

Universidad Autónoma Metropolitana
Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería
Ingeniería en Computación

Algoritmos para plegado de mapas cuadrados

Alumno:
Ramos León Anthony Damián
207200255

Trimestre 11-O

2 de diciembre del 2011

Segunda versión

Asesores:

Dr. Francisco Javier Zaragoza Martínez
Departamento de Sistemas
Profesor Titular

M. en C. Cynthia A. Rodríguez Villalobos
Departamento de Sistemas
Profesora Asociada

OBJETIVOS

OBJETIVO GENERAL

Implementar un sistema que busque las diferentes formas en que se puede doblar una hoja con dobleces ortogonales, que decida si es posible hacer un replegado y que muestre una visualización.

OBJETIVOS PARTICULARES

- I. Diseñar e implementar un módulo que busque todas las diferentes formas de plegado ortogonal de un mapa.
- II. Diseñar e implementar un módulo que decida si cierto replegado es posible.
- III. Diseñar e implementar un módulo de visualización de los resultados.

Introducción

El plegado y replegado es un tema apasionante de la geometría computacional. Hay una amplia gama de problemas de plegado y replegado, algunos que se remontan varios siglos atrás y todavía no se resuelven, como el plegado de poliedros convexos, mientras que otros son más recientes, como el plegado de proteínas. En los últimos años ha habido un tremendo progreso en muchos de los problemas fundamentales de plegado y replegado, sin embargo, algunas de las preguntas más importantes quedan pendientes. Esto deja el tema en un estado interesante. Los distintos problemas de plegado son atractivos desde el punto de vista puramente matemático, por la belleza de los mismos problemas.

Sin embargo, la mayoría de estos problemas tienen estrechas relaciones con importantes disciplinas de la matemática, teoría de la computación y hasta el arte del origami [1]. En el área de la matemática está la matemática combinatoria, la teoría de grafos, la geometría y hasta la teoría de números. Por parte de la teoría de la computación está el análisis y diseño de algoritmos, la geometría computacional y las estructuras de datos.

Este proyecto terminal se basará únicamente en un problema llamado **plegado plano**. En la matemática combinatoria el problema de **plegado plano** es la cuestión de calcular de cuántas maneras se puede doblar un mapa rectangular a lo largo de sus pliegues. Este problema fue propuesto hace varios años por Stanislaw M. Ulam [2]. Por lo tanto el objetivo de este proyecto es diseñar e implementar algoritmos que busquen las diferentes formas que puede adoptar una hoja con plegados ortogonales, así como también decidir si es posible un plegado dado y mostrar una visualización.

Justificación

Nuestro problema está relacionado con algunos problemas clásicos de combinatoria de plegado de mapas. En la mayoría de estos problemas, se quiere contar de cuántas formas distintas se puede plegar un mapa a una figura plana.

Dos pliegues suelen ser considerados "diferentes" en este contexto, si difieren en el orden total de las caras en el plegamiento. Estos problemas han sido estudiados extensamente, en particular para los mapas de dimensión $1 \times n$, pero siguen en gran medida sin resolverse.

Por ejemplo, hay seis maneras de plegar una tira de tres segmentos, como muestra la Figura 1.

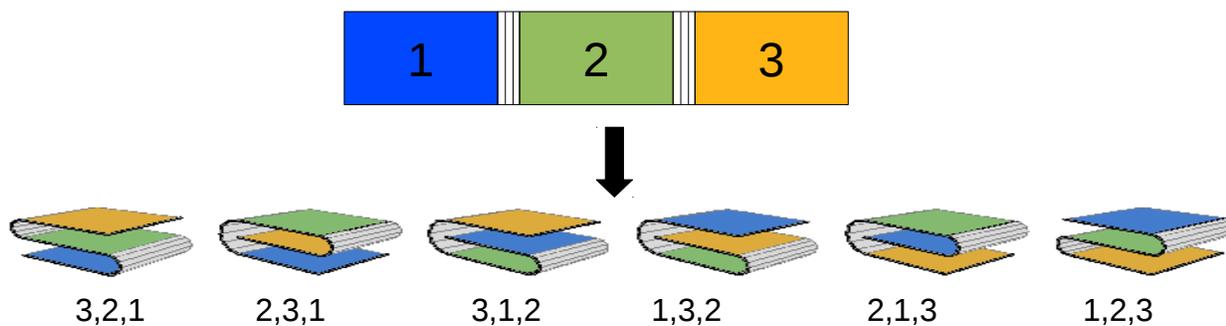


Figura 1

Y hay ocho maneras de plegar un mapa de 2×2 a lo largo de sus segmentos como muestra la Figura 2.

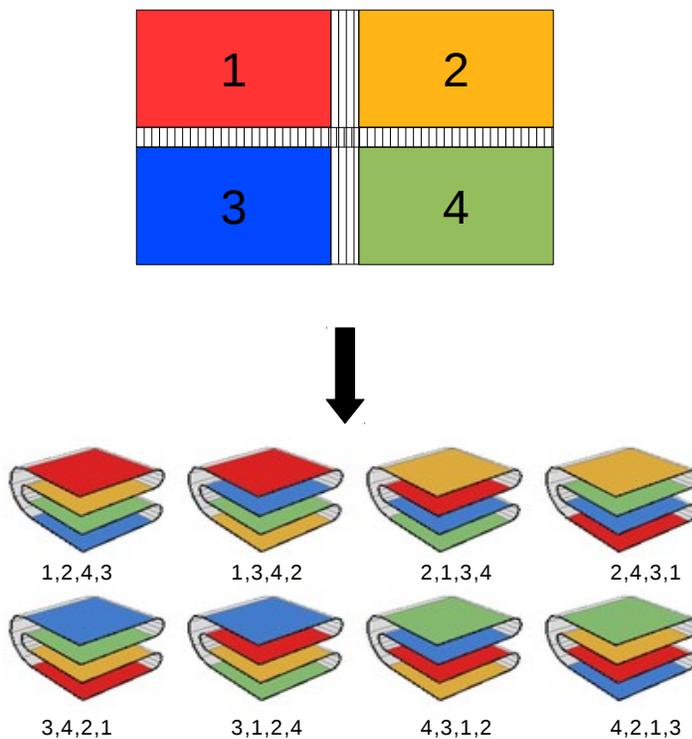


Figura 2

Hay dos problemas de decisión¹ con los que se va a trabajar:

1. El primer problema es dado un mapa de dimensión $m \times n$ y una permutación de los números $1, 2, \dots, m \times n$, decidir si esa permutación puede formarse con un plegado de mapa. Por ejemplo: para el mapa 2×2 mostrado en la Figura 3, formar la permutación 1, 3, 2, 4 no es posible. Los plegados posibles de este mapa se muestran en la Figura 2.

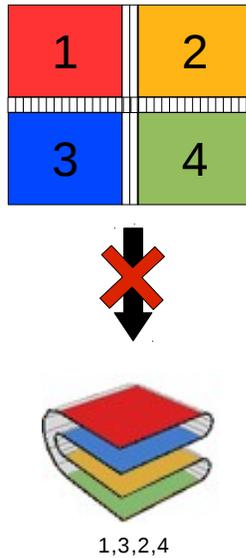


Figura 3.

2. El otro problema de decisión se puede llamar un “replegado”, ya que una vez que se hace el plegado y desplegado de un mapa se puede ver que se crean cierto tipo de patrones que se llaman “montañas” o “valles” lo que limitará la forma en que puede ser el plegado del mapa. Haremos una convención para los dibujos en donde los valles se representarán con puntos y las montañas se dibujan como líneas discontinuas como muestra la Figura 4.

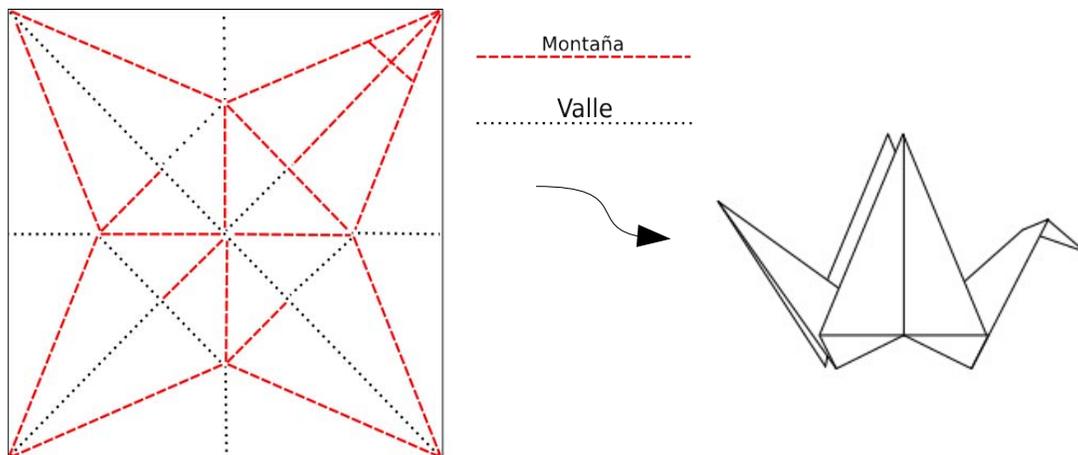


Figura 4.

¹ Un problema de decisión es aquel en el que cada instancia tiene una de dos respuestas posibles, por ejemplo sí o no, verdadero o falso, etc.

Por ejemplo en la Figura 5 se muestran dos mapas con sus montañas y valles, donde el mapa de la izquierda puede plegarse y el de la derecha no.

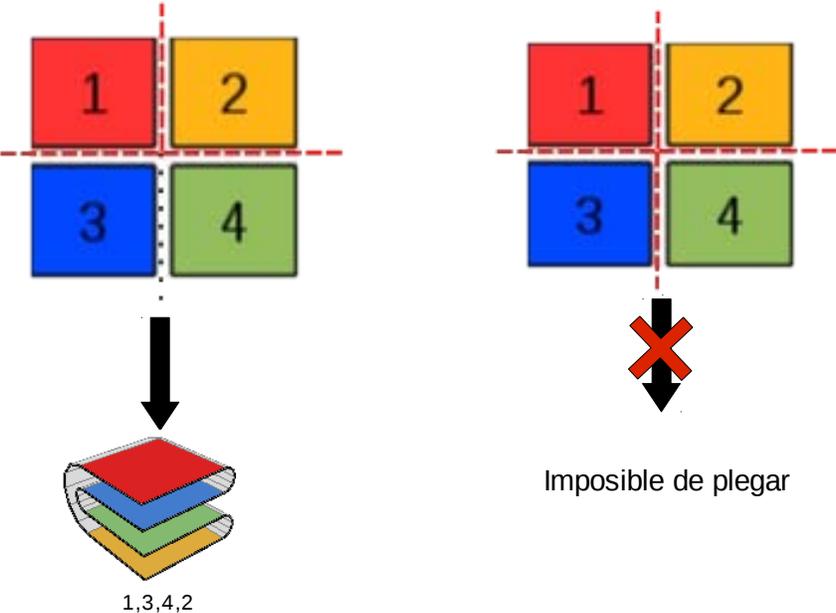


Figura 5.

Todos los ejemplos de mapas mostrados anteriormente se pueden ver como arreglos de pilas, en los que un pliegue hace que se vacíen unas pilas sobre otras. Por ejemplo en la Figura 6 se muestra el proceso de plegar un mapa que comienza con cuatro pilas y termina como una sola pila.

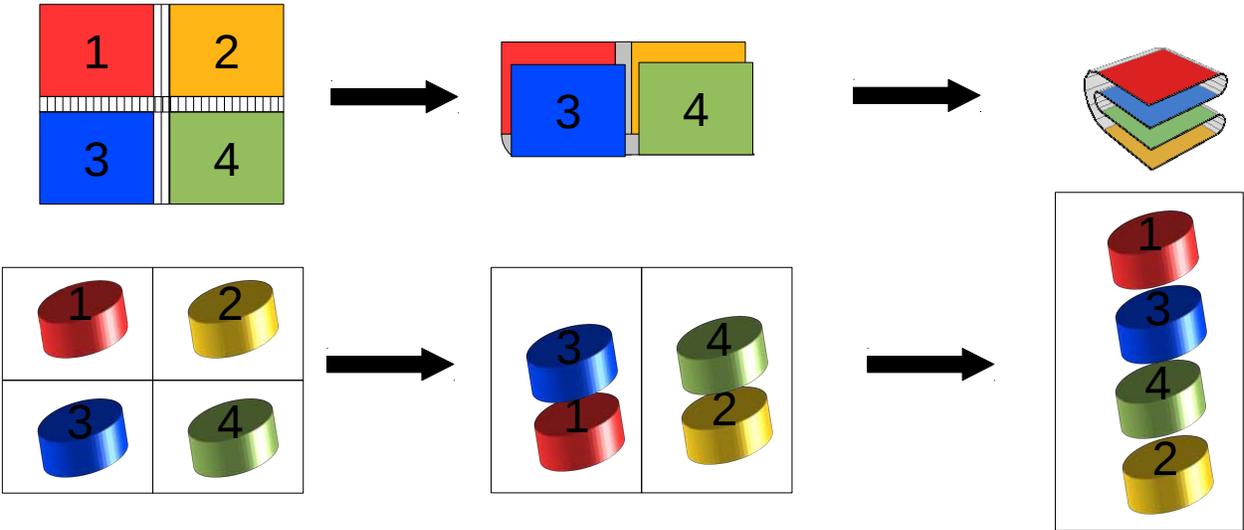


Figura 6.

Antecedentes

Referencias externas

When Can You Fold a Map? [3]. En este artículo se describe el problema de plegado de mapas y se presentan algunos resultados generales teóricos. En este artículo aparece el algoritmo que implementaremos para resolver el segundo problema de decisión. Además en este artículo se desarrollan algoritmos eficientes para plegados de dimensión $1 \times n$.

Referencias Internas

Generación de polígonos con triangulaciones ortogonales [4]. El proyecto propuesto también trata un tema de generación de objetos combinatorios, por lo cual tiene relación con este proyecto. Sin embargo se generan figuras a través de triángulos y en nuestro caso a través de cuadrados.

Algoritmos para coloración robusta de árboles binarios [5]. En este proyecto terminal se toca también un tema de generación de objetos combinatorios. Sin embargo, se generan árboles binarios y en nuestro caso se generan arreglos de cuadrados.

Algoritmo de búsqueda de configuraciones de enlaces ortogonales en dos y tres dimensiones [6]. Este es el proyecto terminal más parecido ya que aquí también se diseñan e implementan algoritmos de plegado para después ser visualizados en OpenGL. La diferencia es que las configuraciones de este proyecto terminal son de alambres que se doblan y no de mapas que se doblan.

Descripción técnica

El proyecto contará con una interfaz gráfica de usuario, para que el usuario interactúe con los diferentes módulos, los cuales se muestran en la Figura 3:

- Módulo de búsqueda de plegados de un mapa.
- Módulo que decida si cierto plegado es posible.
- Módulo de visualización de resultados.

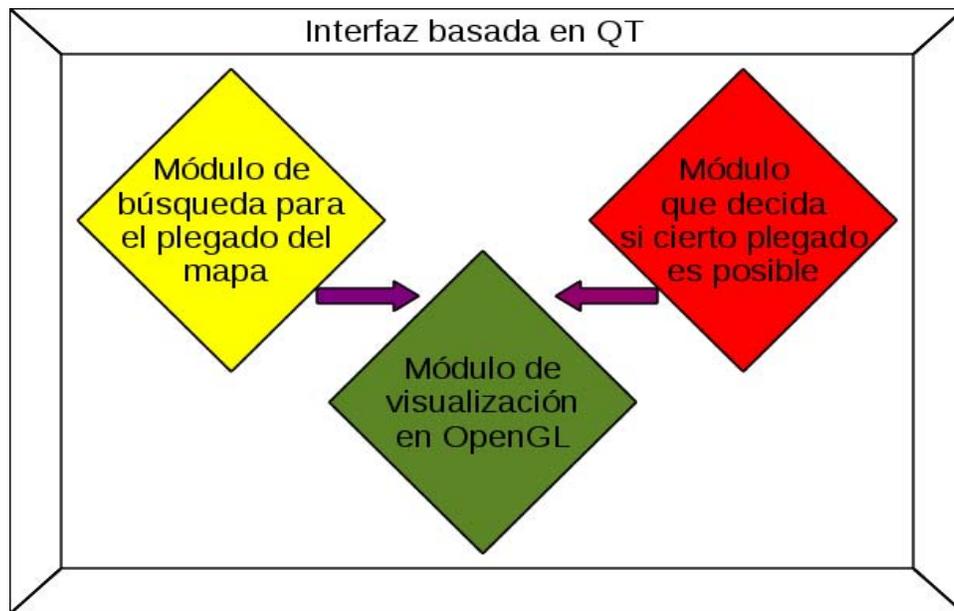


Figura 3.

Módulo de búsqueda para el plegado de un mapa. En este módulo se determinarán los plegados para mapas rectangulares dependiendo del tamaño que introduzca el usuario.

Se sabe que no hay una fórmula general para dar el número de formas distintas de plegar una mapa rectangular de $m \times n$. Pero se tienen valores de hasta $n \leq 28$ [7]. Algunos resultados se muestra en la Tabla 1.

n	1	2	3	4	5	6
1	1	2	6	16	50	1392
2	2	8	60	320	198	10512
3	6	60	1368	15552	201240	2016432

Tabla 1

Para que este módulo cumpla con el objetivo se diseñará un algoritmo con la técnica de búsqueda con retroceso, que genera todas las formas posibles de plegar un mapa además que se podrá saber cuántas formas hay. Estos resultados serán enviados al módulo de visualización para ver gráficamente los plegados.

Módulo que decida si cierto plegado es posible. En este módulo se determinará si un cierto plegado es posible. En este módulo estarán las dos formas de decisión explicadas en la parte de la justificación. El usuario elegirá si es por medio de la secuencia de caras o por el mapa de montañas y valles.

Para este módulo se diseñará, un algoritmo de búsqueda con retroceso y además se implementará el algoritmo contenido en [3].

Módulo de visualización de resultados. En este módulo se dibujará la animación del resultado de los módulos anteriores, en donde se mostrará cómo fueron realizados los plegados. Este módulo leerá los resultados generados por los módulos anteriores y genera la visualización.

Especificación técnica

Los módulos:

- Módulo de búsqueda de plegados de un mapa.
- Módulo que decida si cierto plegado es posible.
- Módulo de visualización de resultados.

Todos estos módulos estarán contenidos en una interfaz gráfica de usuario por medio de QT. En donde los dos primeros módulos serán implementados utilizando C o C++ y una vez que se obtengan los resultados, estos pasarán al módulo de visualización.

El módulo de visualización de resultados se implementará por medio de OpenGL, la cual es una Interfaz de programación de aplicaciones, que ofrecen al programador una interfaz con gráficos a nivel hardware.

El desarrollo del proyecto concluye con un programa ejecutable, en donde sus características mínimas serán que se ejecuten los módulos todos descritos anteriormente en una plataforma Linux. Y sus valores máximos de los mapas es de dimensión 10×10 y el valor mínimo es de dimensión 1×2 .

Se toma un valor máximo para mapas de 10×10 ya que el número de plegados es muy grande además con la generación de plegados con la técnica de búsqueda con retroceso, tiene una ejecución lenta.

A continuación se presenta un prototipo de la interfaz en Qt con una imagen generada en OpenGL (ver la Figura 6).

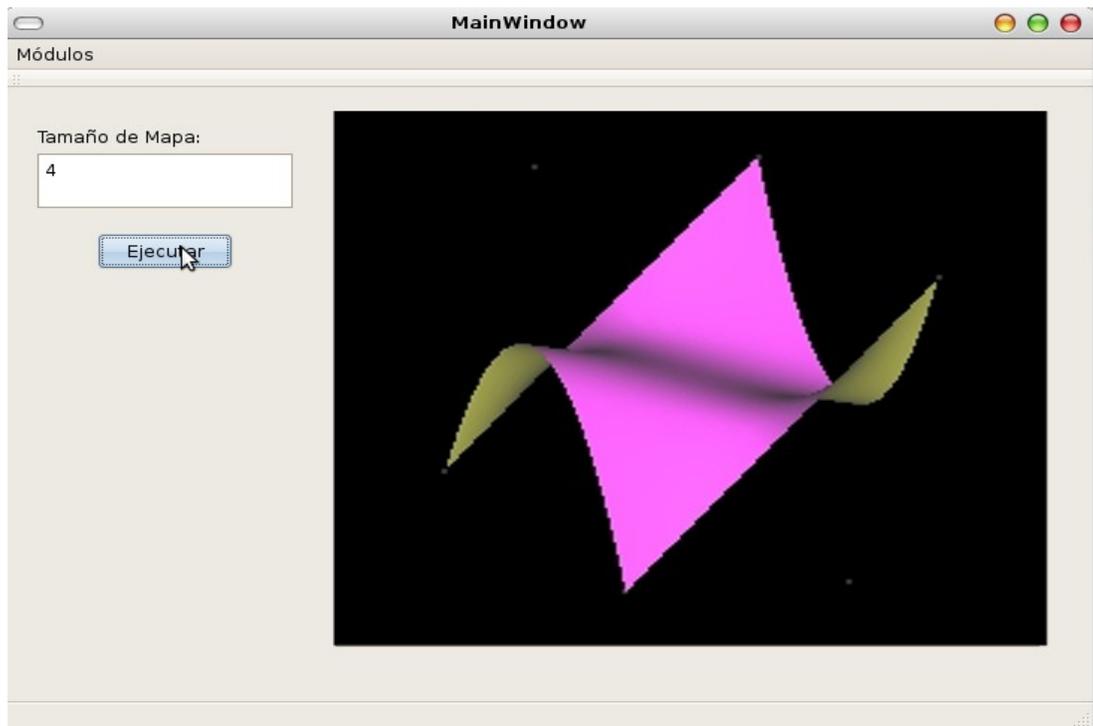


Figura 6.

La interfaz contará con un menú donde se elegirán los módulos:

- Módulo de búsqueda de plegados de un mapa.
- Módulo que decida si cierto plegado es posible.

Y en cada uno de ellos se pedirán los datos necesarios para ejecutarlos y hacer la visualización en OpenGL que estará del lado derecho de la interfaz.

A continuación se muestra la pantalla del módulo de decisión por medio de montañas y valles en la Figura 7, donde el usuario introduce el tamaño del mapa y marca cuales son las montañas y valles con las casillas de verificación, los valles son las casillas de verificación sin marcar y las montañas las casillas de verificación marcadas.

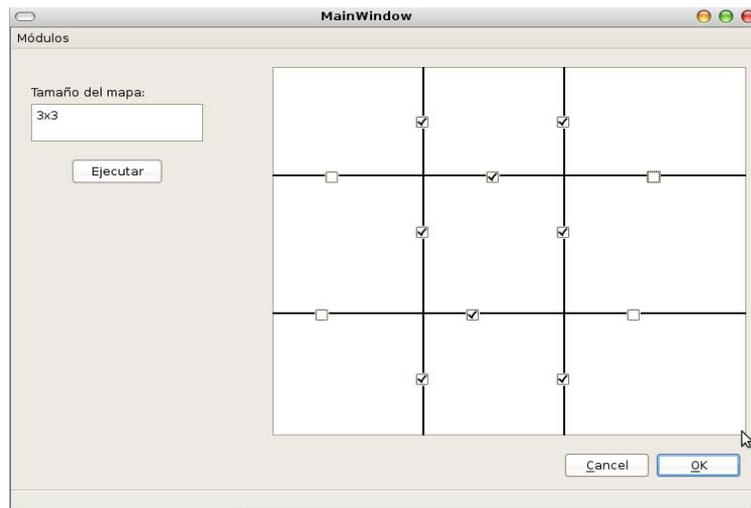


Figura 7.

Después de aquí el programa dará el resultado para ver si se puede plegar el mapa o no. En caso de que el plegado sea posible, se mostrará el procedimiento de plegado en OpenGL.

La otra parte del módulo de decisión es la de dar una secuencia de pliegues, y en la cual el programa decidirá si es posible formar un plegado de mapa con esa secuencia como muestra la Figura 8.

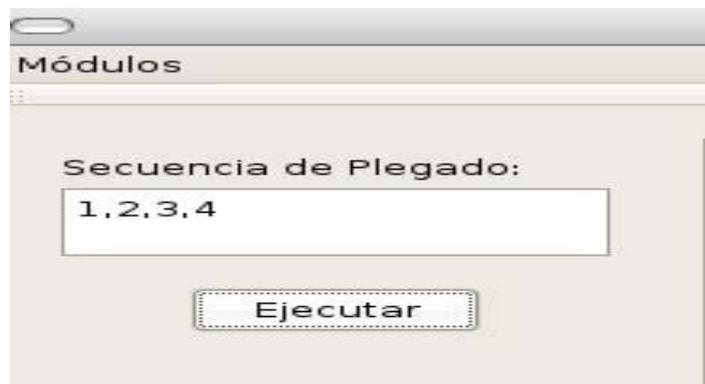


Figura 8.

Trabajos Futuros

- El desarrollo de otro algoritmo para mapas con otro tipo de formas (triangulares, hexagonales).
- La mejora del algoritmo para mapas más grandes.
- El análisis de la complejidad de los algoritmos desarrollados.

Entorno de desarrollo

Sistema Operativo Linux (Debian 6.0)

Lenguajes de programación C/C++, API de programación de QT y OpenGL

Productos de Trabajo

Los productos que se entregarán en un medio de almacenamiento son:

- Manuales de usuario e instalación.
- Código fuente documentado.
- Archivos para la ejecución.
- Reporte final en formato PDF

Calendario de Trabajo

A continuación se presenta el plan de trabajo de las actividades. Para el trimestre 12-I. Sólo se pone un calendario ya que sólo me falta la UEA de Proyecto terminal II, por lo que cumpliré con las horas necesarias para terminarlo.

Actividades por semana	Trimestre 12-I										
	1	2	3	4	5	6	7	8	9	10	11
Implementación de la interfaz QT	■	■	■								
Implementación del módulo de búsqueda de plegados de un mapa			■	■	■	■					
Implementación del módulo que decida si cierto plegado es posible						■	■	■	■	■	
Pruebas										■	■
Elaboración de reporte	■	■	■	■	■	■	■	■	■	■	
Elaboración de manuales y disposición de documentación del proyecto			■	■	■	■	■	■	■	■	■

Recursos

Los recursos necesarios para la realización de la actual propuesta son:

- Computadora tipo PC con procesador tipo AMD 64 bits, Disco Duro de 40GB, 2Gb memoria RAM.
- GNU Compiler Collection.
- OpenGL versión 4.0
- QT

Y están disponibles por parte del alumno y se cuenta con todas las aplicaciones de software necesarias ya que son software libre.

Bibliografía

- [1] J. Royo Prieto, "Matemáticas y papiroflexia ", *Sigma*, no. 21, pp. 2175-192, Sep. 2002
- [2] M. Gardner. "The Combinatorics of Paper Folding." in *Wheels, Life, and Other Mathematical Amusements*. Ed. New York: W.H. Freeman, pp.60-73, 1983.
- [3] E. M. Arkin et al., "When can you fold a map?," *Comput. Geom. Theory Appl.*, vol. 29, no. 1, pp. 23-46, Sep. 2004.
- [4] C. Olaguibert Segura, "Generación de polígonos con triangulaciones ortogonales ", Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, D.F,México, 2009
- [5] C. A. Rodríguez Villalobos, "Algoritmos para coloración robusta de árboles binarios", Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, D.F,México, 2009
- [6] P. Torres Jiménez, "Algoritmo de búsqueda de configuraciones de enlaces ortogonales en dos y tres dimensiones ", Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, D.F,México, 2009
- [7] W. Lunnon, "Multi-Dimensional Strip Folding." *Computer J.* no. 14, pp 75-79, 1971.

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería
Ingeniería en Computación

Algoritmos para plegado de mapas cuadrados

REPORTE DE PROYECTO TERMINAL

Alumno:

Ramos León Anthony Damián
207200255

Asesores:

Dr. Francisco Javier Zaragoza Martínez
Departamento de Sistemas
Profesor Titular

M. en C. Cynthia A. Rodríguez Villalobos
Departamento de Sistemas
Profesora Asociada

Tabla de Contenido

Resumen.....	3
Introducción.....	4
Historia del Origami.....	4
La relación de la matemática y computación con el origami.....	4
Objetivos.....	5
Objetivos particulares.....	5
El problema de plegado de mapa a través de dobleces sencillos	5
Descripción Técnica.....	9
Algoritmo generador de dobleces con búsqueda con retroceso.....	10
Algoritmo decisión	10
Algoritmo de decisión de replegado.....	11
Resultados.....	13
Conclusiones.....	15
Bibliografía.....	16

Resumen

En este proyecto se implementaron dos algoritmos para plegados en todas las capas utilizando la técnica de búsqueda con retroceso, en la cual se van generando las soluciones por etapas hasta llegar a una solución o varias, el problema de esta técnica es que cuando el conjunto de soluciones va incrementando exponencialmente o factorialmente también lo hace el tiempo de búsqueda.

Y un tercer algoritmo que se implementó fue el que venía contenido en el artículo E. M. Arkin et al., “When can you fold a map?” *Comput. Geom. Theory Appl.*, vol. 29, no. 1, pp. 23–46, Sep. 2004. el cual describe el algoritmo de decisión para una línea utilizando plegado en todas las capas.

Introducción

Historia del Origami

La palabra Origami proviene del japonés, que está compuesta de las palabras Ori que significa doblar y de la palabra kami que significa papel. El origami fue popularizado en gran medida por la cultura japonesa. ¿Cuándo y dónde surgió el origami? Es un tema de debate, pero generalmente se cree que comenzó en China, Corea, y Japón, a su vez se cree que fue inventado por Ts'ai Lun, un funcionario de la corte China, en 105 DC [5].

Con la invención del papel y este viajando de país a país, una teoría dice que algunos modelos de origami viajaron junto con el papel. Otra teoría es que el plegado de papel fue descubierto de forma independiente en varios lugares. En cualquier caso, el origami surgió en varios países después de China, Corea y Japón, sobre todo en España, Italia, Inglaterra, y hace muy poco tiempo en América del Norte [5].

La historia del Origami en Japón está bien documentada. Durante el período Heian, 794-1185 d. de JC, se desarrollaron pliegues para las ceremonias y algunos modelos básicos. El Origami se extendió en todo Japón durante el período Kamakura, 1185-1333 d. de JC. El Origami moderno se cree que fue desarrollado durante el período Muromachi, 1333-1573. Por el período Edo, 1603-1867 d. de JC, el origami se había convertido en un pasatiempo popular. El primer libro para describir el origami es de 1682. Por el período Meiji, 1868-1912, el origami era de uso popular en las escuelas, como lo es ahora en Japón [5].

El resurgimiento del interés en el origami en el siglo XX y XXI, y su propagación por el mundo, a menudo se atribuye a la influencia del artista Akira Yoshizawa (1911 - 2005), que introdujo el sistema de notación de origami líneas punteadas y las flechas en su libro del año 1954, el mismo sistema se sigue utilizando en la actualidad con ligeras modificaciones. Lillian Oppenheimer (1898-1992) desempeñó un papel clave en la popularización de origami en los Estados Unidos [5].

La relación de la matemática y computación con el origami.

El Origami tiene una geometría intrínseca que es un tema de estudio. El libro más antiguo conocido en referencia al Origami en el contexto de la geometría es de 1840 por el Rev. Dionisio Lardner, que ilustra varios conceptos geométricos con el plegado de papel. Uno de los trabajos más influyentes y accesibles es un libro de 1893 por T. Sundara Fila, que ilustra una variedad de las construcciones geométricas a través de origami, que tradicionalmente se ejecutan por regla y compás. Implícito en este trabajo muestra que el origami puede simular las construcciones de la geometría euclidiana, pero sólo como herramienta de ilustración, en lugar del análisis de sí mismo. Después aparece un trabajo realizado por el italo-japonés Humiaki Huzita que es considerado la primera contribución a la matemática del origami. Ya que en este se proponen varios axiomas que se asemejan a los axiomas de Euclides.

En consecuencia aparecen varios teoremas fundamentales sobre los patrones de pliegue en torno a un único plano plegado, establecidos por Jun Maekawa, Toshikazu Kawasaki, y Justin Jacques. Thomas Hull extendió este trabajo de plegado de un plano en una serie de publicaciones, 1994, 2002, 2003 y más recientemente en un libro de matemáticas del origami en 2006 [5].

Robert Lang uno de los mejores origamistas de nuestros tiempos desarrolla un algoritmo en alrededor del año de 1993, después publica un libro llamado Diseños secretos del origami en 2003, aplicando matemáticas, su trabajo es considerado en inicio del Origami computacional, en donde Erik Demaine ha hecho grandes contribuciones [5].

Además, la mayoría de estos problemas tienen estrechas relaciones con importantes disciplinas como son:

- Geometría Plana
- Trigonometría
- Geometría del espacio
- Cálculo y Geometría Diferencial
- Álgebra lineal
- Teoría de grafos
- Teoría de grupos
- Complejidad / computabilidad
- Geometría Computacional
- Análisis y diseño de algoritmos
- Estructuras de datos

En la matemática combinatoria el problema de plegado plano es la cuestión de calcular de cuántas maneras se pueden doblar un mapa rectangular a lo largo de sus pliegues.

Este problema fue propuesto hace varios años por Stanislaw M. Ulam [2]. Por lo tanto el objetivo de este proyecto es diseñar e implementar algoritmos que busquen las diferentes formas que puede adoptar una hoja con plegados ortogonales, así como también decidir si es posible un plegado dado.

Objetivos

Objetivos particulares

- I. Diseñar e implementar un módulo que busque todas las diferentes formas de plegado ortogonal de un mapa a través de pliegues sencillos en todas las capas.
- II. Diseñar e implementar un módulo que decida si cierto replegado es posible.

El problema de plegado de mapa a través de dobleces sencillos

Este proyecto terminal se basará únicamente en un problema llamado **plegado plano a través de plegados simples en todas las capas**. En la matemática combinatoria el problema de **plegado plano** es la cuestión de calcular de cuántas maneras se puede doblar un mapa rectangular a lo largo de sus pliegues. Este problema fue propuesto hace varios años por Stanislaw M. Ulam [2].

Un pliegue simple es aquel en el que se dobla a lo largo de una línea.

Hay tres tipos de pliegues simples los cuales son:

- Una sola capa
- Varias capas
- Todas las capas

como muestra la Figura 1.

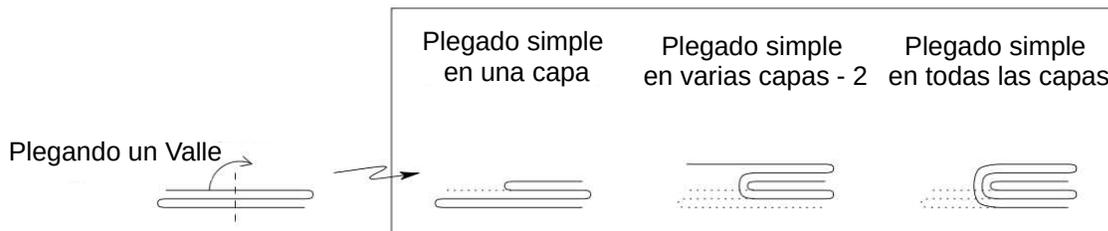


Figura 1. Plegados simples

Por lo tanto el objetivo de este proyecto es diseñar e implementar algoritmos que busquen las diferentes formas que puede adoptar una hoja con pliegados ortogonales mediante pliegado simple en todas las capas sin que las capas se crucen, así como también decidir si es posible un pliegado dado que la mayoría de pliegados está en este conjunto.

Dos pliegues suelen ser considerados "diferentes" en este contexto, si difieren en el orden total de las caras en el plegamiento. Estos problemas han sido estudiados extensamente, en particular para los mapas de dimensión $1 \times n$, pero siguen en gran medida sin resolverse.

Por ejemplo hay seis maneras de plegar una tira de tres segmentos utilizando pliegado simple en todas las capas, como muestra la figura 2.

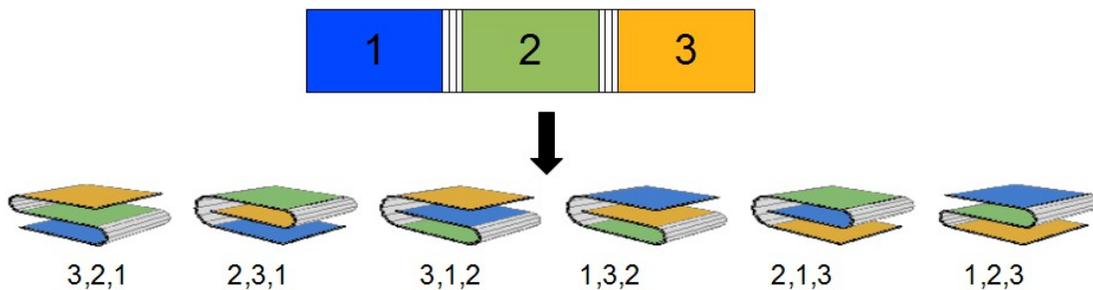


Figura 2 plegados de 1×3

Y hay ocho maneras de plegar un mapa de 2 x 2 con plegados simples en todas las capas como muestra la figura 3

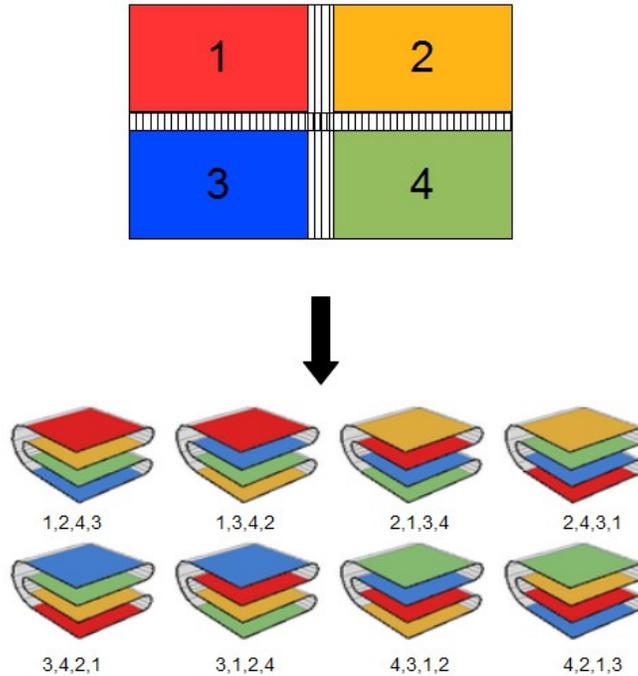


Figura 3 todos los plegados de un mapa de 2 x 2

Hay dos problemas de decisión¹ con los que se va a trabajar:

El primer problema de decisión es dado un mapa de dimensión $m \times n$ y una permutación de los números $1, 2, \dots, m \times n$, decidir si esa permutación puede formarse con un plegado de mapa. Por ejemplo: para el mapa 2x2 mostrado en la Figura 4, formar la permutación 1, 3, 2, 4 no es posible. Los plegados posibles de este mapa se muestran en la Figura 3.

¹ Un problema de decisión es aquel en el que cada instancia tiene una de dos respuestas posibles, por ejemplo sí o no, verdadero o falso, etc.

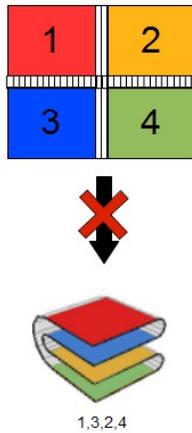


Figura 4 Plegado imposible

El segundo problema de decisión se puede llamar “replegado”, ya que una vez que se hace el plegado y desplegado de una tira de papel se puede ver que se crean cierto tipo de patrones que se llaman “montañas” o “valles” lo que limitará la forma en que puede ser el plegado del mapa. El algoritmo decidirá si dado el patrón de montañas y valles puede plegarse totalmente la tira usando dobleces simples en todas las capas. Por ejemplo en la Figura 5.



Figura 5. Algoritmo de replegado

Este último algoritmo de replegado es mostrado en el artículo de M. Gardner. "The Combinatorics of Paper Folding." in *Wheels, Life, and Other Mathematical Amusements*. Ed. New York: W.H. Freeman, pp.60-73, 1983. y en el proyecto se hizo la implementación.

El plegado y replegado es un tema apasionante de la geometría computacional. Hay una amplia gama de problemas de plegado y replegado, algunos que se remontan varios siglos atrás y todavía no se resuelven, como el plegado de poliedros convexos, mientras que otros son más recientes, como el plegado de proteínas. En los últimos años ha habido un tremendo progreso en muchos de los problemas fundamentales de plegado y replegado, sin embargo, algunas de las preguntas más importantes quedan pendientes. Esto deja el tema en un estado interesante. Los distintos problemas de plegado son atractivos desde el punto de vista puramente matemático, por la belleza de los mismos problemas.

Descripción Técnica

Este proyecto fue desarrollado sobre el lenguaje de programación C++ apegado al estándar ANSI C++.

Consiste en varios archivos de código fuente los cuales contienen cada uno, un algoritmo del proyecto de forma independiente con todo lo necesario para su buen funcionamiento, además cuenta con una Interfaz Gráfica que fue desarrollada utilizando QT 4.0 desde aquí se puede ejecutar los algoritmos implementados .

Para poder ejecutar, compilar y ejecutar estos programas es necesario tener instalado un compilador de C++ .

El nombre de los archivos son los siguientes:

- plegadonxnVSILVERbusqueda.cxx (implementación del algoritmo de búsqueda con retroceso para generar plegados)
- plegadonxnVSILVERdecision.cxx (implementación del algoritmo de decisión de una permutación)
- ReplegadoV8.0.cxx (implementación de algoritmo de Replegado)

El funcionamiento de cada uno de estos programas es sobre la línea de comandos.

Compilación:

```
$ g++ <Nombre> -o <Nombre>
```

Ejecutar:

```
$/<Nombre>
```

La salida del plegadonxnVSILVERbusqueda.cxx programa es en un archivo llamado pelgado.txt

La salida de las otras dos implementaciones es en la salida estándar.

Adicionalmente se proporciona un script (.sh) y utilizando el compilador GNU C++) el cual compilará los archivos y mostrará un menú para ejecutar el algoritmo deseado.

Para usar la interfaz gráfica es necesario tener las bibliotecas de QT versión 4.0 esta cuenta con un script (.sh) que compila y ejecuta el proyecto en la siguiente figura No. 6 se muestra la interfaz gráfica.

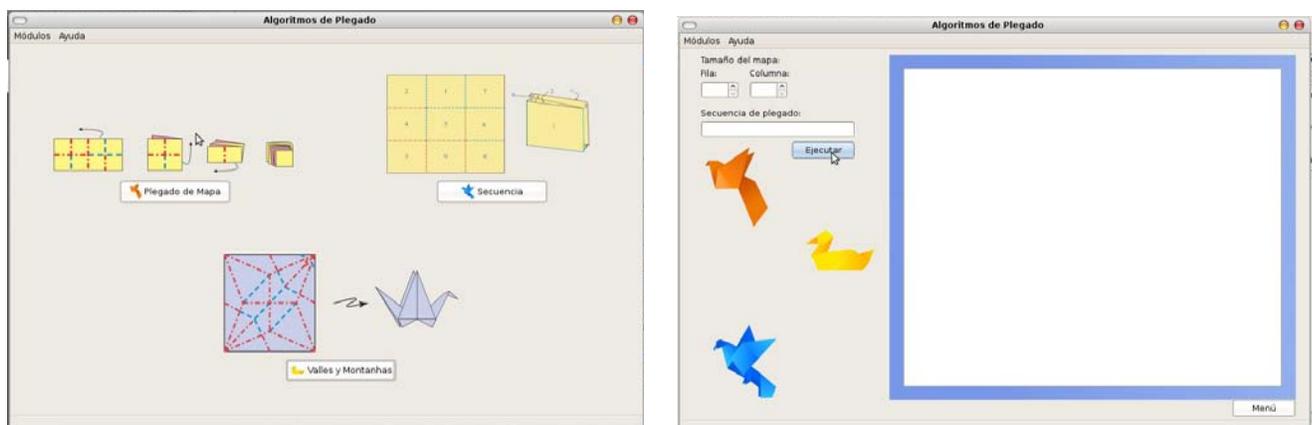


Figura No. 6. Interfaz Gráfica

Algoritmo generador de dobleces con búsqueda con retroceso

El algoritmo desarrollado obtiene al conjunto de todas la formas en que se puede plegar un mapa rectangular de $n \times m$ a través de pliegues sencillos en todas las capas, la única entrada sera el tamaño de n y m para formar dicho conjunto sea este V .

T y I son variables que indican en que parte del mapa se va hacer el plegado vertical y horizontal respectivamente.

Algoritmo 1 Pseudocódigo para el algoritmo exhaustivo

```
1: FuncionPlegados(mapa, iniciox, finx, iniciy, finy )
2: Si iniciox== finx y iniciy == finy entonces
3:     Hay una solución
4: si no /*Generar soluciones */
5:     T = iniciy , I = iniciox
6:     Mientras T < finy ó I < iniciox
7:         Generar Monte en I
8:         Funcionplegados(mapa,N,finx,iniciy,finy);
9:         Deshacer Monte I
10:        Generar Valle en I
11:        Funcionplegados(mapa,iniciox,i,iniciy,finy);
12:        Deshacer Valle en I
7:        Generar Monte en T
8:        Funcionplegados(mapa,N,finx,iniciy,finy);
9:        Deshacer Monte
10:       Generar Valle en T
11:       Funcionplegados(mapa,iniciox,i,iniciy,finy);
12:       Deshacer Valle en T
```

Algoritmo decisión

Sea V el conjunto de los elementos con pliegues sencillos en todas las capas de $m \times n$ y sea x una permutación de $1, 2, \dots, n \times m$.

El algoritmo desarrollado nos dice si $x \in V$, si esto es afirmativo nos muestra como se forma ese pliegue.

La entrada para este algoritmo es el tamaño de n y m asi como también la permutación que se desea ver si pertenece al conjunto de configuraciones de plegados en todas las capas.

T y I son variables que indican en que parte del mapa se va hacer el plegado vertical y horizontal respectivamente.

Algoritmo 2 Pseudocódigo para el algoritmo de decisión

```
1: FuncionPlegados(mapa, iniciox, finx, iniciy, finy, permutacion )
2: Si iniciox== finx y iniciy == finy y permutacion == mapa [finx][finy]
3:   El mapa se puede plegar y mostrar la construcción del plegado.
4: si no /*Generar soluciones */
5:   T = iniciy , I = iniciox
6:   Mientras T < finy ó I < iniciox
7:     Generar Monte en I
8:     plegados(mapa,N,finx,iniciy,finy);
9:     Deshacer Monte
10:    Generar Valle en I
11:    plegados(mapa,iniciox,i,iniciy,finy);
12:    Deshacer Valle en I
7:     Generar Monte en T
8:     plegados(mapa,N,finx,iniciy,finy);
9:     Deshacer Monte
10:    Generar Valle en T
11:    plegados(mapa,iniciox,i,iniciy,finy);
12:    Deshacer Valle en T
```

Algoritmo de decisión de replegado

El algoritmo fue desarrollado en el artículo M. Gardner. "The Combinatorics of Paper Folding." in *Wheels, Life, and Other Mathematical Amusements*. Ed. New York: W.H. Freeman, pp.60-73, 1983. en el proyecto se hizo una implementación de este para una tira de papel (1- Dimesión) para dobleces sencillos en todas las capas.

La implementación del algoritmo solo necesita una cadena donde esta representa las montañas y valles así como la distancia entre cada uno de ellas.

Ejemplo Figura 7:



Figura 7. Ejemplo de Cadena de entrada

La cadena de la Figura 6 sería: 4M2VM5V6

Sea S la cadena de entrada.

Algoritmo 3 Pseudocódigo para el algoritmo de decisión de replegado

```
1: Replegado(S)
2: Para todo i=1 , i < N , i+=2
3:     1 ≤ fin ≤ min(i - 1, N - i)
4:     Si ( i == 1 || i == (N-2) )
5:         Si N > 3
6:             Si S[i - x] = comp(S[i + x])
7:                 Se puede doblar
8:                 GenerarNueva S
9:                 Replegado(S)
10:     Si N == 3
11:         Se puede doblar
12:         GenerarNueva S
13:         Replegado(S)
14: Si no
15:     Para todo x=1; x ≤ fin ; x++
16:         S[i - x] == comp(S[i + x])
17:         Se puede doblar
18:         GenerarNueva S
19:         Replegado(S)
```

Resultados

Con el algoritmo generador de plegados en todas las capas se generó la Tabla No. 1 obteniendo los siguientes resultados.

$n \times m$	1	2	3	4	$n = m$
1	1	2	6	16	1
2	2	8	40	168	8
3	6	40	296	1720	296
4	16	168	1720	12960	12960
5	50	712	9320		
6	144	2752	45120		
7	448	10568	208280		
8	1328	38568			
9	4052	138856			
10	12040				

Tabla No. 1

La Tabla No. 2 contiene los tiempos que tardo en calcularse los plegados.

	1	2	3	$n = m$
1	0m0.000s	0m0.000s	0m0.000s	0m0.000s
2	0m0.004s	0m0.004s	0m0.004s	0m0.008s
3	0m0.000s	0m0.004s	0m0.008s	0m0.016s
4	0m0.004s	0m0.012s	0m0.324s	0m19.533s
5	0m0.008s	0m0.132s	0m13.761s	
6	0m0.040s	0m3.448s	9m43.368s	
7	0m0.712s	1m48.051s		
8	0m16.745s	56m49.341s		
9	6m55.406s			
10				

Tabla No. 2

Se consulto información en la siguiente pagina:

N. J. A. Sloane, (2007), The On-Line Encyclopedia of Integer Sequences, www.research.att.com/~njas/sequences/,

En donde se consultaron las siguientes secuencias : A000136, A001010, A001415, A001416, A001418. Con la series consultadas se llenó la Tabla No. 3

$n \times m$	1	2	3	$n = m$
1	1	2	6	1
2	2	8	60	8
3	6	60	1368	1368
4	16	320	15552	300608
5	50	1980	2016432	186086600
6	144	10512	21582624	
7	462	60788		
8	1392	320896		
9	4536	1787904		
10	14060	9381840		

Tabla No. 3

Llamemos al conjunto de pliegues sencillos en todas las capas V y al conjunto que contiene todos los pliegues simples U .

La Tabla No. 3 solo contiene la $|U|$ y es la única información que se obtiene de aquí.

Estas secuencias tienen contados los pliegues *totales* para el plegado de una mapa (aquí están pliegues simples en una sola capa, varias capas y todas las capas).

Un ejemplo donde un $x \in U$ pero $x \notin B$ para el caso de 7×2 en donde tenemos que:

$$|V|=448 \text{ y } |U|=462$$

Para este caso nos apoyamos en el Wolfram CDF Player y `FoldingAStripOfLabeledStamps.cdf` que pueden ser descargados. Este un ejemplo donde muestra U y a todos sus elementos $|U|=462$

El elemento es:

$x = 6, 1, 7, 2, 3, 4, 5$ como muestra la figura 7.

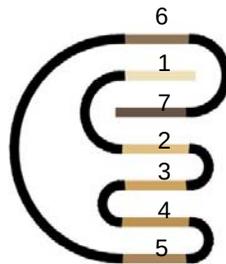


Figura 7.

Para plegar este elemento se hace de la siguiente forma y se utiliza un plegado simple en varias capas.

[1][2][3][4][5][6][7] → [2, 1][3][4][5][6][7] → [3,2, 1][4][5][6][7] → [2, 1][3][4][5][6,7] → [3,2, 1][4][5][6, 7] → [1,2,3,4][5][6, 7] → [5,4,3,2, 1][6, 7] hasta aquí eran plegados en todas las capas el siguiente solo pliega una capa obteniendo [5,4,3,2][6, 1, 7] y finalmente [6, 1, 7, 2, 3, 4, 5]

Por lo tanto $V \subset U \rightarrow |V| \leq |U|$

Conclusiones

El proyecto resultó mucho más complejo de lo que inicialmente parecía, por lo que fue necesario redefinir el alcance de éste. Las limitantes de tiempo no permitieron que se trabajara en todo el intervalo propuesto inicialmente ($n = 10$ y $m = 10$), ya que el tiempo de ejecución del algoritmo aumenta muy rápidamente cuando aumentan los valores de n y m .

Además el algoritmo de replegado se tenía pensado para \mathbb{R}^2 y solo se implementó para \mathbb{R} .

Bibliografía

- [1] J. Royo Prieto, “Matemáticas y papiroflexia ”, *Sigma*, no. 21, pp. 2175-192, Sep. 2002
- [2] M. Gardner. "The Combinatorics of Paper Folding." in *Wheels, Life, and Other Mathematical Amusements*. Ed. New York: W.H. Freeman, pp.60-73, 1983.
- [3] E. M. Arkin et al., “When can you fold a map?,” *Comput. Geom. Theory Appl.*, vol. 29, no. 1, pp. 23–46, Sep. 2004.
- [4] W. Lunnon, "Multi-Dimensional Strip Folding." *Computer J.* no. 14, pp 75-79, 1971.
- [5] Erik D. Demaine and Joseph O'Rourke. 2008. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra* (Reprint ed.). Cambridge University Press, New York, NY, USA.