FACULTAD DE INGENIERÍA

DEPARTAMENTO DE COMPUTACIÓN E INFORMÁTICA

CARRERA DE INGENIERÍA INFORMÁTICA MENCIÓN EN GESTIÓN DE LA INFORMACIÓN

IMPLEMENTACIÓN DE SISTEMA ESTEGANOGRAFÍA LSB UTILIZANDO CHOCO SOLVER

Profesor Guía: Franklin Johnson

Valparaiso, Chile

2016

Identificación

Tesis	Implementación de sistema Esteganografía LSB utilizando Choco Solver
Alumno/A	Alvaro Alonso Arias Pizarro
Institución	Universidad de Playa Ancha
Carrera/Programa	Ingeniería Informática Mención en Gestión de la Información
Año/Fecha	2016
Profesor Guía	Franklin Johnson

1. Resumen ejecutivo ante-proyecto

El uso de internet en la actualidad es cada vez mayor y el acceso a sistemas de informacion tambien lo es, la manipulación de esta información puede estar en riesgo, es por eso que se debe tener en cuenta que información ya sea de empresas, usuarios entre otros debe ser protegida para poder tener un control sobre el acceso al sistema y los derechos de usuarios al sistema de información.

Existen diversos métodos de seguridad informática para abordar diferentes problemas, este proyecto se enfocara en la esteganografía, la cual es considerada como el arte y la ciencia de la comunicación invisible, esta permite ocultar información (mensajes), dentro de otro objeto (imagen), de modo que no se perciba su existencia [1].

Como resultado a este trabajo se desarrollara una aplicación en Java, que consta de una imagen con un mensaje incrustado en ella a través esteganografía LSB, la cual nos permite ocultar información en el byte menos significativo de cada uno de los pixeles de una imagen [2].

2. Descripción General

2.1. Descripción del Problema

En la actualidad es muy común el compartir información por internet sin analizar en profundidad qué tan seguro es el enviar un archivo a otra persona a través de la web, de hecho si queremos mandar un código o un mensaje secreto con la intención de que solo el destinatario pueda comprenderlo, habitualmente se recurre a entregar dicho mensaje en persona.

Una forma de abordar este problema es mediante Esteganografía, proceso que se encarga de ocultar un mensaje secreto dentro de otro objeto, de tal manera que alguien no pueda conocer la presencia del mensaje oculto. La estructura básica de la Esteganografía está compuesta por el portador, el mensaje y la llave [3].

Este proyecto se centra en el método LSB (Bit menos significativo), junto con otras herramientas como programación con restricciones y cuadrados latinos más conocidos como Sudokus, que facilitan la comprensión del tema.

La programación con restricciones trata de resolver problemas mediante la declaración de restricciones sobre el área del problema, puede dividirse en dos ramas, una es la satisfacción de restricciones que abarca los problemas de dominios finitos, la cual será utilizada en este proyecto, mientras que la resolución de restricciones esta orientada a problemas de dominios infinitos [5].

Dentro de la resolución de problema de satisfacción de restricciones (CSP) se tienen dos fases:

- Modelar, que es expresar el problema mediante una sintaxis de CSP , es decir mediante un conjunto de variables, dominios y restricciones.
- Procesar el problema, para esto hay dos maneras:
    - Técnicas de consistencia, las cuales buscan la resolución basándose en la eliminación de valores inconsistentes de los dominios de las variables.
    - Algoritmos de búsqueda, estos se basan en la exploración sistemática del espacio de soluciones hasta encontrar una solución o probar que no existe tal solución [5].

Para desarrollar el proyecto y dar un robustez a la hora de resolver este tipo de programación, se utiliza la herramienta Choco Solver, que es una librería de Java, especializada en el área de resolver problemas de programación con restricciones.

Teniendo mas claro algunas de las herramientas, técnicas que se utilizan en el proyecto, se debe comentar que se desea resolver con todo esto.

Los sudokus son una matriz cuadrada de 9x9 subdividida en nueve sub matrices de 3x3 llamadas cajas, en las celdas de la matriz se encuentran números del conjunto D = {1,2,. . . .,9}, de tal forma que cada fila, cada columna y cada caja contiene uno y solo uno de los elementos del conjunto D [7].

Este Sudoku se pretende resolver gracias a la programación con restricciones, pero el primer desafío que se tiene es generar Sudokus desde cero, para posteriormente ser resueltos con algún algoritmo de búsqueda. El primer paso a seguir es modelar el problema de sudoku, describiendo sus variables, dominios y restricciones con claridad para luego aplicar algún tipo de algoritmo a resolver.

Para formalizar el problema de Sudokus se consideran como tableros de 9 x 9 con 81 celdas. Cada una de estas celdas corresponde con una variable del CPS.

$$V = V_{ji}, i, j\{1, 2, 3, \ldots., 9\}$$

Los índices de cada variable corresponde con el numero de fila y el numero de columna respectivamente. Por lo tanto, realizando el producto cartesiano de los índices de cada fila y columna se obtiene:

$$V = \{V11, V12, V13, V14, \ldots\ldots, V98, V99\}$$

Cada celda del tablero puede contener un valor entre 1 y 9. Esto significa que el dominio del problema son los 9 primero números naturales.

$$D = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

También las celdas pueden contener un valor de serie pre definido que no se puede modificar. Esto restringe el dominio de las celdas predefinidas a un único valor, que coincide con el valor predefinido. Al establece esos valores predefinidos, se considera que cada variable tiene un dominio propio no compartido con el resto de variable. Por lo tanto, existirá un conjunto de 81 dominios, y cada uno de ellos será un subconjunto del dominio D [6].

$$D' = \{D_{11}, D_{12}, D_{13}, D_{14}, \ldots., D_{98}, D_{99}\}/D_{ij} \subseteq D, \forall i, j \in \{1, 2, 3, \ldots, 9\}$$

- Restricción de fila: dos celdas de la misma fila no pueden tener el mismo valor asignado.

$$\neg\exists k \in D / V_{ij} = V_{ik} \wedge j \neq k, \forall i, j \in D$$

- Restricción de columna: dos celdas de la misma columna ni puede tener el mismo valor asignado.

$$\neg\exists k \in D / V_{ij} = V_{kj} \wedge i \neq k, \forall i, j \in D$$

- Restricción de bloque: dos casillas del mismo bloque no pueden tener el mismo valor de asignado.

$$\neg\exists k, k' \in D / V_{ij}, V_{kk'} \wedge mismobloque\ (V_{ij}, V_{kk'}), \forall i, j \in D$$

$$mismobloque\ (V_{ij}, V_{kk'}) \equiv (i\ div\ 3 = k\ div\ 3) \wedge (j\ div\ 3 = k'\ div\ 3)$$

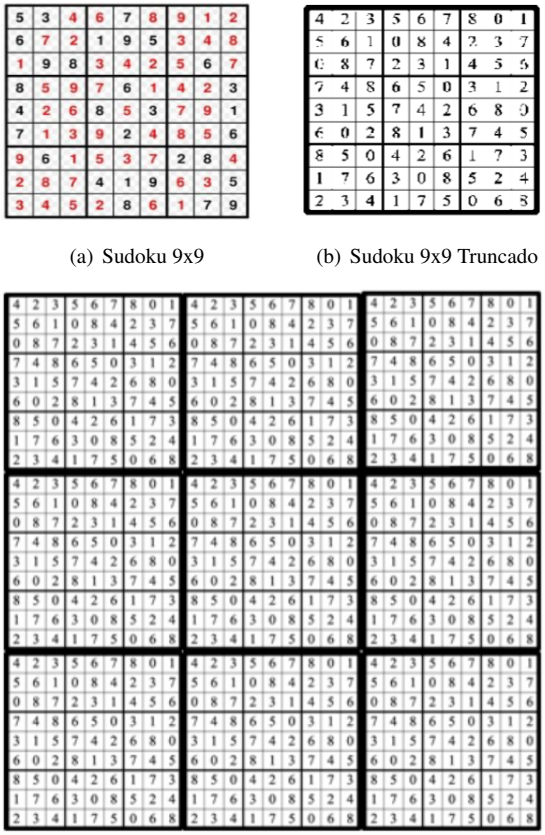
Se logra tener un total de 1944 restricciones dentro del conjunto de restricciones C. Cada una de las 81 variables tiene una restricciones con las 8 variables restantes de su fila, otra restricción con las 8 variables restantes de su columna, y otra restricción con las 8 variables restantes de su bloque. Es decir que cada variable tiene un total de 24 restricción con otras variables. Por lo tanto 81 x 24 = 1944 restricciones en el problema [6].

El método propuesto para poder generar los sudokus es el Algoritmo de Fuerza Bruta, el cual genera una matriz 9x9 y asigna valores del conjunto D aleatoriamente, después comprueba si la matriz resultante es un Sudoku, para poder comprobar se utiliza el Algoritmo de Marcha atrás, el cual consiste en asignar un valor aleatorio a una celda vacía, verificando que dicha asignación cumpla con las condiciones de Sudoku, de lo contrario, devuelve el Sudoku inicial original e intenta con un valor distinto. Al momento de encontrar la solución al Sudoku, se procede a guardar la matriz inicial de 9 x 9 generada por el Algoritmo de Fuerza Bruta y los valores obtenidos por el Algoritmo de Marcha atrás [7].

Método LSB consiste en ocultar información en el bit menos significativo de cada uno de los pixeles de una imagen, consiguiendo así que el cambio sea invisible al ojo humano. Las imágenes digitales pueden ser de 8 bits o 24 bits, en nuestro caso trabaja sobre las de 24 bits que nos permiten incrustar 3 bits de información en cada pixel [4].

Ahora teniendo en cuenta que se trabaja sobre imágenes de 24 bits, se tiene que generar un el sudoku que sirve de guía para poder incrustar información en la imagen, para esto se toma el sudoku bien generado

de 9 x 9, el cual es resuelto obteniendo una matriz m, la que es truncada para ser compatible con los valores que se desean incrustar y como ultimo paso aumentada a una matriz 27 x 27.



(a) Sudoku 9x9 (b) Sudoku 9x9 Truncado

(c) Sudoku 27x27

Figura 1: Sudokus

Para realizar la incrustación de datos, se debe corroborar que el mensaje este encriptado por seguridad, esto se logra traspasando cada letra del mensaje a código ASCII que posteriormente es convertido a base-9, este valor obtenido cae en el rango de [0-8] que consta de 3 dígitos [1], un pequeño ejemplo seria pasar la letra "A" a ASCII quedando como el numero 65, el cual será convertido a base-9 obteniendo el valor 027.

La forma de llegar al pixel que se desea modificar parte con tomar una ubicación aleatoria, luego encontrar valores regidos por las siguiente formula.

$$X = (R\,\%6) + 6$$

$$Y = (G\,\%6) + 6$$

Donde R (rojo) y G (verde) son componentes de ese píxel, los valores (X, Y) sirven en la matriz de 27x27 como ubicación siendo el valor de X designado para las filas y el valor de Y para la columnas, la intersección entre estas dos nos genera un punto central de ubicación para empezar a trabajar en la búsqueda del valor que menos distorsione la imagen.

Si nos guiamos en el ejemplo anterior de la letra "A" que corresponde a 027 en base-9 nuestro 0 representa a la R, el 2 a la G y el 7 a la B de los colores del pixel RGB, es así ubicándonos en el centro formado por la intercesión de X e Y, empezamos a buscar el valor 0 que mas cerca se encuentra en las direcciones arriba, derecha y en el bloque de 3x3, este proceso se realiza para cada número de nuestra letra "A", siempre se deben elegir los que tenga una menor distancia al centro, ya que estos son los pixeles que generan una menor distorsión [1].

2.2. Objetivo General

Desarrollar una aplicación, la cual permite incrustar texto en una imagen, de modo que la imagen no se vea alterada a la percepción del ojo humano.

2.3. Objetivo Específicos

- Generar y resolver sudokus bien definidos utilizando programación con restricciones a través de la librería Choco Solver.
- Aplicar esteganografía LSB para ocultar texto en una imagen.

2.4. Resultados y Productos Esperados

Se busca desarrollar una aplicación en java, la cual permite incrustar texto en una imagen utilizando Esteganografía LSB apoyado por la librería Choco Solver. Para lograr el desarrollo de la aplicación se utilizará el IDE Eclipse para escribir el código java.

3. Antecedentes y Justificación

3.1. Antecedentes

Este proyecto se desarrolla en diferentes áreas que se relacionan para alcanzar la creación de la aplicación, herramientas (Choco Solver), paradigma (Programación con Restricciones), técnicas (Esteganografía LSB) entre otras.

La programación con restricciones es un paradigma fascinante, entre los años 60 y 70 se ven los primeros trabajos en el campo de la Inteligencia Artificial, tiene un fuerte lazo con la investigación de operaciones y la programación lógica.

Investigando mas a fondo sobre trabajos relacionados con la programación con restricciones nos encontramos con la investigación de Rodrigo Ronald Gumucio Escobar donde expone de una forma, si bien no completamente detallada, aporta una idea de cómo se deben plantear los puzles Sudokus en forma de modelo de restricciones, para así poder aplicar diferentes técnicas de la programación con restricciones [8].

Para entregar robustez al desarrollo de la aplicación, se debe elegir una librería de apoyo para resolver de manera eficiente los problemas de restricciones, para este proyecto se toma como referencia Choco Solver, ya que es una librería open source para java, a su ves tiene distinciones a nivel mundial como una de los mejores complementos para resolver problemas con restricciones.

Realizando una búsqueda de trabajos relacionados con resolver sudokus modelados a base de la programación con restricciones gracias a esta librería , para así posteriormente implementar la Esteganografía LSB, no se encuentra nada en ingles, tampoco en español, es por el motivo que se pretende profundizar en esta área y entregar un aporte considerable a la investigación en habla hispana.

Si bien no se encontró trabajos relacionados directamente, si se utilizo documentación para poder profundizar en el tema de Choco Solver obtenida de su pagina oficial www.choco-solver.org.

Los Sudokus son un pilar fundamental en la investigación para que esta se pueda llevar a acabo, entender como aparecieron nos remonta a épocas del Siglo XII donde se conocían como cuadrados mágicos, siendo el sudoku un caso especial de los cuadrados latinos, ya que impone una restricción adicional, la que dice que los subgrupos de 3*3 deben contener también los dígitos del 1 al 9. A fines de la década de los 70 se publicó por primera vez este tipo de rompecabezas en la revista “Math Puzzles and Logic Problems”, lo

que en años después tras añadir algunas mejoras y reduciendo las pistas dadas se convirtió en uno de los rompecabezas más vendidos en Japón [9].

Estos Sudukos son el nexo que se usa entre la programacion con restricciones y la estenografia como se nombra en la investiagacion de Leona Antony, Harlay Maria Mathew los sudokus son la matriz de refencia para implementar la esteganografia LSB [1].

Uno de los punto mas interesantes de la investigacion es la esteganografia la cual habre un nuevo mundo de posibilidades en ambitos como la seguridad de la informacion. No hay que confundir la esteganografia con la criptografia que si bien trabajan en cunjunto de una manera impecable, ya que si se aplican estas das tecnicas se da una solides a la informacion que se desea proteger [1].

Pero entrando a mas detalles la esteganografia es un arte de ocultar informacion en un objeto que pase desapercibido. Para poder entender bien esta tecnica se centro la investigacion en el metodo LSB (el bit menos significativo), investiagaciones que son importantes en el desarrollo como lo es la de Hngfu Yang, Xingming Sun, Guan Sun donde dan a conocer como debe quedar una imagen interbenida como canal seguro para mandar informacion importante, ya que esta no debe parecer modificada ante la percepcion del ojo humano [2].

En esta misma investigacion se entrega el metodo de cómo encontrar el bit menos significado y asi poder intervenir en el con el mensaje que se desea inscrustar, como tambien explican que el uso de la criptografia para usar como medio de seguriddad del mensaje es una tecnica fundamental a aplicar, para entregar la mayor seguridad posible [2].

3.2. Justificación del ante-proyecto

Este proyecto es necesario para aportar mas información sobre la Esteganografía LSB, Programación con Restricciones, Choco Solver y Sudokus en habla hispana, con el objetivo de lograr ser un pilar en futuras investigaciones que se enfoquen en perfeccionar la herramienta para grandes desarrollos en el área de seguridad, considerando un aspecto importante el tener una herramienta que entregue seguridad al enviar información valiosa entre múltiples destinatarios.

Como primera etapa se realiza un prototipo de aplicación, el cual sea eficiente al momento de ocultar información valiosa en imágenes, con esto se obtiene un par de paper donde se detalla mas a fondo las diferentes aristas del trabajo, por ejemplo como se resolvió el desarrollo de los sudokus bien definidos gracias a la programación con restricciones, así también aportar en la aplicación de Esteganografía LSB para ocultar información en imágenes y el uso de las librerías Choco Solver.

4. Planificación y Métodos

4.1. Requerimientos y necesidades

- Usuario para testear los resultados de la aplicación y verificar su buen funcionamiento, en este caso sera el profesor guía.

4.2. Metodología

Dentro de este proyecto se trabajará en dos partes la primera es el desarrollo del anteproyecto donde se aplicó metodología de reuniones, estas se realizaron de forma semanal, enviando previamente los avances para que el día de las reuniones se conversara y corrigieron errores, para seguir avanzando.

La segunda parta del proyecto, que es el desarrollo de la aplicación se guía en una metodología de espiral evolutiva, la cual se acomoda al desarrollo de este software.

La metodología Espiral evolutiva se basa en una serie de ciclos los cuales se repiten en forma de espiral, cada ciclo tiene cuatro etapas que se deben cumplir para pasar al otro nivel del espiral, esta orientado a evitar riesgos de trabajo. Cada vez que se avanza un ciclo se va alcanzando un nivel superior hasta concluir el proyecto.

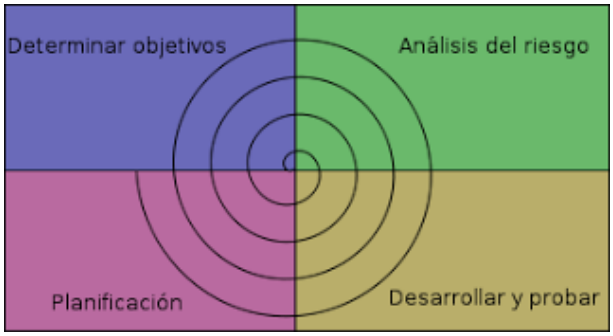


Figura 2: Modelo Espiral

Para cada cilco habrá cuatro etapas:

1. Planificación: Determinar o fijar objetivos.
    - Fijar también los productos definidos a obtener: requerimientos, especificación, manual de usuario.
    - Fijar las restricciones.

- Identificación de riesgos del proyecto y estrategias alternativas para evitarlos.
- Hay una cosa que solo se hace una vez: planificación inicial o previa.

2. Análisis del riesgo.

   - Se estudian todos los riesgos potenciales y se seleccionan una o varias alternativas propuestas para reducir o eliminar los riesgos.

3. Ingeniería: Desarrollar, verificar y validar (probar).

   - Tareas de la actividad propia, desarrollo y pruebas.
   - Análisis de alternativas e identificación resolución de riesgos.

4. Evaluación del Cliente.

   - Revisamos todo lo hecho, evaluándolo, y con ello decidimos si continuamos con las fases siguientes. Planificamos la próxima actividad en case de ser necesaria una nueva iteración.

4.3. Planificación de actividades

A continuación se presenta la carta Gantt que cubrirá el proyecto de título I y II.

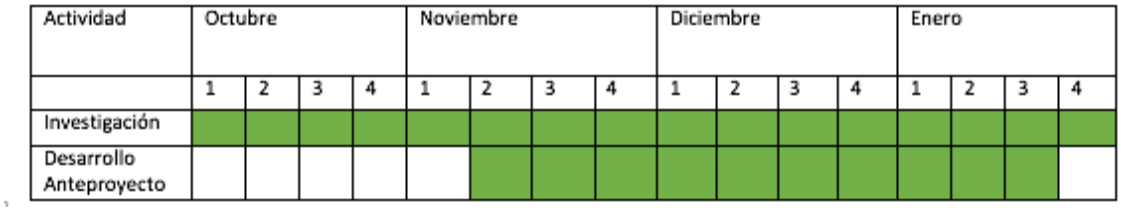


Figura 3: Carta Gantt Desarrollo Anteproyecto

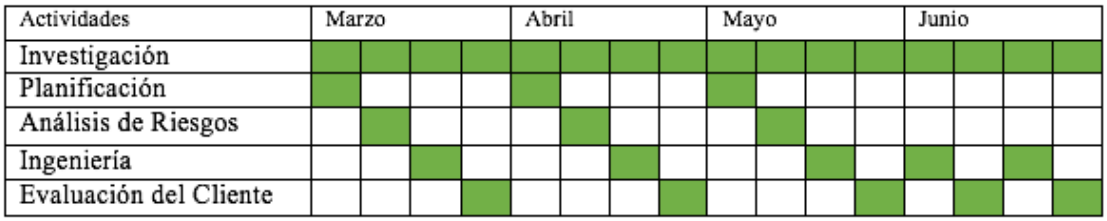


Figura 4: Carta Gantt Desarrollo Proyecto Titulo II

5. Referencia

[1] P.JAYAKUMAR LEONA ANTONY, HARLAY MARIA MATHEW. A new steganographic approach using sudoku with digital signature. International Journal of Computer Engineering Technology (IJCET), 5:177–185, 2014.

[2] Hengfu Yang, Xingming Sun, and Guang Sun. A high-capacity image data hiding scheme using adaptive lsb substitution. Radio Eng, 18(4):509, 2009.

[3] Nick Nabavian. Cpsc 350 data structures: Image steganography. nabavl00@ chapman. edu, 2007.

[4] Kshetrimayum Jenita Devi. A Sesure Image Steganography Using LSB Technique and Pseudo Random Encoding Technique. PhD thesis, National Institute of Technology-Rourkela, 2013

[5] Miguel Angel Salido Gregorio and Federico Barber Sanchís. Introducción a la programación de restric- ciones. Inteligencia Artificial: Revista Iberoamericana de Inteligencia Artificial, 7(20):13–30, 2003.

[6] Cristian Aguilera Martinez. Fundamentos de sudoku. upe edu, 2009.

[7] Armando Cabrera Pacheco, Edwin Meneses Rodrıguez, and Roger Pacheco Castro. Un algoritmo sencillo de sudokus de 9 x 9.

[8] Rodrigo Ronald Gumucio Escobar. Programando con restricciones. Acta Nova, 5:72, 2011.

[9] Rocio Casco Muga Daniel de la Casa Riballo. Historia sudoku. Universidad de Madrid, 2011.