

DISEÑO DE UNA METODOLOGÍA PARA LA LECTURA Y VISUALIZACIÓN DE UN MAPA RASTER UTILIZANDO LENGUAJE C#

Álvaro Peraza Garzón¹, Mónica Avelina Gutiérrez Haros², Juan Francisco Peraza Garzón³

¹Instituto Tecnológico de Mazatlán (MÉXICO)

²Universidad Politécnica de Sinaloa (MÉXICO)

³Universidad Autónoma de Sinaloa, Facultad de Informática Mazatlán (MÉXICO)

Resumen

Este estudio presenta una metodología para la lectura y visualización de mapas raster utilizando el lenguaje de programación C#. Los mapas raster, son ampliamente utilizados en Sistemas de Información Geográfica (SIG), contienen información geoespacial en una estructura matricial que requiere herramientas especializadas para su análisis y representación. La metodología propuesta permite acceder a la información de píxeles individuales de un mapa raster mediante programación, sin depender exclusivamente de software SIG preexistente. Se desarrolló un software experimental denominado Mapixia Beta, que procesa archivos raster, garantizando su correcta lectura y visualización. Se realizaron pruebas comparativas con el software Idrisi Selva, verificando la exactitud en la representación de los mapas. Los resultados evidencian que el método propuesto es eficiente para interpretar mapas raster y visualizarlos con precisión, abriendo nuevas posibilidades para su aplicación en análisis geoespacial, planificación territorial y modelado ambiental. Se identificó como trabajo futuro la implementación de soporte para datos raster de tipo real y la optimización de la paleta de colores para una mejor interpretación visual.

Palabras clave: Bitmap, dato geoespacial, lenguaje C#, mapas raster, SIG.

Abstract

This study presents a methodology for reading and visualizing raster maps using the C# programming language. Raster maps, widely used in Geographic Information Systems (GIS), contain geospatial information in a matrix structure that requires specialized tools for analysis and representation. The proposed methodology allows access to individual pixel information from a raster map through programming, without relying exclusively on pre-existing GIS software. An experimental software called Mapixia Beta was developed to process raster files, ensuring their correct reading and visualization. Comparative tests were conducted with Idrisi Selva software, verifying the accuracy of map representation. The results demonstrate that the proposed method is efficient for interpreting raster maps and visualizing them accurately, opening new possibilities for their application in geospatial analysis, territorial planning, and environmental modeling. Future work includes implementing support for real-type raster data and optimizing the color palette for better visual interpretation.

Keywords: Bitmap, geospatial data, C# Language, raster maps, SIG.

1 INTRODUCCIÓN

En una sociedad cada vez más interconectada, el acceso a mapas y sus datos geoespaciales o geográficos, impulsa el desarrollo en cualquier ámbito, como en el transporte, el monitoreo social, ambiental, etc. Además, facilita la implementación de soluciones tecnológicas como los sistemas de navegación, de geolocalización, aparte de contribuir al desarrollo de ciudades inteligentes. Su integración en la vida cotidiana y en sectores estratégicos refuerza su papel como herramienta indispensable para el desarrollo sostenible y la innovación [1]. El manejo de estos datos tiene una amplia aplicación en diversos sectores como la planificación urbana, la gestión de recursos naturales, la agricultura, la logística, la respuesta a emergencias, entre otros. Esta tecnología permite analizar patrones espaciales y temporales, lo que permite servir como una herramienta para la toma de decisiones [2].

En la representación de la información geoespacial se utilizan principalmente dos formatos de mapas, los vectoriales o shape (con extensión de archivo *.shp) y los matriciales o raster (con extensión *.rst), cada uno de ellos tienen sus características específicas ideales para ciertos tipos de cálculos, entre los que destacan: análisis de usos de suelo, modelado, simulación territorial, etc.

A través de los Sistemas de Información Geográfica (SIG) se puede procesar la información en formato raster [3], pueden realizarse tareas como [4] [5]:

- **Análisis de Superficies:** Evaluación de elevaciones, pendientes y orientaciones del terreno para aplicaciones en geología, hidrología y planificación urbana.
- **Clasificación de Uso del Suelo:** Identificación y categorización de diferentes tipos de cobertura terrestre mediante imágenes satelitales.
- **Análisis de Riesgos Naturales:** Determinación de áreas susceptibles a inundaciones, deslizamientos u otros desastres naturales.
- **Planificación de Infraestructuras:** Selección de rutas óptimas para carreteras, líneas eléctricas u otras infraestructuras considerando el relieve y otros factores geográficos.

Estas tareas, se han realizado en distintas investigaciones como "Evaluación Multicriterio y Delimitación de Espacios Funcionales: Aplicación SIG para la Definición de Mapas Comarcales" en este trabajo, se emplea una combinación lineal de capas raster con diferentes criterios para delimitar espacios funcionales en mapas comarcales [6]. También en "Análisis espacial de los cambios de usos del suelo. Aplicación con SIG y teledetección" en este estudio se analizan los cambios en los usos del suelo mediante técnicas de SIG y teledetección, elaborando cartografía temática homogénea en formato raster con un tamaño de píxel de 20x20 metros [7].

Cabe señalar que en las diversas investigaciones analizadas, era notable la falta de una metodología específica para trabajar con un mapa raster fuera del ámbito de los SIG, por ejemplo, para acceder a la información de un píxel específico desde un lenguaje de programación. La falta y necesidad de tal información, fue el motivo principal de la presente investigación, siendo el objetivo central el de proporcionar una metodología para la lectura y visualización de un mapa raster, específicamente de aquellos con valores enteros y de tipo byte, que para este caso se utilizó el lenguaje C#.

En la siguiente sección se presentan los elementos necesarios para conocer y comprender las distintas herramientas y procedimientos, utilizados en el desarrollo de las actividades.

2 METODOLOGÍA

Se presenta en dos secciones (materiales y métodos), el conjunto de herramientas como: El lenguaje C# y los comandos necesarios para el manejo de archivos, los Sistemas de Información Geográfica (SIG) necesarios para el procesamiento de información geoespacial, y se analiza también la estructura interna de los mapas raster necesarios para los experimentos. Por último se detallan las funciones para la apertura y muestra de un mapa utilizando C#.

2.1 Materiales

2.1.1 Lenguaje C#

C# es un lenguaje de programación, orientado a objetos y desarrollado por Microsoft como parte de su plataforma .NET. Fue diseñado combinando características de lenguajes como C++ y Java, es ideal tanto para aplicaciones de escritorio como para desarrollo web y móvil. Ofrece una sintaxis clara y robusta, además de un sólido manejo de memoria. Es ampliamente utilizado en el desarrollo de aplicaciones empresariales y servicios en la nube. Su interoperabilidad con otros lenguajes y plataformas, junto con un fuerte soporte de herramientas, lo convierte en una opción versátil y completa [8]. Para la presente investigación se utilizó el lenguaje C# de Microsoft Visual Studio 2010 (64 bits) – Versión 10.0.40219.1 SP1.

2.1.2 Sistemas de Información Geográfica

Un Sistema de Información Geográfica (SIG) es una herramienta computacional diseñada para recopilar, almacenar, analizar y visualizar datos geospaciales o geográficos. Combina mapas digitales con bases de datos, permitiendo la representación y estudio de información vinculada a ubicaciones específicas en la superficie terrestre. Los SIG son esenciales en áreas como urbanismo, gestión ambiental, transporte, agricultura y planificación de infraestructuras, ya que facilitan la toma de decisiones basada en datos espaciales precisos. Además, permiten realizar análisis complejos, como modelado de terreno, evaluación de riesgos naturales y monitoreo de cambios ambientales. Su capacidad para integrar información de diversas fuentes, como imágenes satelitales y sensores remotos, aumenta su utilidad. Los SIG son herramientas clave para el desarrollo sostenible y la gestión eficiente de recursos, fomentando un enfoque basado en datos para resolver problemas espaciales [9]. En la presente investigación se utilizó el SIG Idrisi Selva 17.0 ®, para visualizar los mapas del experimento y como punto de comparación con lo desarrollado en C#.

2.1.3 Mapas y su estructura

Un mapa raster contiene información en un formato de malla de dos dimensiones, cada región del mapa está formada por un conjunto de píxeles, donde cada uno de ellos tiene una coordenada (x, y) y un valor numérico que va desde un conjunto de 8 bits a uno de 32 bits por píxel, esto eleva considerablemente la necesidad de espacio para almacenar su información [10]. Un mapa raster debe interpretarse a través de un segundo archivo, tiene extensión *.RDC (Raster Data Configuration) es básicamente un archivo de texto plano que contiene la información necesaria para importar, a una matriz, la información contenida en el mapa. A continuación (Figura 1), se muestra un ejemplo de mapa raster “abc.rst”, y con el uso del comando `type <nombre de archivo>` desde la consola en un sistema Windows 9, se muestra el RDC que le corresponde.

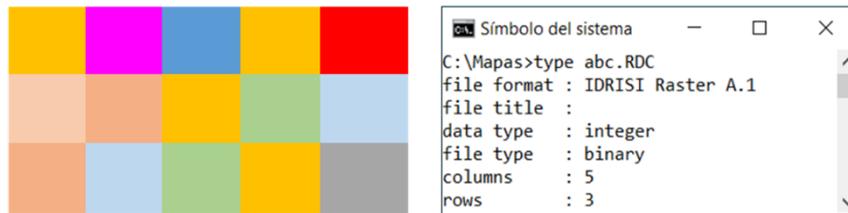


Figura 1. Mapa raster “abc.rst” y su par “abc.RDC”, cada celda del raster representa un píxel.

Se pueden apreciar 4 propiedades fundamentales para la apertura del mapa:

- Data type: Tipo de dato contenido en el archivo, puede ser Integer, Byte o Real
- File Type: El tipo de archivo
- Columns y Rows: El total de columnas y filas que componen el mapa

Otro aspecto técnico a tomar en cuenta es que, al ser valores enteros de 16 bits para cada píxel, estos se encuentran divididos en dos bytes (8 bits por cada byte), por tanto, se deberá leer una secuencia total de 16 bits para después convertirlos a decimal. En el mapa “abc.rst” (Figura 2) se muestra el valor para cada píxel del mapa, y enseguida (Figura 3) se muestra la secuencia de lectura en donde los primeros dos bytes forman el valor de 10 decimal, correspondiente al píxel de la primera fila y columna del mapa (en una matriz esto equivaldría a la celda [0,0]), y el 5to par de bytes con valor de 1720 decimal, corresponden al último píxel de la primera fila del mapa (celda [0,4] en una matriz).

10	15	9	10	1720
1	14	10	11	13
14	13	11	10	12

Figura 2. Mapa raster “abc.rst” con los valores enteros para cada píxel

```

Archivo binario:

00000000000010100000000000011110000000000001001
00000000000010100000111010111000000000000000001
000000000000111000000000000010100000000000001011
000000000000110100000000000011100000000000001101
000000000000101100000000000010100000000000001100

10 = 0000000000001010
1720 = 0000011010111000
    
```

Figura 3. Valores binarios del archivo.

Con estas propiedades se puede observar que, para el caso de la Figura 1, los datos de cada pixel del mapa son del tipo integer (entero de 16 bits), el archivo que contiene el mapa es de tipo binario y que el mapa raster consta de 3 filas y 5 columnas. Se debe tener claro que al abrir el archivo binario, atendiendo lo indicado en el RDC, los datos se encontrarán de manera secuencial, es importante tomar en cuenta las filas y las columnas para construir la matriz y depositar en ella cada uno de los valores de los pixeles.

En la presente investigación, los experimentos se llevaron a cabo utilizando un mapa raster que representa 9 zonas de uso urbano, de la ciudad de Culiacán Sinaloa, para el año 2004 (Figura 4) y su par RDC (Figura 5), en donde se pueden apreciar los parámetros que se usarán para la apertura del mapa.

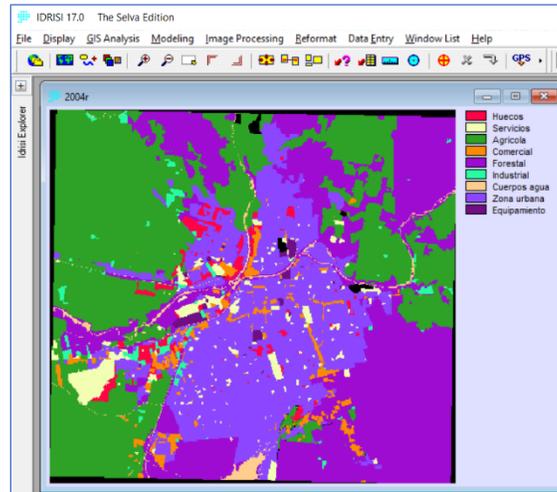


Figura 4. Vista de un mapa raster (archive 2004r.rst) utilizando el SIG IDRISI selva versión 17.0

El par RDC de la Figura 5 muestra que, el tipo de dato a interpretar es un valor entero (de 16 bits), el tipo de archivo es binario y el total de filas y columnas del mapa es de 366 filas x 397 columnas.

```

C:\Mapas>type 2004r.RDC
file format : IDRISI Raster A.1
file title :
data type : integer
file type : binary
columns : 397
rows : 366
    
```

Figura 5. Datos del RDC del mapa raster 2004r.rst

2.2 Métodos

2.2.1 Lectura y visualización del mapa

Se utilizaron principalmente las siguientes clases para el manejo e interpretación de archivos:

- `StreamReader`: Lee caracteres de un flujo de datos.
- `FileStream`: Permite leer y escribir bytes en un archivo.
- `BinaryReader`: Lee datos binarios desde un flujo de datos.
- `BitConverter`: Convierte tipos de datos primitivos a arrays de bytes.
- `Bitmap`: Representa una imagen de mapa de bits en memoria.
- `PictureBox`: Control de Windows Forms para mostrar imágenes.

En conjunto, las herramientas y métodos mencionados fueron utilizados para llevar a cabo los experimentos y elaborar los procedimientos necesarios.

3 RESULTADOS

3.1 Metodología

Como se observa en el apartado 2.1.3, el primer paso para la lectura de un mapa es verificar su RDC y sus 4 propiedades principales. Después, es necesario construir una matriz dados los valores del archivo RDC, además de respetar el orden de lectura de los valores binarios para posteriormente almacenarlos en la matriz. En este orden de ideas, se presenta a continuación (Figura 6) un algoritmo como componente metodológico, con las instrucciones necesarias para realizar la lectura y vaciado del mapa en una matriz.

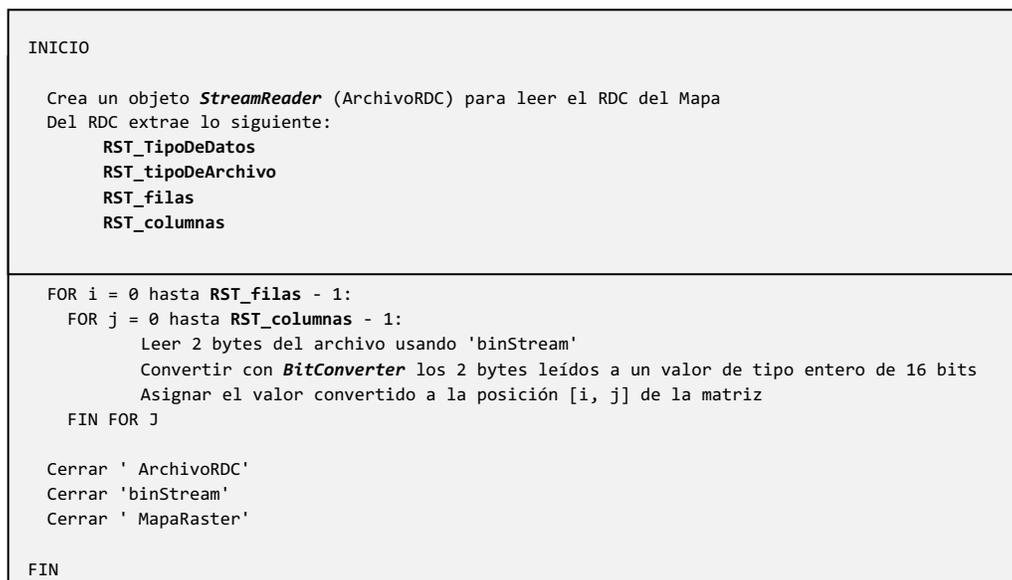


Figura 6. Algoritmo para la apertura del raster y su carga en una matriz.

Continuando con la explicación del algoritmo, una vez que se extraen datos necesarios del RDC con `StreamReader` para la creación de la matriz, se crea un objeto `FileStream` llamado `MapaRaster` que abre el archivo en modo solo lectura, se verifica el tipo de archivo (debe ser binario), y a continuación se crea un objeto `BinaryReader` (`binStream`) para leer datos binarios desde `MapaRaster`. Con un par de ciclos anidados se recorre la matriz por fila y columna, en cada iteración se leen 2 bytes del archivo, con `BitConverter` se convierten esos 2 bytes en un número entero de 16 bits y se almacena el resultado en la posición correspondiente de la matriz. Al finalizar los ciclos se cierran todos los flujos de datos.

Para la visualización del mapa en un área determinada (Figura 7), específicamente en un PictureBox, se crea el Bitmap con el mismo tamaño del mapa raster, se recorre la matriz celda por celda, y según su valor, se le asignará un color determinado al pixel.

```
INICIO
  mapaBit ← CrearBitmap(RST_columnas, RST_filas)

  FOR i DESDE 0 HASTA RST_filas - 1
    FOR j DESDE 0 HASTA RST_columnas - 1
      switch mapaMatrizInt[i, j]
        CASO 0: SetPixel(mapaBit, j, i, Negro)
        CASO 1: SetPixel(mapaBit, j, i, Rojo)
        CASO 2: SetPixel(mapaBit, j, i, Amarillo)
        CASO 3: SetPixel(mapaBit, j, i, Verde)
        CASO 4: SetPixel(mapaBit, j, i, Naranja)
        CASO 5: SetPixel(mapaBit, j, i, Morado)
        CASO 6: SetPixel(mapaBit, j, i, Aqua)
        CASO 7: SetPixel(mapaBit, j, i, Café)
        CASO 8: SetPixel(mapaBit, j, i, Azul)
        CASO 9: SetPixel(mapaBit, j, i, Morado)
        OTRO CASO: SetPixel(mapaBit, j, i, Negro)
      FIN switch
    FIN FOR j

  pictureBox1.Tamaño ← ObtenerTamaño(mapaBit)
  pictureBox1.Imagen ← mapaBit

FIN
```

Figura 7. Algoritmo para la visualización de un raster en un PictureBox.

Teniendo en cuenta que los valores posibles para el tipo de dato de un mapa raster pueden ser **integer**, **byte** o **real**, es necesario definir una paleta de colores para interpretar cada uno de los valores o un rango de ellos. Cabe señalar que, en el algoritmo de visualización se utilizaron solamente 11 colores, debido a que eran suficientes para representar las 9 zonas urbanas presentes en el mapa del experimento, sin embargo el catálogo podría expandirse tanto como se ocupe.

Por último, el seguimiento a estos algoritmos es fundamental en el procesamiento de datos geoespaciales, asegurando la conversión de datos binarios a una representación matricial que se puede analizar o visualizar posteriormente. Para comprobar que el mapa haya sido correctamente leído y visualizado en C#, se debe comparar con el mismo mapa, pero ahora mostrado con IDRISI Selva.

3.2 Visualización de mapas

Para la elaboración de los experimentos, se desarrolló un software (Mapixia Beta) siguiendo lo descrito en los algoritmos del apartado anterior, escrito en lenguaje C#, fungió como área de pruebas para la apertura de distintos mapas. A continuación (Figura 8) se muestra el mapa raster de Culiacán 2004, a la izquierda en Mapixia Beta y a la derecha en Idrisi Selva. Visualmente se puede apreciar la similitud de ambos mapas.

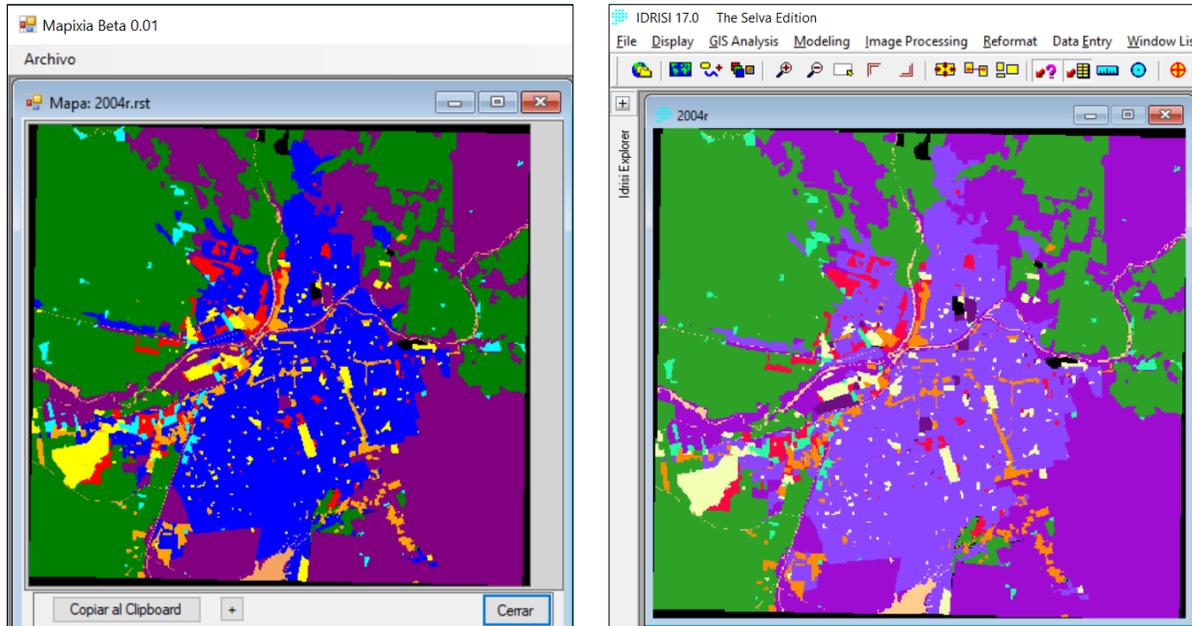


Figura 8. Comparativa del mapa raster de Culiacán 2004.

De acuerdo al RDC presentado en la Figura 5, en ambos mapas se procesaron un total de 145,302 píxeles, categorizados en 9 zonas identificadas por color, en píxeles con valor CERO (por lo regular esto significa que no hay información presente para ese píxel) se imitó al SIG Idrisi asignando el color negro en esos casos. Se puede apreciar que Mapixia Beta muestra un mapa idéntico al mostrado con Idrisi Selva, con esto se puede decir que los algoritmos implementados funcionan como se esperaba.

La siguiente prueba se realizó con el mapa de 1997 para la misma zona urbana de Culiacán (Figura 9), el RDC de este mapa (Figura 10) muestra que el tipo de dato es entero, tipo binario y cuenta con la misma cantidad de filas y columnas del mapa del 2004 y, por ende, los mismos píxeles.

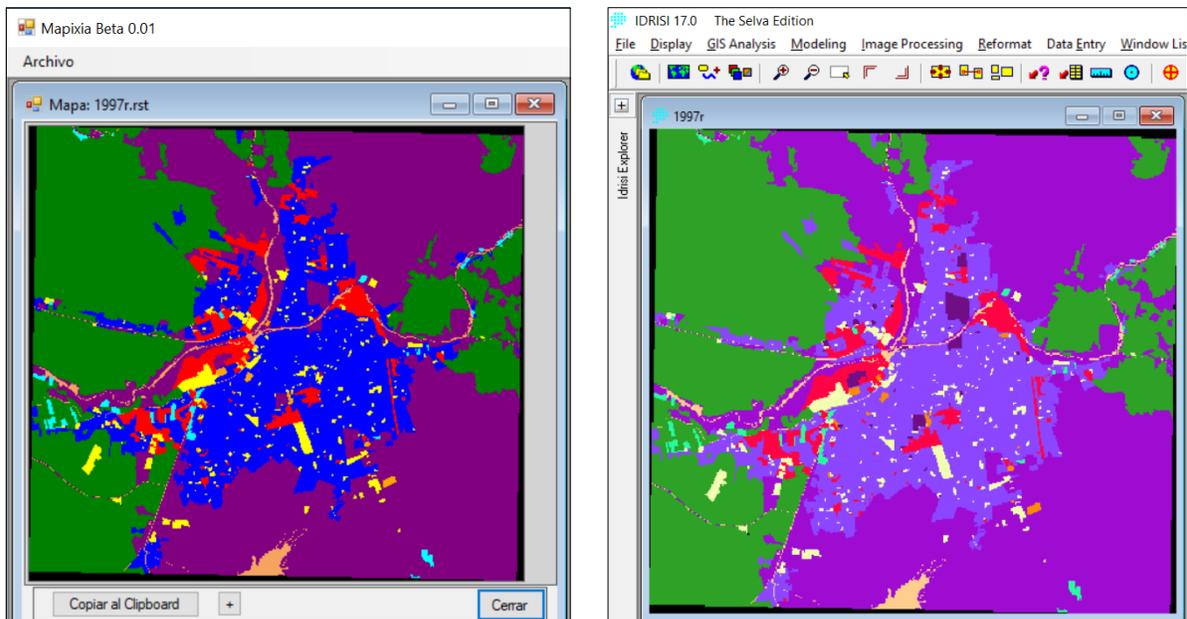


Figura 9. Comparativa del mapa raster de Culiacán 1997.

```

C:\Mapas\final2>type 1997r.RDC
file format : IDRISI Raster A.1
file title :
data type : integer
file type : binary
columns : 397
rows : 366
    
```

Figura 10. Datos del RDC del mapa raster 1997r.rst

La última prueba se realizó con un mapa del 2004, con tipo de datos Byte, es un archivo que maneja solo dos valores 0 y 1. En su RDC (Figura 11) se observan los datos referentes al mapa. Este tipo de mapas son utilizados como “mascaras” para aislar ciertos tipos de zonas de otras. Se aprecian solo dos colores (Figura 12) y de nuevo, visualmente similares en ambas versiones.

```

C:\Mapas\final>type 2004mask.RDC
file format : IDRISI Raster A.1
file title :
data type : byte
file type : binary
columns : 397
rows : 366
    
```

Figura 11. Datos del RDC del mapa raster 2004mask.rst

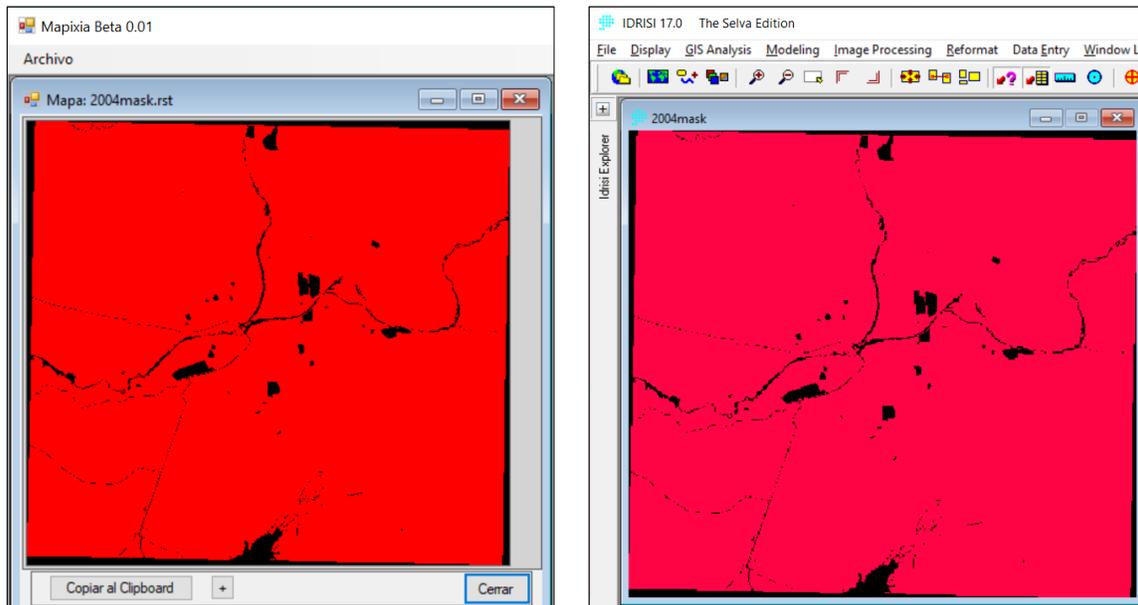


Figura 12. Comparativa del mapa raster de tipo Byte.

Como se puede apreciar, cada uno de los mapas raster mostrados en el software Mapixia Beta, son idénticos a los mostrados en el software Idrisi Selva. La escritura del programa a partir de los algoritmos presentados, resultaron en una apertura y visualización exitosa en cada uno de los mapas del experimento.

4 CONCLUSIONES

La metodología desarrollada e implementada funcionó en cada mapa del experimento, la similitud de los mapas mostrados entre el software desarrollado y el profesional fue idéntica pixel a pixel. Sin embargo, solo se procesaron mapas de tipo entero y byte, hace falta procesar los de tipo real, y no es porque no se puedan abrir, sino que la paleta de colores necesita una atención especial al ser valores continuos y no discretos, queda como trabajo futuro la implementación de los mapas con tipo de dato real, en donde se trabaje tanto en la carga del archivo como en la confección de una paleta de colores con un rango continuo para visualizar correctamente ese tipo de mapas. Mientras tanto, el uso de la metodología de apertura y visualización lograda, representa un avance en el trabajo de procesamiento de mapas con herramientas propias que permitan un manejo profundo y a detalle de los datos geoespaciales, esto permitirá trabajar con ellos en cualquier otro ámbito, y sin limitantes.

REFERENCIAS

- [1] S. Berling-Wolff y J. Wu, «Modeling urban landscape dynamics: A review», *Ecological Research*, vol. 19, pp. 119-129, 2004.
- [2] R. White, G. Engelen y I. Uljee, *Modeling Cities and Regions as Complex Systems*, MIT Press, 2015.
- [3] M. N. DeMers, *GIS Modeling in Raster*, Pearson, 2001, pp. 21-25.
- [4] K.-t. Chang, *Introduction to Geographic Information Systems*, McGraw Hill, 2018.
- [5] R. R. Zamora, A. P. Garzón y A. S. Hernández, «A Dynamic System Model Using Agile Method Techniques Applied to Geomatics», *Communications in Computer and Information Science*, vol. 1053, pp. 71-82, 2019.
- [6] G. O. d. Cos y E. M. Latorre, «EVALUACIÓN MULTICRITERIO Y DELIMITACIÓN DE ESPACIOS FUNCIONALES: APLICACIÓN SIG PARA LA DEFINICIÓN DE MAPAS COMARCALES,» *GeoFocus*, nº 7, pp. 256-280, 2007.
- [7] L. Humacata, «Análisis espacial de los cambios de usos del suelo. Aplicación con Sistemas de Información Geográfica,» *Revista cartográfica*, vol. 98, pp. 239-257, 2022.
- [8] Microsoft, «Microsoft Learn,» [En línea]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>. [Último acceso: 21 10 2024].
- [9] I. O. Pastor, E. O. Pérez, B. M. Ramos y A. E. Canalejo, *Sistemas de informacion geografica. Teoria y practica*, Dextra, 2016.
- [10] S. Wise, *GIS Fundamentals*, CRC Press, 2021.