	Android 4.2.2, Procesador Dual Core 1.3 Mhz, RAM 512 Mb
4	Samsung Galaxy Duos	Android 2.3, Procesador Dual Core 832 Mhz, RAM 512 Mb
5	Acer V5	Windows 8, Procesador Intel Core i5 3th G. 1. Ghz, RAM 4Gb

Cuadro 5. Tiempos de Respuesta en Magic Square

No. Unidad	Dispositivo	Tiempo(ms)		
		3x3	5x5	7x7
1	Samsung Galaxy S3	.279	1587	55800
2	Sony XPeria Mini Pro	200	3108	115361
3	Huawey Y511	393	1934	83884
4	Samsung Galaxy Duos	353	2812	108292
5	Acer V5	853	1986	6423

Cuadro 6. Tiempos de Respuesta en Sudoku

No. Unidad	Dispositivo	Tiempo(ms)	
		Vacía	Plantilla
1	Samsung Galaxy S3	1040	664
2	Huawey Y511	899	385
3	Samsung Galaxy Duos	1229	688
4	Acer V5	465	301

Cabe aclarar que la casilla denominada “Vacía” significa que el motor realizo el ingreso de todos los valores de la cuadrícula del Sudoku; además el campo “Plantilla” se tomó una cuadrícula con algunos campos ingresados para que el motor complete el Sudoku, la plantilla fue la siguiente: (ver imágenes 15-20)

Imagen 15. Plantilla Sudoku

	5	2		8		7	6	
8		9		7		5	4	
					9	2	8	
2						8		
		5	3		8	4		
4				9	5	6		
9	2	7				3		
5		8						
		3		2				

Imagen 16. Relación entre Dispositivos y tiempos de respuesta para Magic Square 3x3

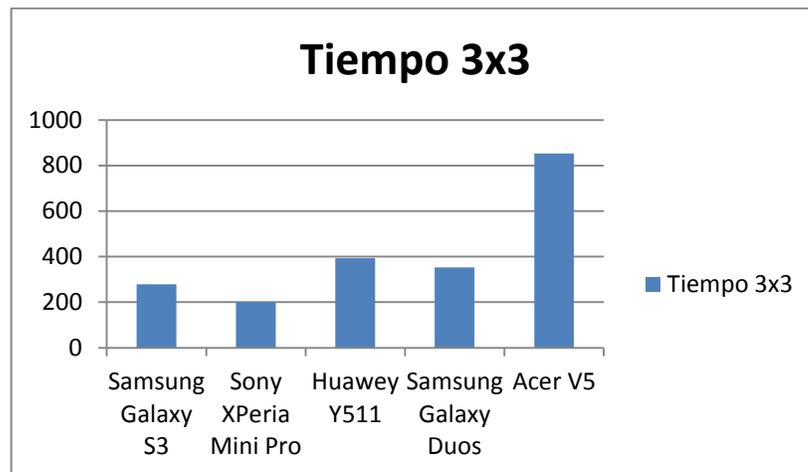


Imagen 17. Relación entre Dispositivos y tiempos de respuesta para Magic Square 5x5

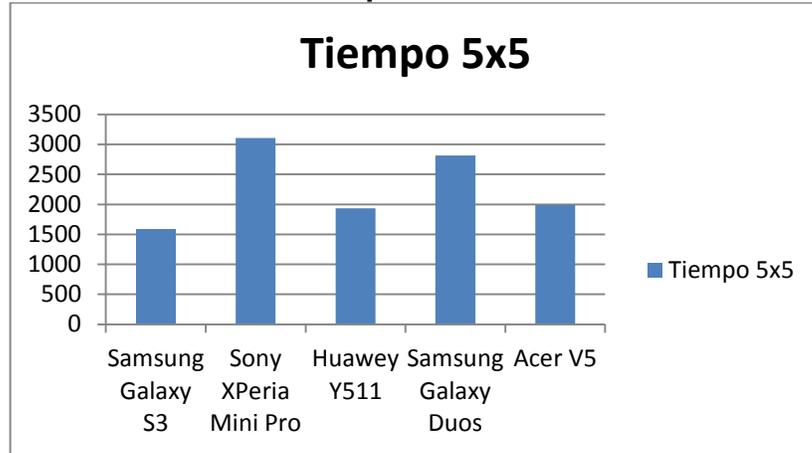


Imagen 18. Relación entre Dispositivos y tiempos de respuesta para Magic Square 7x7

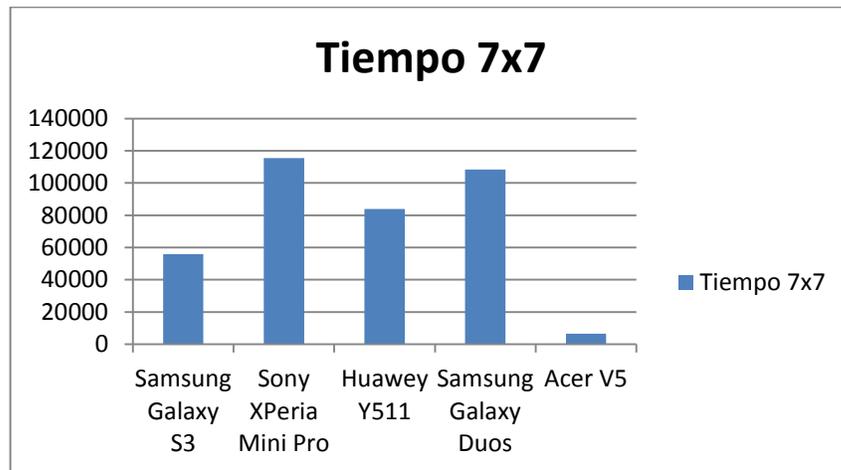


Imagen 19. Relación entre Dispositivos y tiempos de respuesta para Sudoku Vacío

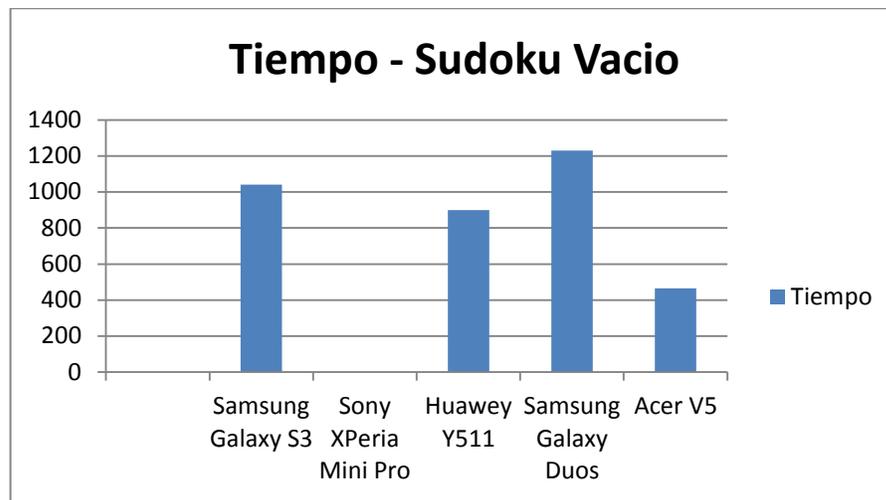
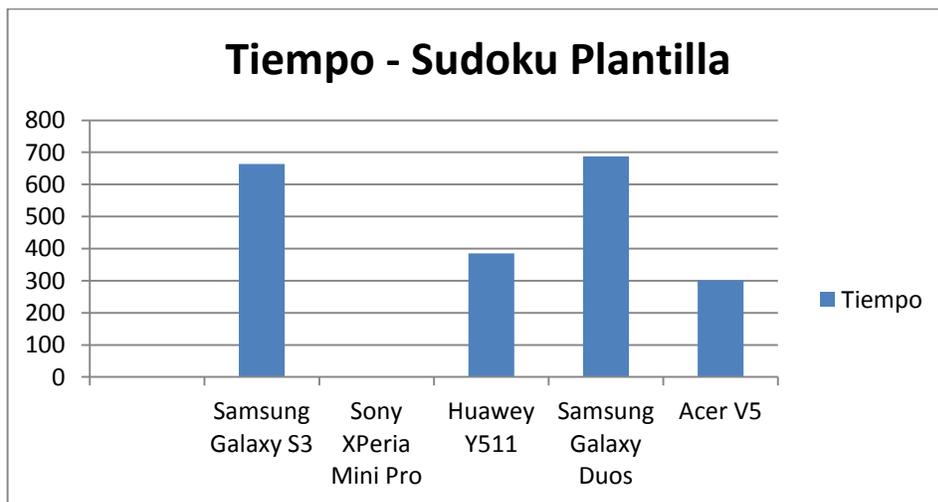


Imagen 20. Relación entre Dispositivos y tiempos de respuesta para Sudoku con Plantilla



4.5. ETAPA V: PUBLICACIÓN

En esta etapa del proceso se realiza un artículo científico con normas IEEE y se presenta como candidato a la categoría de Informática y Sociedad en la Conferencia Latinoamericana en Informática (CLEI 2014) a realizarse en Montevideo, Uruguay del 15 al 19 de Septiembre de 2014.

El artículo se encuentra el sitio web de la investigación:
<http://sonar.udenar.edu.co/Android-constraint-engine/?lang=es>.

5. ANÁLISIS DE RESULTADOS

Teniendo en cuenta los resultados de cada etapa de esta investigación, se puede afirmar que existe una diferencia alta en los tiempos de respuesta entre dispositivos móviles a las de un equipo portátil de gama alta como se observa en las figuras presentadas.

A pesar de que las soluciones desarrolladas son de complejidad alta todos los dispositivos lograron ejecutar correctamente la aplicación propuesta en cada caso.

6. CONCLUSIONES

A partir del proceso de investigación realizado se logró implementar un prototipo de sistema de restricciones para dispositivos móviles bajo sistema operativo Android nombrado CONSTDROID que permite solucionar juegos combinatorios y problemas de tipo no-polinomiales incorporando funcionalidades del motor de restricciones de Choco Java bajo el lenguaje Java, además se generó una versión portable del entorno de programación sobre Android en la cual se integran las bibliotecas de Choco para los desarrolladores interesados en construir aplicaciones con restricciones bajo Android.

El eStudio también permitió realizar soluciones en las que se trabaja con un motor de programación por restricciones acoplada en el sistema operativo Android como lo son Sudoku y Magic Square.

Se realizó el artículo “Uso de un motor de restricciones sobre Android” el cual se propuso en un evento internacional.

Por lo anterior se puede concluir que es posible el uso de un motor de restricciones sobre dispositivos Android, se obtienen los mismos resultados pero en diferentes tiempos de respuesta dependiendo de las características de hardware que tenga cada dispositivo ya que todos los dispositivos lograron ejecutar las aplicaciones desarrolladas.

De igual manera otro punto a concluir según los resultados analizados es que para aplicaciones de cantidad elevada de variables es recomendable en cuanto eficiencia usar las aplicaciones sobre equipos de escritorio o portátiles pero para cantidades pequeñas de variables es más óptimo y eficiente usar las aplicaciones sobre dispositivos móviles, esto permite afirmar que para el desarrollo de aplicaciones de tareas cotidianas y/o juegos combinatorios es conveniente desarrollarlas sobre dispositivos móviles para que sean más óptimos.

7. RECOMENDACIONES

Realizar la misma investigación pero con un motor de restricciones basado en C Gecode e igualmente realizar publicaciones que aporten al conocimiento de lenguajes de programación.

Verificar los resultados de esta investigación ya constituyen un aporte al conocimiento de la programación por restricciones y al desarrollo de aplicaciones Android y proporcionó información útil acerca del rendimiento de los dispositivos usados.

BIBLIOGRAFÍA

CORMEN, T., LEISERSON, C., RIVEST, R. & STEIN, C. (2009). Introduction to Algorithms, The MIT Press. USA.

APT, K., (2003). Principles of Constraint Programming. Cambridge. UK.

SHULTE, C., TACK G. & Lagerkvist, M. (2013) Modeling and Programming with Gecode. <http://www.Gecode.org/doc-latest/MPG.pdf>

LABURTHE, F., JUSSIEN, N. (2011). Choco Solver - Documentation. <http://Choco.mines-nantes.fr/>

LABURTHE, F., JUSSIEN, N. (2011). Choco Solver - Tutorial. <http://Choco.mines-nantes.fr/>

The C++ Resources Network (2013). www.cplusplus.com/

Java™ Platform, Standard Edition (2013). <http://docs.oracle.com/javase/7/docs>