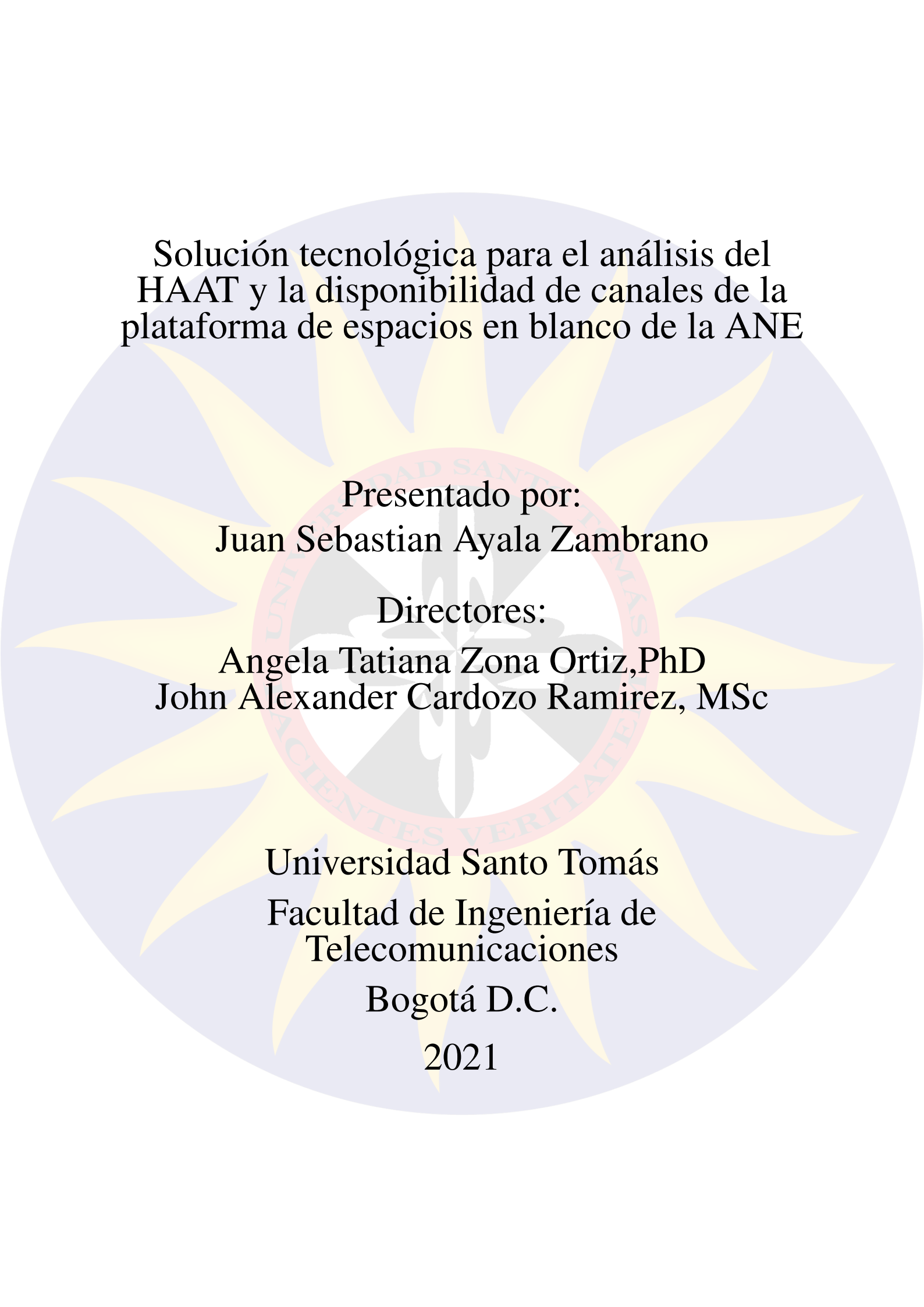


Solución tecnológica para el análisis del
HAAT y la disponibilidad de canales de la
plataforma de espacios en blanco de la ANE

Universidad Santo Tomás
Facultad de Ingeniería de
Telecomunicaciones

Bogotá D.C.

2021



Solución tecnológica para el análisis del HAAT y la disponibilidad de canales de la plataforma de espacios en blanco de la ANE

Presentado por:

Juan Sebastian Ayala Zambrano

Directores:

Angela Tatiana Zona Ortiz, PhD

John Alexander Cardozo Ramirez, MSc

Universidad Santo Tomás

Facultad de Ingeniería de
Telecomunicaciones

Bogotá D.C.

2021

Agradecimientos

Agradezco a la Agencia Nacional del Espectro (ANE) por el financiamiento para la obtención de los resultados presentados en este trabajo, en el marco del convenio especial de cooperación N° 64 -2020 celebrado entre la ANE y la Universidad Santo Tomás. A mis tutores, especialmente a la Doctora Tatiana Zona, quien me brindó las herramientas para llevar a cabo el desarrollo de este proyecto.

Dedicatoria

A mis padres,
quienes me han brindado todo su apoyo
y cariño.

Índice general

Introducción	9
1. Presentación del proyecto	11
1.1. Planteamiento del problema	11
1.2. Justificación	13
1.3. Objetivos	14
1.4. Marco Teórico	15
1.4.1. Ráster	15
1.4.2. Bases de datos NoSQL	17
2. Acondicionamiento de los datos	18
2.1. Ráster del módulo de cálculo	18
2.2. Shapefile del módulo de cálculo	21
2.3. Remuestreo	23
2.4. Recorte	28
2.5. Reproyección	34
3. Diseño de la base de datos	37
3.1. Motor de base de datos	38
3.2. Estructura de la base de datos	38
3.3. Construcción de la base de datos	41
3.4. Automatización	45
4. Procesamiento de los datos	48
4.1. Multiproceso	48
4.2. Uso del módulo de cálculo	50
4.2.1. Cálculo del sitio	52
4.2.2. Cálculo del HAAT	53
4.2.3. Cálculo del Radio Cocanal (CCR)	54
4.2.4. Cálculo de la lista de canales	55
5. Construcción de los mapas	57

6. Elaboración de los conceptos técnicos	65
7. Conclusiones y Producción	74
Referencias	77

Índice de figuras

2.1. Ráster módulo de cálculo.	19
2.2. Tamaño del píxel en metros.	19
2.3. Propiedades del ráster.	20
2.4. Dimensiones del ráster.	20
2.5. Shapefile de Colombia.	21
2.6. Propiedades del Shapefile de Colombia.	22
2.7. Campos del Shapefile de Colombia.	22
2.8. Polígonos clase 1 y 2.	23
2.9. Polígono clase 3.	23
2.10. Remuestreo de un ráster <i>Tomado de: https://gisgeography.com/raster-resampling/</i>	24
2.11. Método de remuestreo: Vecino más cercano. <i>Adaptado de: The Importance of Scale in Land Use Models: Experiments in Data Conversion, Data Resampling, Resolution and Neighborhood Extent</i>	25
2.12. Ubicación del mapa.	26
2.13. Salida de la instrucción de remuestreo.	26
2.14. Nuevo archivo en el directorio del ráster.	27
2.15. Ráster remuestreado.	27
2.16. Propiedades del ráster remuestreado.	27
2.17. Dimensiones del ráster remuestreado.	28
2.18. Tamaño del píxel en metros del ráster remuestreado.	28
2.19. Ráster con contorno de Colombia	29
2.20. Departamento de San Andrés y Providencia.	30
2.21. Carpeta shape.	30
2.22. Nuevo shapefile.	31
2.23. Propiedades del nuevo shapefile.	31
2.24. Salida del comando recorte.	32
2.25. Ráster recortado.	33
2.26. Propiedades del ráster recortado.	33
2.27. Sistema de coordenadas del ráster de elevación.	34

2.28. Reproyección de un ráster. <i>Tomado de: https://cartographicperspectives.org/index.php/journal/article/view/cp71-finn-et-al/html</i>	34
2.29. Sistema de coordenadas XYZ.	35
2.30. Salida de la instrucción de reproyección.	36
2.31. Archivo CSV generado.	36
2.32. Cantidad de líneas del archivo CSV generado.	36
3.1. Separador del archivo generado por la instrucción de reproyección.	42
3.2. Separador del archivo posterior a la instrucción del editor de flujo.	43
3.3. Ejecución de la instrucción “MongoImport”.	44
3.4. Base de datos creada.	44
3.5. Colección creada.	44
3.6. Estructura de la colección creada.	44
3.7. Cantidad de documentos de la colección creada.	45
3.8. Ráster de montería.	46
3.9. Propiedades del ráster de montería.	46
3.10. Ejecución herramienta “tif_to_mongo”.	46
3.11. Creación de la base de datos (Montería).	47
3.12. Creación de la colección “locations” (Montería).	47
3.13. Estructura de la colección “locations” (Montería).	47
3.14. Cantidad de documentos de la colección “locations” (Montería).	47
4.1. Flujo del hilo principal.	49
4.2. Flujo de los hilos secundarios.	49
4.3. Arquitectura del multiproceso simétrico.	50
4.4. Flujo del módulo de cálculo.	50
4.5. Cantidad de documentos de la colección “place_queue”.	52
4.6. Estructura de la cola de sitio.	52
4.7. Estructura de la colección de salida etapa 1.	53
4.8. Anillo sobre raster.	53
4.9. Cantidad de documentos de la colección “haat_queue”.	54
4.10. Estructura de la colección de salida etapa 2.	54
4.11. Cantidad de documentos de la colección “ccr_queue”.	55

4.12. Estructura de la colección de salida etapa 3.	55
4.13. Ejemplo de lista de canales disponibles.	55
4.14. Cantidad de documentos de la colección “channels_queue”.	56
4.15. Estructura de la colección de salida etapa 4.	56
5.1. Estructura CSV driver XYZ.	57
5.2. Sentido de ordenamiento de un archivo CSV driver XYZ.	58
5.3. Colección de datos agrupados y ordenados.	59
5.4. Salida de mongoexport.	60
5.5. Archivo CSV entregado por MongoDB.	60
5.6. Salida de la instrucción translate.	61
5.7. Raster de HAAT de Colombia.	61
5.8. Propiedades del Ráster de HAAT de Colombia.	62
5.9. Salida de la herramienta mongo_to_tif.	62
5.10. Ráster de cantidad de canales para el método uno de Colombia.	63
5.11. Propiedades del Ráster de cantidad de canales para método uno de Colombia.	63
5.12. Ráster de cantidad de canales para el método dos de Colombia.	64
5.13. Propiedades del Ráster de cantidad de canales para método dos de Colombia.	64
6.1. Estructura CSV haat_stats departamento.	66
6.2. Estructura CSV haat_stats municipio.	66
6.3. Plantilla para departamento página 1.	67
6.4. Plantilla departamento página 2.	68
6.5. Plantilla departamento página 3.	69
6.6. Ejemplo de tabla de departamento.	70
6.7. Carpeta con informe de un departamento.	70
6.8. Carpeta con las gráficas de un departamento.	70
6.9. Tabla del departamento Amazonas.	71
6.10. Plantilla de municipio.	71
6.11. Reporte de municipio El Encanto.	72
6.12. Flujo macro principal.	73
6.13. Cantidad de paginas concepto técnico HAAT.	73
6.14. Cantidad de paginas concepto técnico método uno.	73

6.15. Cantidad de paginas concepto técnico método dos.	73
--	----

Índice de tablas

2.1. Parámetros de entrada: Instrucción remuestreo.	26
2.2. Parámetros de entrada: Instrucción filtro.	31
2.3. Parámetros de entrada: Instrucción recorte.	32
2.4. Parámetros de entrada: Instrucción remuestreo.	35
3.1. Parámetros de entrada: Instrucción de importación.	43
5.1. Parámetros de entrada: Instrucción de exportación.	59
5.2. Parámetros de entrada: Instrucción de reproyección.	61
5.3. Parámetros de entrada: Instrucción de conversión de MongoDB a Tif.	62
7.1. Registros de software realizados.	76

Índice de Códigos

2.1. Instrucción de remuestreo.	26
2.2. Instrucción de filtro.	30
2.3. Instrucción de recorte.	32
2.4. Instrucción de reproyección.	35
3.1. Colección “locations”	39
3.2. Colección “calculated_locations”	39
3.3. Objeto “place”	40
3.4. Objeto “haat”	40
3.5. Objeto “met_n”	41
3.6. Instrucción sed para cambiar el separador por coma.	42
3.7. Instrucción para importar los datos a MongoDB.	43
3.8. Herramienta “tif_to_mongo” convertir de ráster a base de datos MongoDB.	46
5.1. Comandos para agrupar y ordenar los datos	58
5.2. Instrucción para exportar los datos de MongoDB.	59
5.3. Instrucción sed con múltiples parámetros de ejecución.	60
5.4. Instrucción de reproyección.	60
5.5. Herramienta mongo_to_tif.	62

Introducción

Actualmente en nuestra sociedad, el acceso a internet por parte de los individuos se ha convertido en una necesidad, ya sea para mantenerse informado de lo que pasa en el mundo, acceder a la educación y a diversas fuentes de información, como medio de ocio y entretenimiento. En el caso de Colombia, aproximadamente el sesenta por ciento (60 %) de la población dispone de una conexión a la red[9], esto quiere decir que una gran cantidad de colombianos aún carecen de conectividad a internet.

Una gran parte de la población desconectada se encuentra ubicada en las zonas rurales del país, esto se debe a lo complicado y poco rentable que es para los operadores de internet desplegar la infraestructura necesaria. Con el objetivo de dinamizar el uso del espectro, la Agencia Nacional del Espectro (ANE) en el capítulo tres la resolución 105 de 2020 establece los requerimientos para el uso de los canales no empleados en la banda de televisión para servicios de conectividad rural.

En varios países ya se encuentra implementado el uso de espacios en blanco en televisión (TVWS) para conectividad. En el caso de Colombia, la Agencia Nacional del Espectro en cooperación con la Universidad Santo Tomás diseñó e implementó la plataforma de espacios en blanco en televisión (BDEB). Esta plataforma se comunica con los dispositivos TVWS haciendo uso del protocolo PAWS, el objetivo es entregar al dispositivo TVWS una lista de canales disponibles en la banda de televisión, según su ubicación geográfica. En esta plataforma se cuenta con la información del espectro disponible a nivel nacional de la banda de televisión en cada uno de los municipios del país, además cuenta con módulos que permiten a la ANE hacer vigilancia y control sobre el uso del espectro que realizan los dispositivos TVWS.

Teniendo en cuenta que la intención de la cooperación es habilitar el uso de los espacios en blanco a nivel nacional, sin interferir el servicio de televisión, surge la necesidad de conocer la disponibilidad de canales para determinar la viabilidad del despliegue de la tecnología TVWS sobre el territorio nacional. Generar los insumos necesarios para el análisis de disponibilidad en todo el territorio nacional implica una alta demanda de tiempo de procesamiento y una gran cantidad de consultas geográficas.

En el presente documento se presentan el planteamiento y la implementación de los procesos necesarios en cuanto a infraestructura, entorno de desarrollo y tratamiento de los datos, para el análisis de disponibilidad de TVWS sobre el territorio nacional. Estas soluciones utilizan el módulo de cálculo de la plataforma de espacios en blanco en televisión para el cálculo del HAAT y la lista de canales disponibles. El documento también muestra el proceso de construcción de la base de datos con la información consultada y que será necesaria para la construcción de los mapas georeferenciados. Estos datos son el insumo para la elaboración de reportes automatizados a nivel de departamento y municipio, con la información HAAT, número de canales disponibles en el método uno y en el método dos. En la parte final del documento además de mostrar las conclusiones, se listan los registros de software relacionados con las soluciones presentadas.

Capítulo 1

Presentación del proyecto

1.1. Planteamiento del problema

En las zonas rurales de nuestro país, generalmente apartadas de las zonas conectadas, es notable la falta de recursos, la falta de inversión por parte del estado y el desinterés por parte de los operadores en cuanto al despliegue de infraestructura de conectividad. Esta situación se refleja en la brecha digital que hoy en día tiene Colombia[6].

El cuadro nacional de atribuciones de bandas de frecuencias (CNABF) cuenta con una serie de canales de radiofrecuencia disponibles en la banda de televisión, los cuales no están siendo utilizados por los proveedores de servicios de televisión. En los últimos años, la ANE se ha esforzado en dinamizar el uso del espectro, ya que es un recurso limitado y el medio para la conectividad inalámbrica, uno de estos esfuerzos es habilitar el uso de los canales de televisión no usados para conectividad mediante la tecnología TVWS .

Actualmente, la Agencia Nacional del Espectro dispone de una plataforma de espacios en blanco en televisión, que le permite conocer el uso del espectro que hacen los dispositivos TVWS en Colombia y dar una lista de canales disponibles que garantice la no interferencia de los servicios de TV. Con el fin de tomar decisiones adecuadas en cuanto a regulación y enfocándose en aumentar la disponibilidad de canales en zonas rurales del territorio colombiano, surge la necesidad de realizar un análisis de disponibilidad de canales a nivel nacional que según la asignación de frecuencias en canales de televisión y el límite de HAAT de 800m[1] permita visualizar las disponibilidad de canales sobre el territorio nacional.

Uno de los módulos de la plataforma de espacios en blanco en televisión es el módulo de cálculo, este módulo se encarga determinar la lista de canales disponibles para una ubicación especificada por sus coordenadas geográficas (latitud, longitud) según las reglas de negocio establecidas para los métodos de calculo [11] [12]. Dicho módulo dispone de diferentes rutinas, entre ellas la de cálculo del HAAT y el cálculo de lista de canales disponibles, las cuales com-

ponen el flujo de ejecución mostrado en la figura 4.4.

El análisis de la disponibilidad de TVWS a nivel nacional requiere conocer el valor de HAAT y el número de canales disponibles para los dos métodos de cálculo a lo largo del territorio colombiano, es decir, para cada punto geográfico de Colombia. Toda esa información debe condensarse en tres documentos de **Concepto Técnico**, que están acompañados de un ráster con la información. Esto implica la consulta de más de 1278 millones de puntos, si el ráster a construir es de 50m de resolución. Además cada consulta al módulo de cálculo tarda entre 0.5 y 1.0 segundos en procesarse.

Ahora bien, teniendo en cuenta que el módulo de cálculo dispone de dos métodos para calcular lista de canales, se debe consultar como mínimo dos veces por cada coordenada geográfica. Asimismo, para la construcción del ráster es importante hallar una manera efectiva de estructurar y visualizar los datos del país. Adicionalmente, considerando un reporte de una página para cada municipio y departamento de Colombia, en total la cantidad de páginas por concepto técnico sería de 1135 como mínimo. Teniendo en cuenta que se proponen tres conceptos técnicos, se deberían escribir más de 3405 páginas lo cual es inviable de manera manual con el tiempo disponible para la tarea dentro del convenio con la ANE.

De acuerdo a la problemática mencionada anteriormente, se plantea la siguiente pregunta:

¿Cómo construir un banco de datos utilizando el módulo de cálculo de la plataforma de espacios en blanco que permita generar los tres conceptos técnicos con un análisis a nivel municipal, departamental y nacional, y la visualización para el estado actual de la disponibilidad de canales para uso de TVWS en Colombia?

1.2. Justificación

La justificación para la creación de soluciones tecnológicas que apoyen la creación de los tres conceptos técnicos del convenio de cooperación con la ANE, tiene tres pilares según el problema planteado: la obtención de los datos, la escritura de los conceptos técnicos, y la visualización de los datos en un ráster.

Para la obtención de los datos necesarios para la construcción manual de los tres conceptos técnicos que permitirán el análisis a nivel municipal, departamental y nacional de la disponibilidad de canales para uso de espacios en blanco en todo el territorio nacional, serían necesarios 40 años, esto teniendo en cuenta las 1278 millones de consultas que durarían 1 segundo cada una. Por lo tanto es necesario no solo optimizar el número de consultas, si no mejorar la infraestructura para realizar un procesamiento en paralelo, que permita la reducción considerable del tiempo necesario para la obtención de información.

Por otro lado, el número de páginas necesarias para la construcción manual de los tres conceptos técnicos que asciende a más 3405 suponiendo un equipo de diez personas durante diez días hábiles requeriría la escritura de 34 páginas diarias por persona, teniendo en cuenta que el análisis estadístico a presentar, es similar entre los departamentos y entre los municipios, se puede llegar a una solución que permita la automatización en la presentación de la información dentro de los conceptos técnicos.

Finalmente, para la visualización de toda la información en un raster, es necesario diseñar no solamente una estructura de base de datos adecuada, si no toda una solución tecnológica que permita el ordenamiento de los datos para la correcta construcción del ráster.

1.3. Objetivos

El objetivo general del presente proyecto es implementar una o varias soluciones de software que suministren los insumos necesarios para la realización de los conceptos técnicos requeridos por la ANE en el convenio especial de cooperación No. 64 de 2020.

Para alcanzar este objetivo general se han propuesto 5 objetivos específicos, a saber:

- Establecer un procedimiento que permita disminuir el número de consultas a realizar al módulo de cálculo y por lo tanto el volumen de datos a trabajar, teniendo en cuenta la cartografía utilizada por el módulo de cálculo.
- Diseñar una estructura de datos que permita su análisis y visualización, con limitaciones en cuanto al volumen de datos, velocidad de acceso, que permita la adaptabilidad de la información.
- Establecer los mecanismos necesarios para la consulta al módulo de cálculo, con el fin de minimizar la pérdida de información y el tiempo de obtención de la totalidad de los datos.
- Visualizar geográficamente la información de la base de datos relacionada con el HAAT y la lista de canales disponibles para cada método.
- Generar los documentos de conceptos técnicos que muestren la información de HAAT y lista de canales disponibles para cada método en cada uno de los departamentos y municipios de Colombia.

1.4. Marco Teórico

Los conceptos técnicos a entregar a la ANE contemplan la construcción de un ráster, y este necesita el ordenamiento de unos datos consultados al módulo de cálculo. Por lo anterior, en este apartado se habla de que es un ráster y como se organizan los datos dentro del mismo, además se habla sobre qué es una base de datos NoSQL, que es donde se almacenarán los datos consultados al módulo de cálculo.

Sin embargo, no se hace una introducción del funcionamiento del módulo de cálculo específicamente, esto se podrá consultar en los entregables [11] y [12] del convenio de cooperación N° 64 de 2020. En cuanto a la tecnología TVWS se recomienda la lectura de “Huecos espectrales en televisión” donde se explica qué es un hueco espectral y cómo está regulado en Colombia[5].

1.4.1. Ráster

Las estructuras de datos tipo ráster son una subcategoría de las estructuras de tipo tabular. La RAE define tabular como expresar valores, magnitudes u otros datos por medio de tablas[4]. Los datos tabulares, los cuales se utilizan para almacenar atributos o información descriptiva sobre datos netamente espaciales tienen atributos. Los atributos pueden almacenarse en una tabla junto con la información espacial o pueden existir en tablas separadas que pueden vincularse o relacionarse con el conjunto de datos espaciales a través de un identificador único.

Los datos tabulares son usados dentro de paquetes de software con un sistema de información geográfica permitiendo de esta manera el almacenamiento de archivos de texto delimitados, archivos dBASE, archivos de Microsoft Excel. Dichas tablas tienen la propiedad de importar más paquetes de software para una gestión de base de datos más robusta, por ejemplo Oracle.

Según el material didáctico de aprendizaje a distancia de la infraestructura de información geográfica para el análisis espacial y modelización, en un modelo de datos ráster, los datos están representados por una superficie dividida en una cuadrícula de celdas o píxeles de tamaño regular[4]. Típicamente, una trama es una cuadrícula que tiene un origen (esquina superior izquierda) y la ubicación de cada píxel está definida por este origen y un desplazamiento. Una cuadrícula de datos es generalmente una forma rectangular y su tamaño está definido por un

número de filas y columnas.

Aunque una amplia variedad de formas de trama son posibles (por ejemplo, triángulos, hexágonos) generalmente se usa una serie de rectángulos, o cuadrados, llamados celdas de cuadrícula o píxeles. Cada píxel almacena un atributo o valor relacionado con una porción de la superficie de la tierra. Un modelo de datos raster es ideal para almacenar información relacionada con características continuas (características que no son espacialmente discretas) por ejemplo datos de temperatura o elevación. Imágenes como, fotos aéreas e imágenes satelitales también se almacenan en una estructura de datos simple basada en filas y columnas vinculadas a ubicaciones de píxel fijas.

Dentro de sus características, los ráster tienen un sistema referencia coordenado (SRC), también llamado proyección, esta depende de la zona geográfica que represente el raster. La proyección permite realizar aproximaciones visuales más exactas, debido a que la concavidad de la tierra es variable a lo largo de su extensión. Los ráster permiten hacer ciertos tipos de operaciones que ayudan a su adaptación dependiendo las necesidades del usuario, esta información se puede ver ampliada en el capítulo 2.

1.4.2. Bases de datos NoSQL

Los sistemas relacionales hoy en día tienen problemas a la hora de gestionar grandes datos como los que se encuentran en un ráster. El término “NoSQL” existe desde 1998. Carlo Strozzi nombró una base de datos de código abierto “NoSQL” para dejar claro que su proyecto no admite ninguna interfaz SQL[3]. La implementación de gestores de información tomaron rápidamente la función de herramienta para brindar soluciones. La posibilidad de usar más nodos para aumentar la potencia de procesamiento cuando habían numerosas cantidades de datos fueron claves para para aventajar computadores con recursos limitados.

Las bases de datos NoSQL se caracterizan por tener un enfoque no relacional, en mayor medida son productos de código abierto, es decir que no necesitan un esquema definido, proporcionan una API para la integración en otros productos de software, utilizan una arquitectura descentralizada para la fácil replicación de los datos y siguen el principio BASE donde el sistema es dinámico pero consistente eventualmente con el cambio del tiempo.

En 2012, Baas introduce las bases de datos almacenadas en gráficos Neo4j incluyendo una extensión espacial que admite todos los tipos de geometría de características simples que pueden utilizarse para análisis de rutas o búsquedas de proximidad [2]. Lo cual permite utilizar las bases de datos NoSQL para gestionar los desafíos típicos del big data, ya sea espacial o no: volumen, variedad y velocidad.

Ciertos tipos de procesamiento de datos y análisis a menudo requieren una demanda de velocidad específica. El procesamiento rápido y consulta de datos espaciales es necesario, especialmente para la aplicación de ciertos algoritmos en un streaming en tiempo real. La navegación por carretera, los servicios de emergencia, el almacenamiento frontal/vehicular de las cámaras requieren respuestas en tiempo real o casi inmediato para ser eficaces. La naturaleza distribuida de NoSQL en combinación con los marcos informáticos distribuidos como Hadoop hacen posible el almacenamiento y análisis de datos de alto rendimiento.

Capítulo 2

Acondicionamiento de los datos

La plataforma de espacios en blanco, está compuesta por módulos que mantienen su funcionamiento a flote. Uno de los más importantes es el módulo de cálculo, este módulo se encarga de hacer las operaciones necesarias para obtener la lista de canales disponibles correspondientes un punto geográfico.

Para realizar dichas operaciones, el módulo de calculo emplea un mapa tipo ráster y un shapefile que aportan información acerca del punto consultado. Dicho esto, el ráster y el shapefile, junto con el módulo de cálculo servirán de insumos para construir un banco de datos, con el que sea posible generar los conceptos técnicos requeridos por la ANE.

El departamento de San Andrés y Providencia, se encuentra geográficamente alejado del resto del territorio nacional, adicionalmente la extensión del ráster del módulo de cálculo no abarca este territorio, por tales motivos se hizo inviable incluirlo en el procesamiento global, para obtener los datos de este departamento, se creará un ráster abarque únicamente la extensión de San Andrés y se replicará el proceso realizado con el ráster de elevación del módulo de cálculo.

2.1. Ráster del módulo de cálculo

El módulo de cálculo de la plataforma de espacios en blanco emplea un mapa tipo ráster para la obtención del HAAT como el que se puede ver en la figura 2.1. La zona gris representa la extensión geográfica para la que se tiene información conocida, mientras que la zona negra representa datos nulos o vacíos. Como se puede ver en la imagen, el contorno de Colombia no es exactamente el de la división política, en otras palabras, se encuentran datos en los alrededores del país. Esto es debido al cálculo del HAAT, para el cual se requieren datos en un radio especificado, en este caso los datos de los alrededores sirven para calcular los HAAT correspondiente a las ubicaciones en los límites del país.



Figura 2.1: Ráster módulo de cálculo.

El ráster del módulo de cálculo está compuesto por píxeles. Cada píxel representa una parcela de aproximadamente $50m^2$ como se ilustra en la figura 2.2. Dependiendo del área que cubra el ráster, y el tamaño de cada píxel se puede tener una idea de la resolución del ráster, entre más grande el área y más pequeño el píxel la resolución aumenta, de igual manera aumentará la cantidad de datos que contenga el ráster.

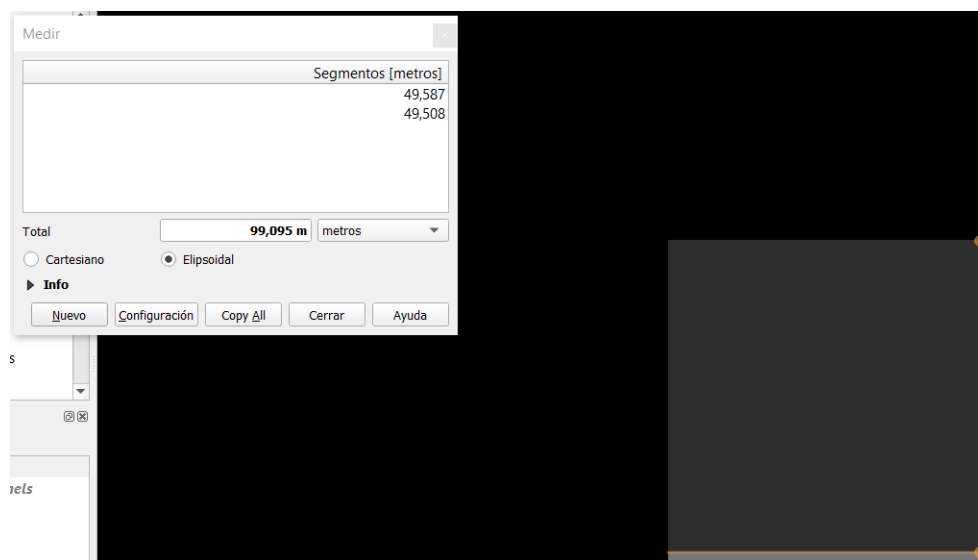


Figura 2.2: Tamaño del píxel en metros.

Los mapas tipo ráster tienen características que se encuentran en sus metadatos, éstas nos proporcionan información valiosa acerca del origen de los datos como se puede ver en la figura 2.3, las mencionadas características ayudan a las herramientas de visualización a realizar un trata-

miento adecuado del archivo en cuestión.

La figura 2.3 indica que el sistema de referencia de coordenadas (SRC) corresponde a “EPSG:4686 - MAGNA-SIRGAS”, este sistema de coordenadas hace referencia a la zona de Colombia incluyendo las islas San Andrés y Providencia. También se puede ver la extensión, combinando las parejas mostradas en este dato, es posible obtener las coordenadas de las cuatro esquinas del ráster.

Nombre	Col_50m_Geog
Ruta	D:\Projects\archivos necesarios\Col_50m_Geog.tif
SRC	EPSG:4686 - MAGNA-SIRGAS - Geográfico
Extensión	-80.0284753983390118,-5.2126723496035368 : -65.7779387636559107,12.8612884709883133
Unidad	grados
Anchura	31748
Altura	40266
Tipo de datos	Float32 - Número de coma flotante de 32 bits
Descripción del controlador de GDAL	GTiff
Metadatos del controlador de GDAL	GeoTIFF
Descripción del conjunto de datos	D:/Projects/archivos necesarios/Col_50m_Geog.tif
Compresión	

Figura 2.3: Propiedades del ráster.

Dimensiones	X: 31748 Y: 40266 Bandas: 1
Origen	-80.0285,12.8613
Tamaño de píxel	0.0004488640744199038236,-0.000448864074419903932

Figura 2.4: Dimensiones del ráster.

La figura 2.4 muestra las dimensiones del ráster, es posible hallar la cantidad de píxeles dentro del ráster, multiplicando sus dimensiones (ancho, alto). Se puede ver que el ráster tiene una banda, a pesar de que no se encuentra información del dato que representa, el autor de ante-mano conoce que esta banda corresponde a la elevación del terreno. También se encuentra el tamaño del píxel, teniendo en cuenta la extensión que tiene el ráster, la cual se puede ver en la figura 2.3, el tamaño del píxel indica que este ráster de elevación tiene una resolución muy alta, esto también se ve reflejado en la cantidad de píxeles que posee (Ecuación 2.1).

$$31\,748 * 40\,266 = 1\,278\,364\,968 \quad (2.1)$$

Ya que el objetivo del presente estudio es construir un ráster similar al de elevación, pero con información referente a la altura efectiva (HAAT) y a la cantidad de canales disponibles en un

punto geográfico, es lógico pensar que con los datos geográficos del ráster de elevación sería posible hacer los cálculos para obtener los datos correspondientes. Sin embargo, una consulta al módulo de cálculo puede tardar entre 0.5 y 1.0 segundos, en el peor de los casos el tiempo de cálculo ascendería a **1 278 364 968** segundos, esto equivale a **40,54** años de procesamiento aproximadamente.

2.2. Shapefile del módulo de cálculo

El módulo de cálculo de la plataforma de espacios en blanco, incluye algunos métodos, que utilizan información con respecto a la división política de Colombia, dicha información es aportada por el shapefile que se puede ver en la figura 2.5.

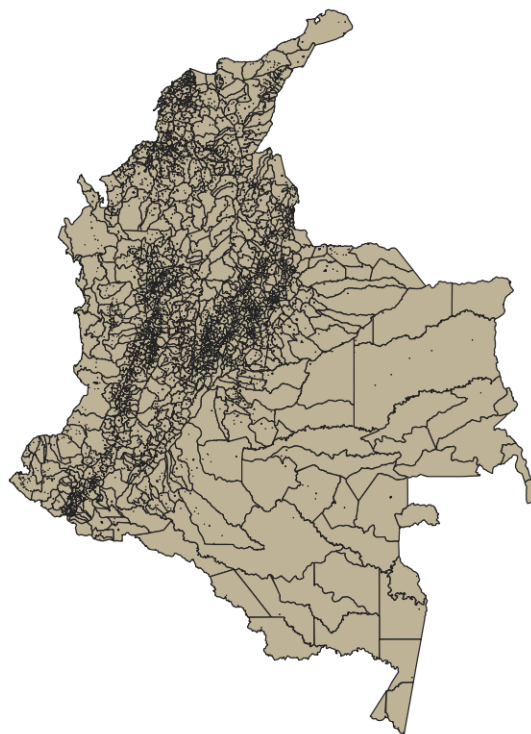


Figura 2.5: Shapefile de Colombia.

Los shapefile al igual que los raster, tienen características que se encuentran en sus metadatos, dichas características ayudan a los software geográficos a darle un correcto tratamiento a los archivos de forma. El ráster de Colombia contiene las propiedades que ilustra la figura 2.6, allí se identifica el sistema de coordenadas, que en este caso corresponde al mismo que el ráster (EPSG:4686 - MAGNA-SIRGAS), debido a que la extensión geográfica es muy parecida a la

del ráster, ya que el shapefile contiene polígonos, en el último campo indica la cantidad de objetos que contiene el archivo.

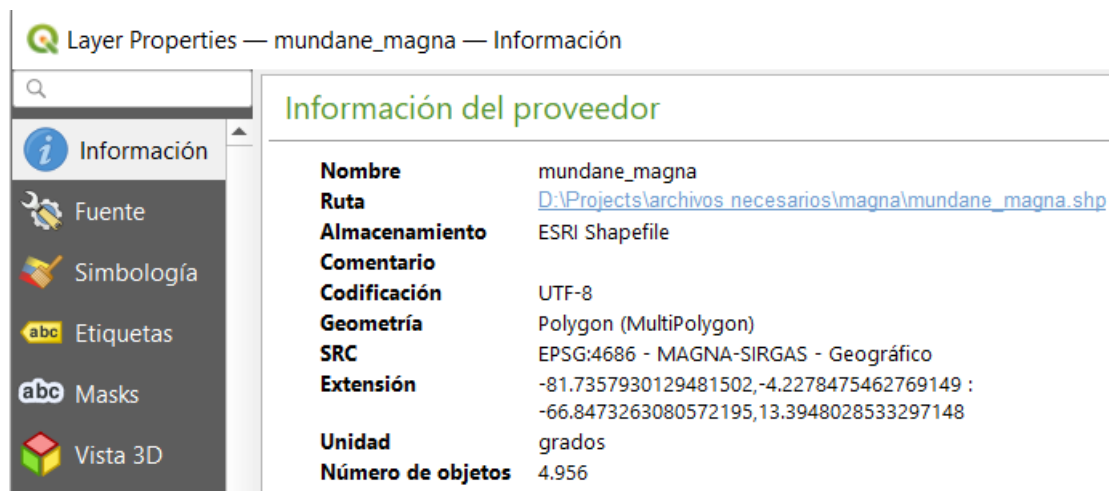


Figura 2.6: Propiedades del Shapefile de Colombia.

Como se mencionó anteriormente el shapefile contiene la división política de Colombia, pero adicionalmente a esto los shapefile pueden tener una tabla de atributos, parecida a una tabla de una base de datos. En la figura 2.7 se puede ver la lista de campos de la tabla con su tipo de dato, longitud, entre otros.

	Id	Nombre	Alias	Tipo	Nombre del tipo	Longitud	Precisión	Coment
abc 0	AREA		QString	String	16	0		
abc 1	PERIMETER		QString	String	13	0		
abc 2	NOM_DPTO		QString	String	26	0		
123 3	DANE_DPTO		int	Integer	2	0		
abc 4	NOM_MPIO		QString	String	27	0		
123 5	DANE_MPIO		int	Integer	5	0		
abc 6	NOM_CPOB		QString	String	56	0		
abc 7	DANE_CPOB		QString	String	8	0		
abc 8	CLASE		QString	String	1	0		
abc 9	ACREAGE		QString	String	12	0		
abc 10	GXCENTER		QString	String	9	0		
abc 11	GYCENTER		QString	String	10	0		

Figura 2.7: Campos del Shapefile de Colombia.

Uno de los campos más importantes de este shapefile es el campo “Clase”, este campo indica qué tipo de zona abarca el polígono en cuestión, dicho campo puede tomar tres valores, el valor “1” para indicar que el polígono corresponde a una cabecera urbana, el valor “2” quiere decir

que el polígono corresponde a una zona rural, para estos dos valores el polígono abarca una porción dentro de un municipio como ilustra la figura 2.8. El valor “3” indica que la extensión del polígono cubre un municipio completo, esto se puede ver en la figura 2.9.

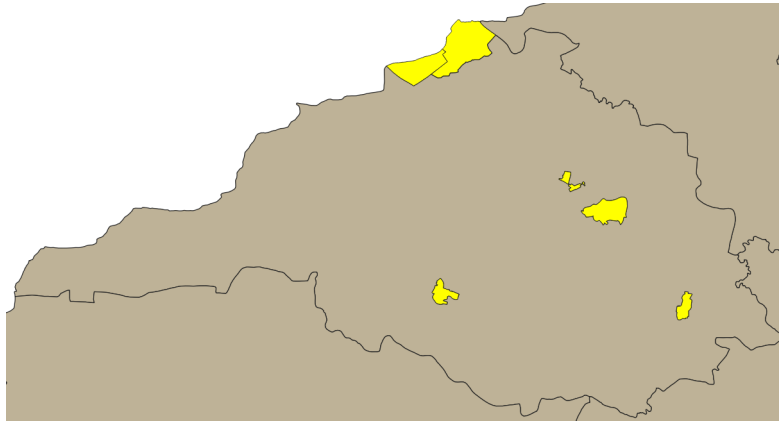


Figura 2.8: Polígonos clase 1 y 2.

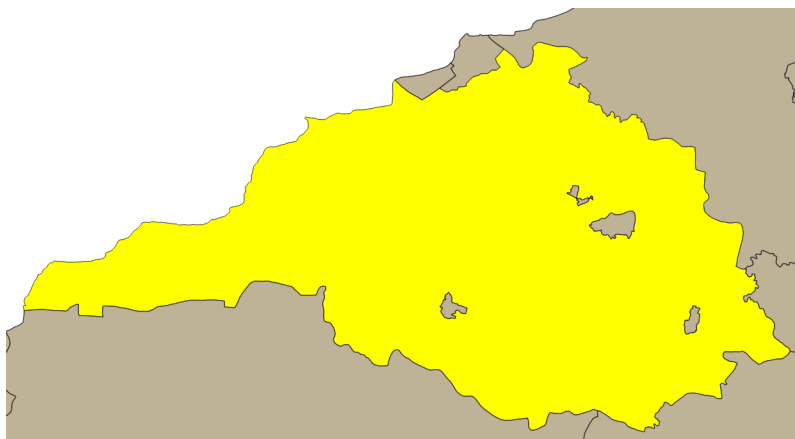


Figura 2.9: Polígono clase 3.

2.3. Remuestreo

Teniendo en cuenta el factor del tiempo, el primer enfoque debe ser reducir la cantidad de píxeles del ráster, de esta manera disminuirá el número de consultas al módulo de cálculo y en consecuencia la cantidad de tiempo de procesamiento empleado. Esto sin afectar la georreferenciación del ráster de elevación que se desea tomar como base para la construcción de un nuevo ráster ya que si esta se altera, es posible que la información no se pueda reconstruir de manera adecuada.

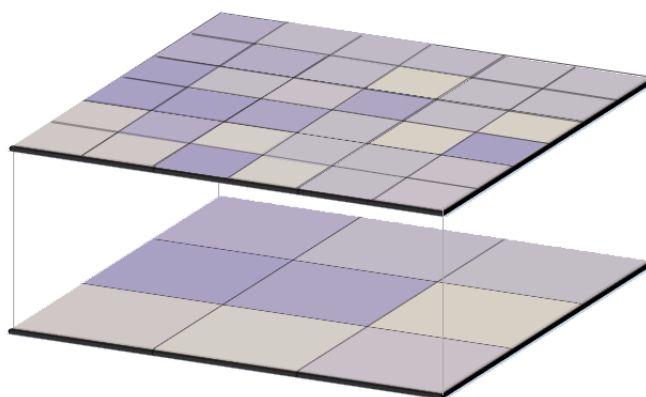


Figura 2.10: Remuestreo de un ráster
Tomado de: <https://gisgeography.com/raster-resampling/>

La operación en la cual se modifica el tamaño del píxel de un ráster se denomina remuestreo (Figura 2.10), esta operación se puede realizar tanto para incrementar como para disminuir el tamaño del píxel, ya sea que se necesite aumentar o reducir la cantidad de píxeles del ráster. Existen varios métodos de remuestreo para un ráster, entre los más usados están los siguientes:

- Vecino más cercano.
- Bilinear.
- Cúbica.
- Promedio.
- Moda.
- Mediana.

En el caso presentado se debe realizar un aumento en el tamaño del píxel con la intención de reducir su cantidad. Para este caso en particular, resulta irrelevante la operación de remuestreo que se utilice, ya que se necesita la base del ráster (los valores de latitud, longitud) para tener una georreferenciación adecuada, sin importar el valor de la banda, debido a que el valor de la banda será reemplazado en un futuro por los datos extraídos del módulo de cálculo. Por tal razón debe usarse el método que ocupe menos tiempo en ser ejecutado.

El método de remuestreo más rápido es el llamado “Vecino más cercano”, el cual consiste en tomar el valor del píxel más cercano a la ubicación geográfica, su velocidad se debe a que no realiza cálculos para interpolar los datos, simplemente toma un valor preexistente en el ráster y lo reubica, como se puede ver en la figura 2.11.

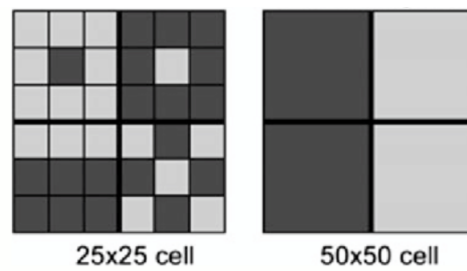


Figura 2.11: Método de remuestreo: Vecino más cercano.

Adaptado de: The Importance of Scale in Land Use Models: Experiments in Data Conversion, Data Resampling, Resolution and Neighborhood Extent

El remuestreo se aplicará utilizando la librería GDAL[10] ya que es la más completa y usada en este entorno, en este caso se hará uso de la herramienta de línea de comandos que ofrece, esta librería ofrece utilidades para abrir ráster, ver sus metadatos, cargar la banda, entre otras. El equipo de trabajo acordó en incrementar en un factor de cuatro el tamaño del píxel del raster de elevación, este tamaño se puede ver en la figura 2.4. El resultado de la ecuación 2.2 nos dará el nuevo tamaño de píxel.

$$0,0004488640744199038236 * 4 = 0,0017954562976796152944 \quad (2.2)$$

Al incrementar el tamaño del píxel en un factor de cuatro, un solo píxel del raster de elevación remuestreado, pasará a representar dieciséis píxeles del raster de elevación original. Naturalmente la cantidad de píxeles se reducirá en un factor de dieciséis, dando como resultado el valor que se muestra en la ecuación 2.3.

$$\frac{1278364968}{16} = 79897810,5 \quad (2.3)$$

Se debe tener en cuenta que las cifras anteriormente mencionadas son estimaciones, por ejemplo, en el resultado de la ecuación 2.3 se puede ver que existe medio píxel, esto en la práctica no se puede dar, la herramienta de remuestreo se encargará de hacer las aproximaciones correspondientes. A continuación se procederá a realizar el remuestreo, para esto se debe tener instalada la librería “GDAL”, los detalles de su instalación se pueden encontrar en su página oficial.

Una vez instalada la librería se debe estar ubicado en la carpeta donde se encuentra el ráster para este caso la ruta se observa en la figura 2.12.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas$ ls
Col_50m_Geog.tif
```

Figura 2.12: Ubicación del mapa.

Hecho esto, se prepara la instrucción para ejecutar el remuestreo, como se ve en el código 2.1. La instrucción ejecuta la herramienta “gdalwarp”, esta utilidad permite realizar mosaico, re-proyección y deformación de una imagen[10]. En la tabla 2.1 se pueden ver los parámetros a utilizar y su descripción.

Parámetro	Descripción
s_srs	Sistema de coordenadas del ráster de entrada.
t_srs	Sistema de coordenadas del ráster de salida.
tr	Tamaño del píxel del ráster de salida.
dstNodata	Valor de dato nulo para el ráster de salida.

Tabla 2.1: Parámetros de entrada: Instrucción remuestreo.

```
gdalwarp -s_srs 'EPSG:4686' -t_srs 'EPSG:4686' -tr
0.0017954562976796152944 0.0017954562976796152944 -dstNodata -9999
Col_50m_Geog.tif Col_200m_Geog.tif
```

Código 2.1: Instrucción de remuestreo.

Una vez ejecutada la instrucción, la consola mostrará una salida como se observa en la figura 2.13, allí se puede ver que inmediatamente la librería . Al iniciar el procesamiento, la herramienta inmediatamente indica la resolución de salida, posteriormente se ve el progreso del remuestreo hasta que es completado.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas$ gdalwarp
-s_srs 'EPSG:4686' -t_srs 'EPSG:4686' -tr 0.0017954562976796152944 0.001
7954562976796152944 -dstNodata -9999 Col_50m_Geog.tif Col_200m_Geog.tif
Creating output file that is 7937P x 10067L.
Processing Col_50m_Geog.tif [1/1] : 0Using internal nodata values (e.g. -
32767) for image Col_50m_Geog.tif.
...10...20...30...40...50...60...70...80...90...100 - done.
```

Figura 2.13: Salida de la instrucción de remuestreo.

Luego de realizarse el remuestreo, en la carpeta donde se encuentra el ráster original, aparecerá el nuevo mapa reproyectado en la lista de archivos como ilustra la figura 2.14

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas$ ls
Col_200m_Geog.tif Col_50m_Geog.tif
```

Figura 2.14: Nuevo archivo en el directorio del ráster.

Al abrir el mapa, a simple vista no se puede evidenciar cambio alguno, ya que tiene la misma forma que el ráster original (figura 2.15).



Figura 2.15: Ráster remuestreado.

Cuando se revisan las propiedades, se puede ver que las dimensiones del ráster cambiaron (figura 2.16), así como el tamaño del píxel (figura 2.17), el factor de reducción de puntos se puede hallar dividiendo la cantidad de puntos del ráster original entre la cantidad de puntos del ráster remuestreado, como se ve en la ecuación 2.4.

Layer Properties — Col_200m_Geog — Información

Información del proveedor

Nombre	Col_200m_Geog
Ruta	C:\Users\Juan\Desktop\mapas\Col_200m_Geog.tif
SRC	EPSG:4686 - MAGNA-SIRGAS - Geográfico
Extensión	-80.0284753983390118,-5.2135700777523724 : -65.7779387636559107,12.8612884709883133
Unidad	grados
Anchura	7937
Altura	10067
Tipo de datos	Float32 - Número de coma flotante de 32 bits
Descripción del controlador de GDAL	GTiff
Metadatos del controlador de GDAL	GeoTIFF
Descripción del conjunto de datos	C:\Users\Juan\Desktop\mapas\Col_200m_Geog.tif

Figura 2.16: Propiedades del ráster remuestreado.

Dimensiones	X: 7937 Y: 10067 Bandas: 1
Origen	-80.0285,12.8613
Tamaño de píxel	0.001795456297679615294,-0.001795456297679615294

Figura 2.17: Dimensiones del ráster remuestreado.

Al medir el tamaño del píxel en metros del ráster remuestreado, se puede ver el incremento con respecto al ráster original, en este caso cuatro veces aproximadamente, esto se observa en la figura 2.18.

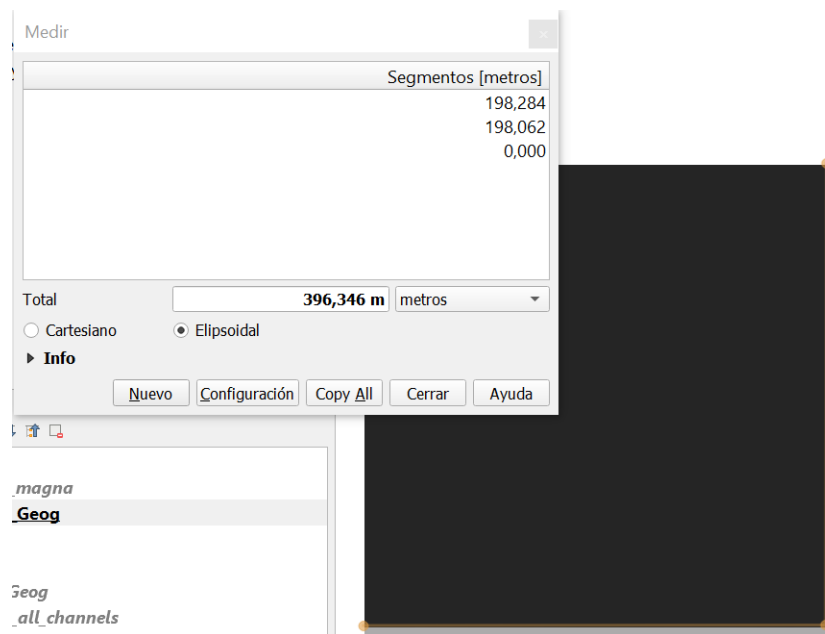


Figura 2.18: Tamaño del píxel en metros del ráster remuestreado.

$$\frac{31\,748 * 40\,266}{7\,937 * 10\,067} = 15,999205 \quad (2.4)$$

Como se observa en la ecuación 2.4 la cantidad de puntos se redujo en un factor cercano al dieciséis. Según la imagen de la figura 2.17, la cantidad de píxeles del ráster remuestreado equivale al valor mostrado en la ecuación 2.5.

$$7\,937 * 10\,067 = 79\,901\,779 \quad (2.5)$$

2.4. Recorte

Una vez realizado el remuestreo, se debe hallar la forma de reducir aún más la cantidad de puntos a calcular. Teniendo en cuenta que el ráster es una imagen, es lógico que tenga una forma

rectangular, en este caso no es favorable, ya que dentro del raster se encuentran puntos que realmente están por fuera de Colombia. Dichos puntos no deben entrar al módulo de cálculo, ya que el módulo de cálculo responde una lista de canales vacía si la ubicación dada no se encuentra dentro del territorio nacional, por lo tanto se incrementa el tiempo de procesamiento.

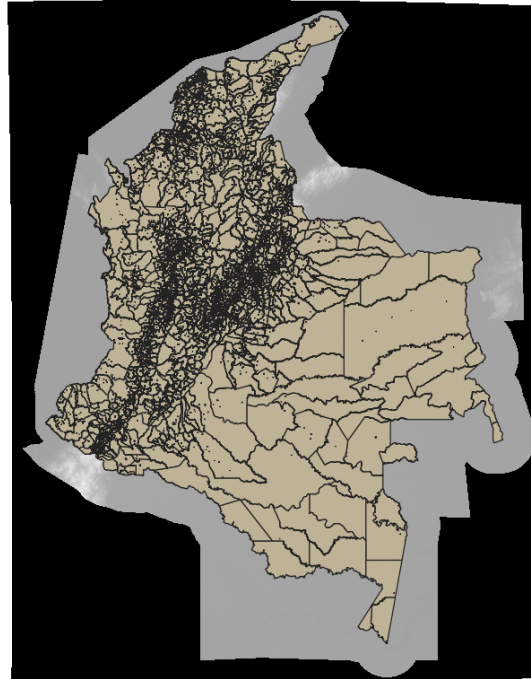


Figura 2.19: Ráster con contorno de Colombia

En la Figura 2.19 se puede ver el shapefile de Colombia, sobrepuesto en el ráster de elevación, la zona gris visible por fuera del shapefile está conformada por píxeles que contienen datos de elevación.

Algunos puntos de la zona que se encuentra por fuera del contorno de Colombia contienen datos de elevación, esto se debe a que el módulo de cálculo trabaja con círculos, si el ráster no tuviera datos para los alrededores del país, no sería posible calcular el HAAT para puntos que se encuentren en la frontera, el módulo de cálculo es capaz de descartar ubicaciones que se encuentren fuera del territorio nacional, pero esto implica realizar algunos cálculos que pueden llevar tiempo de procesamiento.

Con el fin de descartar estas ubicaciones desde el principio, a estos pares latitud y longitud, debe asignarse un valor de dato nulo o vacío. Dicha tarea se puede realizar recortando el ráster

con la forma del shapefile de Colombia, así solo contendrán datos en los puntos que se encuentren dentro del territorio nacional, y podremos omitir los puntos que se encuentran rellenando el raster para mantener su forma rectangular.

Al aplicar un recorte sobre un ráster, este cambiará sus dimensiones, debido a que la extensión del nuevo ráster será la misma que la del shapefile del cual se tome. El shapefile de Colombia tiene una extensión más grande que el ráster de elevación, esto se debe a que el shapefile incluye el departamento de San Andrés y Providencia, para los cuales el ráster no tiene datos (figura 2.20), por tanto se debe crear un nuevo shapefile que no incluya ese departamento.

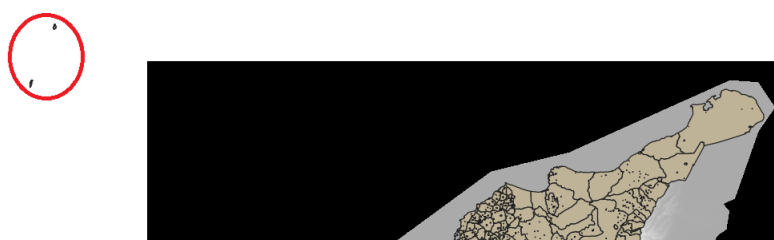


Figura 2.20: Departamento de San Andrés y Providencia.

El primer paso es realizar una carpeta donde se guardará el nuevo shapefile, dentro del directorio de los mapas, allí también se debe copiar el shapefile en su estado actual, como se observa en la figura 2.21.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas
$ mkdir shape
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas
$ ls
Col_200m_Geog.tif  Col_50m_Geog.tif  magna  shape
```

Figura 2.21: Carpeta shape.

Luego de esto, se crea el nuevo shapefile aplicando un filtro sobre el shapefile original, para esto se usará la herramienta llamada “ogr2ogr” de la librería “GDAL”, la instrucción empleada se puede ver en el código 2.2, en la tabla 2.2 se pueden ver los parámetros configurados para la ejecución y su respectiva descripción.

```
ogr2ogr -where "DANE_DPTO != '88'" -f 'ESRI Shapefile' shape/
mundane_magna88.shp magna/mundane_magna.shp -lco ENCODING=UTF-8
```

Código 2.2: Instrucción de filtro.

Parámetro	Descripción
where	Filtro por atributos parecido a SQL.
f	Nombre del formato.
lco	Opciones de creación de capa.

Tabla 2.2: Parámetros de entrada: Instrucción filtro.

Al ejecutar la instrucción no aparece nada en la salida, pero cuando se mira el contenido de la carpeta se puede ver que se crearon los archivos correspondientes al nuevo shapefile, como se observa en la figura 2.22.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas
$ ls shape
mundane_magna88.cpg  mundane_magna88.prj  mundane_magna88.shx
mundane_magna88.dbf  mundane_magna88.shp
```

Figura 2.22: Nuevo shapefile.

Cuando revisamos las propiedades del nuevo shapefile(figura 2.23), se puede ver que se redujo su extensión, ya que no tiene almacenados los polígonos correspondientes al departamento de San Andrés y Providencia.

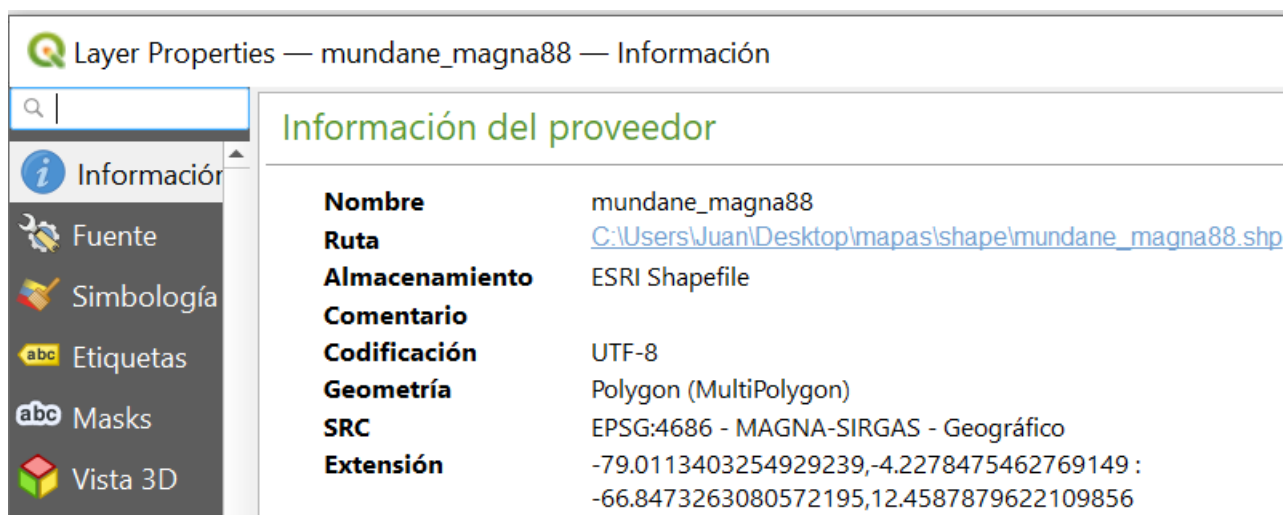


Figura 2.23: Propiedades del nuevo shapefile.

Una vez creado el nuevo shapefile, es el momento de recortar el ráster. Para esto se utilizará la herramienta “gdalwarp”, en este caso se utilizarán parámetros distintos comparado a la vez

anterior como se puede ver en la tabla 2.3, allí se especifican cada uno de los parámetros y se describe su uso. En el código 2.3 se observa el comando con los parámetros configurados.

```
gdalwarp -crop_to_cutline -dstNodata -32767 -cutline shape/
mundane_magna88.shp -t_srs 'EPSG:4686' Col_200m_Geog.tif
Col_200m_Geog_cutted.tif
```

Código 2.3: Instrucción de recorte.

Parámetro	Descripción
crop_to_cutline	Incluir píxeles que toquen la linea exterior del shapefile.
dstNodata	Valor de dato nulo para la salida.
cutline	Nombre del shapefile.
t_srs	Sistema de coordenada de la salida.

Tabla 2.3: Parámetros de entrada: Instrucción recorte.

Al ejecutar el comando, muestra en consola una salida como se ve en la figura 2.24, allí se puede observar que la librería identifica automáticamente el tamaño de salida, y procede a realizar el procesamiento, también se pueden ver otras especificaciones, como lo son el valor de dato nulo que se va a manejar, y por último aparece progresivamente el avance del procesamiento.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas
$ gdalwarp -crop_to_cutline -dstNodata -32767 -cutline shape/mu
ndane_magna88.shp -t_srs 'EPSG:4686' Col_200m_Geog.tif Col_200m
_Geog_cutted.tif
Creating output file that is 6774P x 9292L.
Processing Col_200m_Geog.tif [1/1] : 0Using internal nodata val
ues (e.g. -9999) for image Col_200m_Geog.tif.
...10...20...30...40...50...60...70...80...90...100 - done.
```

Figura 2.24: Salida del comando recorte.

Cuando se abre el nuevo ráster, se puede ver como su forma está mucho más delineada (figura 2.25), debido a que se removieron los excedentes de puntos de que se encontraban fuera de Colombia. También se puede observar, que hubo cambios en la escala de colores de la representación, esto se debe a los cambios en la escala de datos producidos por el recorte realizado.



Figura 2.25: Ráster recortado.

Después de realizar el recorte, se calcula nuevamente la cantidad de píxeles, para esto tomamos las dimensiones que se encuentran dentro de las propiedades del ráster (figura 2.26) y luego se multiplican (ecuación 2.6), en las propiedades del ráster también se puede ver que la extensión del nuevo ráster coincide con la extensión del shapefile creado sin el departamento de San Andrés, no obstante la proyección se mantuvo como era de esperarse.

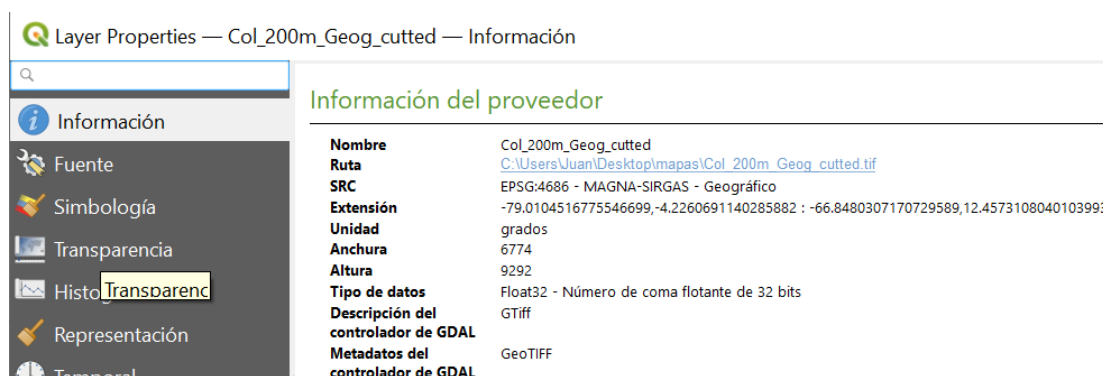


Figura 2.26: Propiedades del ráster recortado.

$$6774 * 9292 = 62\,944\,008 \quad (2.6)$$

Como se puede ver en la ecuación 2.6 gracias a la operación de recorte, fueron descartados casi diecisiete millones de puntos.

2.5. Reproyección

El formato de archivo de un ráster depende del sistema de coordenadas que posea, en el caso del ráster de elevación, corresponde al SRC “EPSG:4686 - MAGNA-SIRGAS” como se puede ver en la Figura 2.27. Para dicho sistema de referencia la extensión del archivo es “.TIFF”. A pesar de que este tipo de archivo es muy común en el entorno de los mapas, existen muy pocos motores de bases de datos que permitan importar datos desde archivos con esta estructura.

Ruta	D:\Projects\archivos necesarios\Col 50m Ge
SRC	EPSG:4686 - MAGNA-SIRGAS - Geográfico
Extensión	-80.0284753983390118,-5.212672349603536E

Figura 2.27: Sistema de coordenadas del ráster de elevación.

La operación de cambiar el sistema de coordenadas de un ráster se denomina reproyección (Figura 2.28), dicha operación puede ser útil en los casos que se requiera cambiar la zona geográfica a la que hace referencia el ráster, puesto que la concavidad no es la misma en todas las zonas del planeta, el sistema de coordenadas o proyección se toma como base para llegar a una aproximación visual más exacta por parte de los software geográficos.

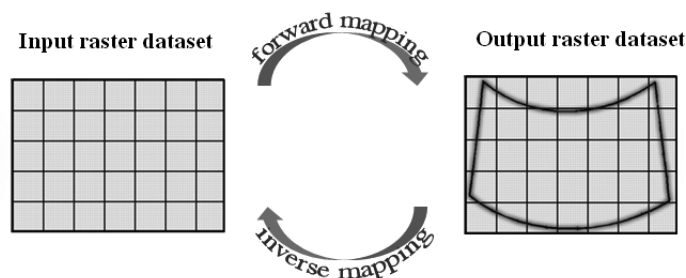


Figura 2.28: Reproyección de un ráster.

Tomado de: <https://cartographicperspectives.org/index.php/journal/article/view/cp71-finn-et-al/html>

En el caso de este estudio, la reproyección se usará como herramienta para cambiar el tipo de archivo y aprovechar las facilidades que ofrecen algunos motores de base de datos para importar información. El sistema de coordenadas más sencillo se denomina “XYZ”, este sistema de coordenadas se caracteriza por almacenar únicamente los datos de ubicación y valor de banda para un punto geográfico, la extensión correspondiente a los archivos ráster que tienen como

proyección o SRC el sistema de coordenadas “XYZ” es CSV, este tipo de archivo tiene una estructura bastante simple, y es ampliamente utilizado en todo tipo de sistemas. En la figura 2.29 se observa la estructura de un archivo de ráster con sistema de coordenadas “XYZ”.

```
1 longitude,latitude,band
2 -75.255812,5.028571,2849.3987
```

Figura 2.29: Sistema de coordenadas XYZ.

Al realizar una reproyección de un ráster al sistema de coordenadas “XYZ”, se genera un archivo de texto CSV el cual contiene una línea por cada uno de los píxeles del ráster, se debe tener en cuenta que los archivos CSV son archivos de texto plano, por lo tanto no tienen aplicado algún tipo de compresión. Si se cuenta con un ráster que posea una gran cantidad de píxeles, el peso del archivo de texto correspondiente será bastante grande.

La reproyección se realizará haciendo uso de la herramienta “gdaltranslate” que ofrece la librería “GDAL”, para traducir entre los diferentes sistemas de coordenadas que existen en el mundo. Los parámetros de entrada de la herramienta se pueden ver en el código 2.4, en la tabla 2.4 se observa a detalle la función de cada uno, para ejecutar esta instrucción es importante estar ubicado en el directorio donde se encuentra el ráster de entrada.

Parámetro	Descripción
of	Formato de salida.

Tabla 2.4: Parámetros de entrada: Instrucción remuestreo.

```
1 gdal_translate -of 'XYZ' Col_200m_Geog_cutted.tif Col_200m_Geog_cutted.
   csv
```

Código 2.4: Instrucción de reproyección.

Después de ejecutar la instrucción la consola mostrará una salida parecida a la de la figura 2.30. Cuando finalice la reproyección, en la carpeta aparecerá un archivo de extensión csv como el se observa en la figura 2.31.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas
$ gdal_translate -of 'XYZ' Col_200m_Geog_cutted.tif Col_200m_Geog_cutted.csv
Input file size is 6774, 9292
0...10...20...30...40...50...60...70...80...90...100 - done.
```

Figura 2.30: Salida de la instrucción de reproyección.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas
$ ls
Col_200m_Geog.tif          Col_50m_Geog.tif  shape
Col_200m_Geog_cutted.csv  magna
Col_200m_Geog_cutted.tif  main.py
```

Figura 2.31: Archivo CSV generado.

Luego de ejecutar la instrucción se obtiene un archivo CSV con **62'944.008** líneas como se observa en la figura 2.32.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas
$ sed -n '$=' Col_200m_Geog_cutted.csv
62944008
```

Figura 2.32: Cantidad de líneas del archivo CSV generado.

Capítulo 3

Diseño de la base de datos

En el caso de este estudio, el volumen de datos a manejar asciende a cifras de millones de datos. Por este motivo se requiere un motor de base de datos adecuado a las necesidades que se han planteado hasta el momento. Para el diseño de la base de datos, se deben tener en cuenta los siguientes aspectos:

- El volumen de datos a manejar.
- La velocidad de acceso a los datos.
- La flexibilidad en la estructura de datos.

En cuanto al volumen de datos que se debe manejar, con lo tratado en la sección anterior, se puede evidenciar que los ráster representan cantidades de datos muy grandes de manera estructurada, y dependiendo del área o extensión a la que haga referencia, puede tener enormes cantidades de datos si se manejan tamaños pequeños de píxel, en el caso del ráster de elevación está por el orden de miles de millones; aunque esta cantidad de datos se vio reducida, para tener un mapa que permita ver una aproximación adecuada del estado actual de la plataforma, este número no debería bajar del orden de los millones.

La herramienta desarrollada para este fin será ejecutada en infraestructura alquilada, la cual se cobra por hora. Debido a esto, los datos deben ser procesados de la manera más rápida y eficiente posible, se hace necesario contar con un motor de base de datos que sea capaz de buscar entre millones de registros en la menor cantidad de tiempo, ya sea para tareas de lectura o escritura de datos. Dicho motor debe ofrecer la posibilidad de optimizar las consultas puesto que se realizarán estadísticas para obtener información a nivel departamental y municipal.

El módulo de cálculo realiza operaciones internas para la obtención del listado de canales correspondiente a una ubicación dada, es posible que luego de realizar el procesamiento de los puntos extraídos del ráster, sea requerida una ampliación, ya sea para algunos departamentos o municipios si se desean agregar otros datos que pudieran enriquecer dicho banco de datos. Por tanto es necesario que el motor de base de datos sea poco restrictivo en cuanto a la estructura

o la cantidad de campos que contiene un registro, en otras palabras, que algunas entradas de datos tengan campos adicionales o carezcan de alguno.

3.1. Motor de base de datos

Teniendo en cuenta lo anterior, se seleccionó el motor de base de datos NoSQL MongoDB[7], debido a que las bases de datos relacionales tienden a reducir considerablemente su rendimiento al aumentar la cantidad de registros por encima del millón. En estos casos los administradores recurren a estrategias para dividir el contenido de una tabla entre varias. Al tener estructuras de datos tan sencillas, lo ideal es evitar este tipo de medidas, ya que podrían retrasar aún más el tiempo de consulta.

MongoDB ofrece la posibilidad de crear índices, los cuales son de bastante utilidad a la hora de realizar consultas, con el uso de índices se puede reducir el tiempo de consulta drásticamente ya que mongo los carga en memoria RAM; De esta manera las consultas realizadas serán optimizadas al máximo.

Sin embargo, el uso de índices podría aumentar considerablemente el uso de memoria RAM de la base de datos, si a esto le adicionamos la carga en memoria de algunos documentos cuando se requieren, en el caso de una cantidad grande de clientes, la base de datos puede llegar a consumir una gran cantidad de memoria RAM.

La estructura de MongoDB basada en documentos, es bastante dinámica, ya que permite adicionar y remover campos según sea necesario. El lenguaje de consulta de MongoDB es javascript, dicho lenguaje tiene una sintaxis sencilla que facilita la interacción con la base de datos, y permite realizar consultas de una manera rápida en su consola.

3.2. Estructura de la base de datos

Para el caso de este estudio se plantea un modelo de base de datos compuesto de tres colecciones, una dedicada a almacenar todos los puntos del ráster, otra que contendrá los puntos que se

introducirán al módulo de cálculo, y una temporal cuya función es la de servir como cola de cálculo para evitar pérdidas de datos en caso de un apagado de emergencia.

La colección que contendrá los puntos tendrá la estructura especificada en el código 3.1:

```
1 {  
2     "_id": ObjectID(),  
3     "latitude" : float,  
4     "longitude" : float,  
5     "band": float  
6 }
```

Código 3.1: Colección “locations”

Esta colección corresponde a los datos extraídos de ráster, el campo “_id” es el índice por defecto creado por mongo, el campo “band” en este caso hace referencia al dato de elevación contenido en el raster.

La estructura de la colección de puntos a introducir en el módulo de cálculo, varía durante la ejecución, la estructura que deberá tener al final de las consultas al módulo se puede ver en el código 3.2:

```
1 {  
2     "_id": ObjectID(),  
3     "latitude" : float,  
4     "longitude" : float,  
5     "place": Object(),  
6     "haat": Object(),  
7     "method_1": Object(),  
8     "method_2": Object(),  
9     "adj_v": array  
10 }
```

Código 3.2: Colección “calculated_locations”

Como se puede ver en el código 3.2 la colección contiene varios objetos, los cuales contienen los datos extraídos de módulo de cálculo, el campo “adj_” corresponde al vector adyacente calculado para esta coordenada. El objeto “place” se estructura como indica el código 3.3, el

campo “is_valid” indica si el punto se encuentra en una zona urbana prohibida, los campos dentro del array “city” hacen referencia a la información obtenida por el módulo de cálculo en cuanto al municipio y departamento donde se encuentra la coordenada.

```
1 {  
2     "is_valid" : boolean,  
3     "city" : {  
4         "cod_dane" : String,  
5         "cod_dep" : String,  
6         "cod_mun" : String,  
7         "nom_dep" : String,  
8         "nom_mun" : String,  
9         "clase" : String,  
10        "verify" : boolean  
11    }  
12 }
```

Código 3.3: Objeto “place”

En el código 3.4 se puede ver la estructura del objeto haat. El campo “haat_value” indica el valor del HAAT para la coordenada ingresada, el campo “haat_altitude” corresponde a la altura obtenida para este punto en el ráster de elevación, el atributo “haat_mean” es la media del HAAT para la coordenada ingresada, el valor del campo “haat_hatvws” equivale a la medida del HAAT adicionando la altura de la antena y por último el valor del atributo “is_haat_valid” indica si el valor del HAAT es menor al umbral definido en la configuración del módulo de cálculo.

```
1 {  
2     "haat_value" : float,  
3     "haat_altitude" : float,  
4     "haat_mean" : float,  
5     "haat_hatvws" : float,  
6     "is_haat_valid" : boolean  
7 }
```

Código 3.4: Objeto “haat”

Los dos métodos de cálculo comparten la misma estructura, la cual se puede ver en el código

go 3.5. Dicha estructura incluye un arreglo llamado “ccr” que contiene los radios cocanal para cada una de las modulaciones admitidas por el módulo, el atributo “is_valid_header” indica si alguno de los radios cocanal toca una zona prohibida, el objeto “channels” contiene la lista de canales indicando si está o no disponible el canal especificado, por último el atributo “all_channels” hace referencia al conteo de canales disponibles para la pareja coordenada.

```

1 {
2   "ccr" : array,
3   "is_valid_header" : boolean,
4   "channels" : {
5     "14" : int, "15" : int, "16" : int, "17" : int, "18" : int, "19" : int
6     , "20" : int, "21" : int, "22" : int, "23" : int, "24" : int, "25" : int, "
7     26" : int, "27" : int, "28" : int, "29" : int, "30" : int, "31" : int, "32
8     " : int, "33" : int, "34" : int, "35" : int, "36" : int, "37" : int, "38"
9     : int, "39" : int, "40" : int, "41" : int, "42" : int, "43" : int, "44" :
10    int, "45" : int, "46" : int, "47" : int, "48" : int, "49" : int, "50" :
11    int, "51" : int
12  },
13  "all_channels" : int
14 }
```

Código 3.5: Objeto “met.n”

La colección temporal comparte su estructura con la colección de salida, en el capítulo 4 se explicarán los cambios que sufren ambas colecciones a medida que avanza el procesamiento de los datos.

3.3. Construcción de la base de datos

Con el fin de acceder a los datos de ubicación del ráster, se deben importar en la base de datos, esto permite el desarrollo de una herramienta que facilite el cálculo de los parámetros correspondientes a una coordenada geográfica. MongoDB posee métodos que facilitan la inserción de datos, entre ellos existe uno llamado “Mongoimport” este método permite importar datos desde un archivo plano, admite formatos JSON, CSV y TSV.

La herramienta “Mongoimport” es un poco restrictiva en cuanto al formato de los archivos

CSV, esta únicamente permite el separador “,” por este motivo no es posible importar directamente el archivo a una colección dentro de la base de datos. Esto se debe a que el archivo CSV generado por la librería “GDAL” usa como separador el espacio (figura 3.1). Teniendo en cuenta lo anterior, se debe cambiar el separador de espacio a coma.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas$ head Col_200m_Geog_cutted.csv
-79.0095539494058272 12.4564130758615601 -32767
-79.007758493108156 12.4564130758615601 -32767
-79.0059630368104706 12.4564130758615601 -32767
-79.0041675805127852 12.4564130758615601 -32767
-79.002372124215114 12.4564130758615601 -32767
-79.0005766679174286 12.4564130758615601 -32767
-78.9987812116197574 12.4564130758615601 -32767
-78.996985755322072 12.4564130758615601 -32767
-78.9951902990243866 12.4564130758615601 -32767
-78.9933948427267154 12.4564130758615601 -32767
```

Figura 3.1: Separador del archivo generado por la instrucción de reproyección.

Con la intención de cambiar el separador, se realizaron intentos con editores de texto como “Visual Studio Code”, “Notepad++” entre otros, pero esto no fue posible debido a que el archivo CSV contiene millones de líneas, lo cual ralentiza mucho cualquier editor de texto. Por este motivo, se consideró el uso de un editor de flujo como lo es “sed”, este editor de flujo viene instalado por defecto con algunas distribuciones de Linux como Ubuntu. La ventaja de los editores de flujo, es precisamente el que pueden manejar archivos enormes puesto que no se cargan en la memoria RAM, estos editores acceden a los archivos línea por línea ejecutando la instrucción dada por el usuario, dichos editores trabajan con expresiones regulares para ejecutar instrucciones.

Una vez mencionado esto, se procede a cambiar el separador del archivo CSV, para esto se ejecuta la instrucción que se observa en el código 3.6, estando en la carpeta donde se encuentra el archivo CSV. La bandera “i” indica que se modifica el archivo, el editor por defecto genera un archivo nuevo.

```
1 sed -i 's/\s/,/g' Col_200m_Geog_cutted.csv
```

Código 3.6: Instrucción sed para cambiar el separador por coma.

La ejecución del comando tarda dependiendo la cantidad de líneas del archivo, ya que este archivo contiene millones, es normal que el editor de flujo gaste más tiempo debido a los accesos

al disco duro. Una vez finalizada la ejecución el archivo ahora está separado por comas, como se observa en la figura 3.2.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas$ head Col_200m_Geog_cutted.csv
-79.0095539494058272,12.4564130758615601,-32767
-79.007758493108156,12.4564130758615601,-32767
-79.0059630368104706,12.4564130758615601,-32767
-79.0041675805127852,12.4564130758615601,-32767
-79.002372124215114,12.4564130758615601,-32767
-79.0005766679174286,12.4564130758615601,-32767
-78.9987812116197574,12.4564130758615601,-32767
-78.996985755322072,12.4564130758615601,-32767
-78.9951902990243866,12.4564130758615601,-32767
-78.9933948427267154,12.4564130758615601,-32767
```

Figura 3.2: Separador del archivo posterior a la instrucción del editor de flujo.

Una vez modificado el separador, se debe importar el archivo separado por comas, dentro de la base de datos Mongo, en MongoDB no es necesario crear una base de datos ni crear colecciones, estas se crearán conforme se inserten documentos en ellas. Para esto se prepara la instrucción que ejecuta la herramienta “Mongoimport” como se observa en el código 3.7 con sus parámetros de entrada, en la tabla 3.1 se pueden ver los parámetros de entrada utilizados para configurar la herramienta mencionada anteriormente.

```
mongoimport --type csv -d col200 -c locations --fields="longitude","
latitude","band" Col_200m_Geog_cutted.csv
```

Código 3.7: Instrucción para importar los datos a MongoDB.

Parámetro	Descripción
type	Tipo del archivo de entrada.
d	Nombre de la base de datos.
c	Nombre de la colección.
fields	Nombres de los campos a importar en orden.

Tabla 3.1: Parámetros de entrada: Instrucción de importación.

Una vez ejecutada la instrucción, en la consola se mostrará constantemente el progreso del procedimiento como se observa en la figura 3.3. Al concluir la ejecución, se deberá tener creada la base de datos y la colección “locations”, donde se guardan todos los puntos del ráster.

```

PS C:\Users\Juan\Desktop\mapas> mongoimport --type csv -d col200
0 -c locations --fields="longitude","latitude","band" Col_200m_
Geog_cuttet.csv
2021-03-13T15:17:36.861-0500    connected to: mongoddb://localhost
/
2021-03-13T15:17:39.862-0500    [.....] col200
.locations    11.8MB/3.12GB (0.4%)
2021-03-13T15:17:42.862-0500    [.....] col200
.locations    23.8MB/3.12GB (0.7%)
2021-03-13T15:17:45.862-0500    [.....] col200
.locations    36.0MB/3.12GB (1.1%)
2021-03-13T15:17:48.861-0500    [.....] col200
.locations    48.0MB/3.12GB (1.5%)
2021-03-13T15:17:51.862-0500    [.....] col200
.locations    60.6MB/3.12GB (1.9%)
2021-03-13T15:17:54.861-0500    [.....] col200
.locations    73.0MB/3.12GB (2.3%)
2021-03-13T15:17:57.861-0500    [.....] col200
.locations    85.3MB/3.12GB (2.7%)
2021-03-13T15:18:00.861-0500    [.....] col200
.locations    98.0MB/3.12GB (3.1%)
2021-03-13T15:18:03.861-0500    [.....] col200
.locations    110MB/3.12GB (3.4%)

```

Figura 3.3: Ejecución de la instrucción “MongoImport”.

Posterior a la ejecución de la instrucción, se comprueba la creación de la base de datos como se observa en la figura 3.4, y de la colección con la estructura definida en la sección 3.2 (figura 3.5).

```

> show dbs
admin      0.000GB
col200     1.894GB
config     0.000GB
local      0.000GB
>

```

Figura 3.4: Base de datos creada.

```

> use col200
switched to db col200
> show collections
locations

```

Figura 3.5: Colección creada.

La estructura creada en la colección “locations” que se observa en la figura 3.6 corresponde a la definida en el código 3.1.

```

> db.locations.find().pretty()
{
  "_id" : ObjectId("604d1de0acf131ce1f5351e3"),
  "longitude" : -79.00057666791743,
  "latitude" : 12.45641307586156,
  "band" : -32767
}

```

Figura 3.6: Estructura de la colección creada.

Como medio de verificación, también se puede contar el número de documentos que contiene la colección, como se observa en la figura 3.7 la cantidad corresponde al número de puntos del ráster.

```
> use col200
switched to db col200
> db.locations.count()
62944008
```

Figura 3.7: Cantidad de documentos de la colección creada.

3.4. Automatización

Con el fin de facilitar el proceso, el autor decidió automatizar la tarea de pasar de un ráster a una base de datos, esto debido a que convertir un ráster a base de datos requiere de atención por parte de la persona encargada, y la transición entre los comandos no tiene un tiempo definido, además de que el archivo CSV que se genera en la transición nunca se vuelve a utilizar, esto podría dejar residuos en la máquina donde se esté trabajando.

La herramienta fue escrita en bash, dentro de ella se realizan paso a paso los siguientes procedimientos:

- Reproyección del ráster (sección 2.5).
- Cambio del separador (sección 3.3).
- Importación de los datos (sección 3.3).

Esta herramienta recibe por parámetro el ráster a convertir, crea en el servicio de mongo la base de datos correspondiente, con el mismo nombre del ráster de entrada. Con el fin de realizar pruebas se creó un ráster que abarca la extensión de montería (figura 3.8), con sus dimensiones(figura 3.9), podemos calcular que contiene **131 472** puntos.



Figura 3.8: Ráster de montería.

Nombre	Monteria
Ruta	C:\Users\Juan\Desktop\mapas\Monteria.tif
SRC	EPSG:4686 - MAGNA-SIRGAS - Geográfico
Extensión	-76.2669950783390078,8.2397847269883133 : -75.6709036863390168,8.9507853029883133
Unidad	grados
Anchura	332
Altura	396

Figura 3.9: Propiedades del ráster de montería.

Usando la herramienta, para convertir este ráster a base de datos MongoDB únicamente se debe ejecutar la instrucción mostrada en el código 3.8.

```
1 ./tif_to_mongo Monteria.tif
```

Código 3.8: Herramienta “tif_to_mongo” convertir de ráster a base de datos MongoDB.

Al ejecutar la instrucción, se evidencia que se realizó correctamente en la figura 3.10.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Users/Juan/Desktop/mapas$ ./tif_to_mongo.sh Monteria.tif
Input file size is 332, 396
0...10...20...30...40...50...60...70...80...90...100 - done.
2021-03-13T18:45:47.565-0500    connected to: localhost
2021-03-13T18:45:49.028-0500    imported 131472 documents
Done
```

Figura 3.10: Ejecución herramienta “tif_to_mongo”.

Luego de finalizar la ejecución, se realizan las comprobaciones de creación de base de datos(figura 3.11) y de colección(figura3.12).

```
> show dbs
Monteria  0.004GB
admin     0.000GB
```

Figura 3.11: Creación de la base de datos (Montería).

```
> use Monteria
switched to db Monteria
> show collections
locations
```

Figura 3.12: Creación de la colección “locations” (Montería).

Por último, se verifica la estructura (figura 3.13) y la cantidad de documentos de la colección (figura 3.14).

```
> db.locations.find().pretty()
{
  "_id" : ObjectId("604d4eab30d14d75b7cf2ce6"),
  "longitude" : -76.253529158339,
  "latitude" : 8.949887574988313,
  "band" : -32767
}
```

Figura 3.13: Estructura de la colección “locations” (Montería).

```
> db.locations.count()
131472
```

Figura 3.14: Cantidad de documentos de la colección “locations” (Montería).

Capítulo 4

Procesamiento de los datos

El procesamiento de los datos es una de las fases críticas de este estudio, ya que repetir alguna operación debido a un apagado de emergencia o errores de cálculo podría ser muy costoso. Para ejecutar este procesamiento se cuenta con una infraestructura alquilada, la cual aumenta sus costos proporcionalmente a la capacidad del servidor y al tiempo que se requiera dicho servidor. Por los motivos mencionados anteriormente, esta etapa debe ser optimizada en su totalidad, ya que de su éxito depende la calidad del estudio realizado, adicionalmente se realizarán pruebas con varios municipios, antes de ejecutar los procedimientos enunciados a continuación sobre los datos de todo el país.

4.1. Multiproceso

Anteriormente vimos estrategias para reducir la cantidad de puntos a procesar, pero hemos llegado a un punto donde no es viable seguir disminuyendo esta cantidad, ya que se puede ver afectada la calidad de los resultados. En el esfuerzo de reducir aun más el tiempo de procesamiento, se realizará un modelo de multiproceso acorde con el planteamiento visto en las secciones anteriores.

Se propone un modelo de multiproceso simétrico, en el cual existe un proceso principal que se encarga de ejecutar el flujo general del módulo de cálculo, adicionando otras tareas como son:

- Creación de hilos secundarios.
- Carga de la colección de cola en memoria.
- Control de uso de memoria.
- Creación y uso de colas de procesamiento en la base de datos.

Se puede apreciar el flujo completo de manera ordenada en la figura 4.1.

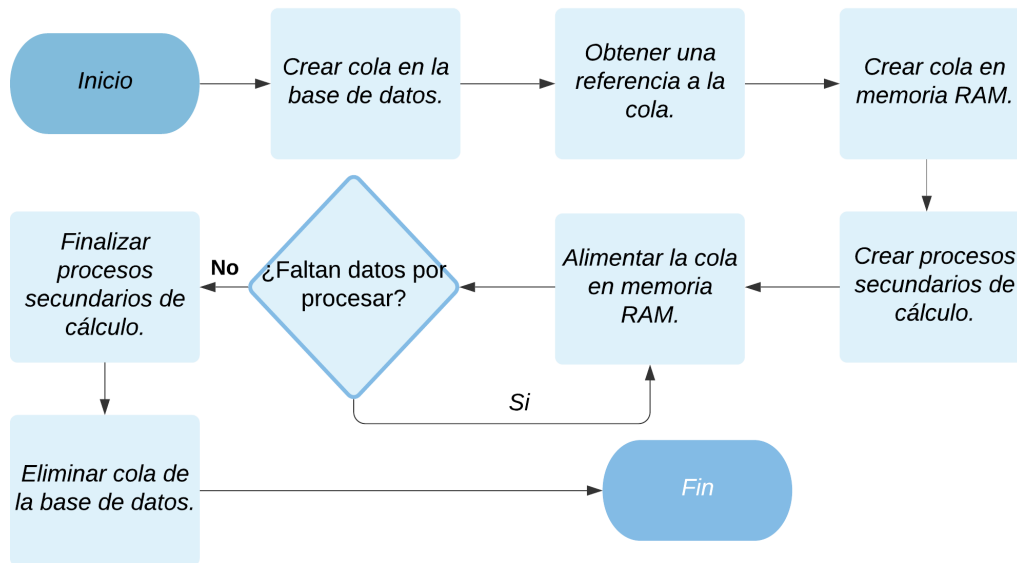


Figura 4.1: Flujo del hilo principal.

Con referencia a los hilos secundarios, estos deben realizar las siguientes tareas:

- Operaciones de escritura en la base de datos.
- Actualizar los registros calculados en la colección de salida.
- Eliminar registros calculados de la colección de cola.
- Ejecutar cualquiera de los métodos de cálculo del módulo.

Los hilos secundarios siguen el flujo especificado en la figura 4.2.

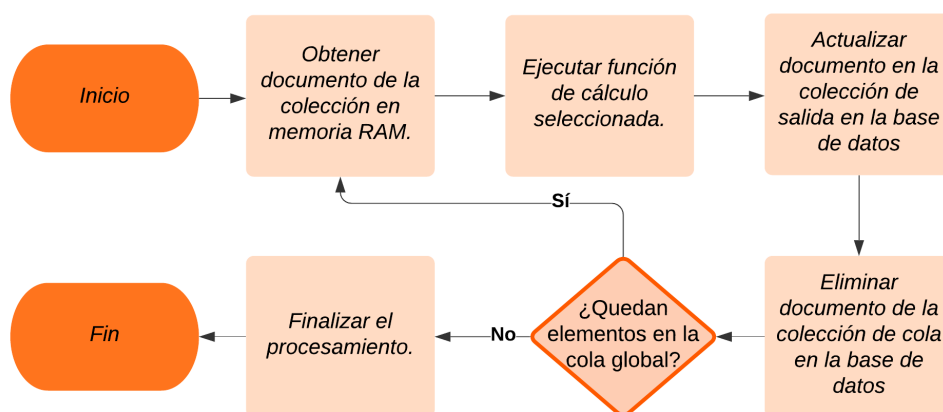


Figura 4.2: Flujo de los hilos secundarios.

Al ser un modelo de multiproceso simétrico, los procesos comparten uso de memoria. Podemos ver todos los componentes del modelo y sus interacciones en la figura 4.3 .

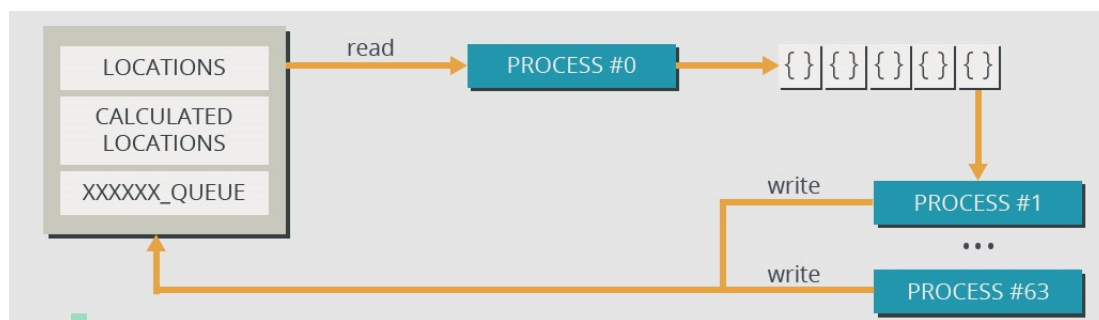


Figura 4.3: Arquitectura del multiproceso simétrico.

En cuanto a la implementación de este modelo, se realizó en el lenguaje de programación “Python”, debido a que el módulo de cálculo se escribió en dicho lenguaje, dentro de esta implementación se hizo necesario hacer de la cantidad de núcleos un parámetro configurable, esto debido a que no se conocía en ese momento la capacidad del servidor que se iba a contratar.

4.2. Uso del módulo de cálculo

El módulo de cálculo dispone de métodos, que se encargan de obtener los datos necesarios para calcular la lista de canales disponibles, asignados a una pareja coordenada. Dichos métodos conforman entre ellos el flujo de ejecución mostrado en la figura 4.4.

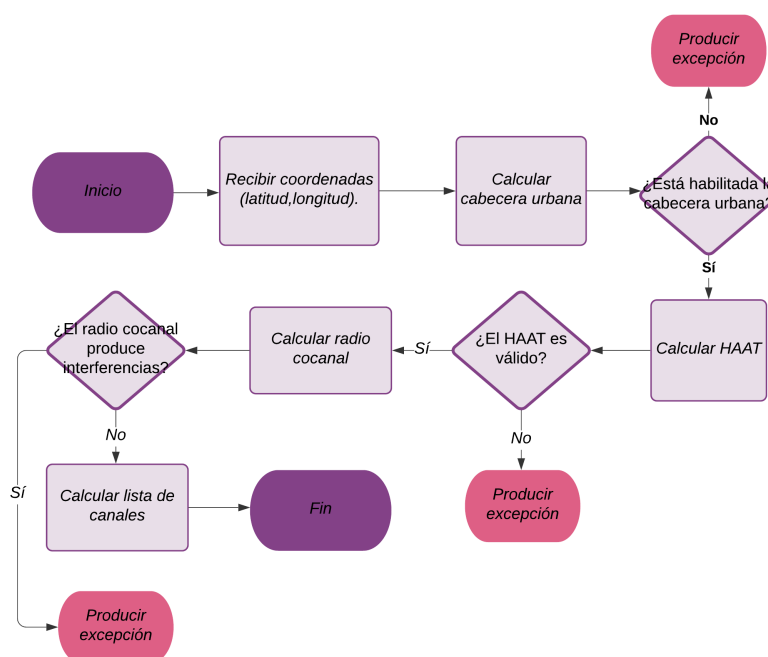


Figura 4.4: Flujo del módulo de cálculo.

A partir del flujo de ejecución, para verificar parcialmente los resultados obtenidos y tener certeza de estar realizando el proceso correctamente, el flujo de cálculo se dividió en las siguientes cuatro etapas:

- Cálculo del sitio.
- Cálculo del HAAT.
- Cálculo del Radio Cocanal (CCR).
- Cálculo de la lista de canales.

El anterior flujo se diseñó para reducir al mínimo el número de llamados por cada método, ya que anteriormente era necesario repetir todos los cálculos para obtener los datos uno por uno, adicional a esto cada etapa será ejecutada para todos los puntos geográficos antes de pasar a la próxima, los condicionales se eliminaron puesto que son realizados en la base de datos durante la creación de las colas de procesamiento, este será el flujo general de operaciones a seguir.

Siendo el objetivo lo anteriormente mencionado, se adaptará el módulo de cálculo para funcionar como una librería, a pesar de que su diseño inicial fue el de un servicio, de esta forma la ejecución se realiza de la manera más limpia posible, liberando algunos recursos usados para su ejecución como servicio. De esta manera cada proceso secundario dispondrá de su propia instancia del módulo de cálculo y de su propio cliente a la base de datos, para evitar cualquier tipo de retardo no deseado.

Debido a que el módulo de cálculo fue escrito en el lenguaje de programación “Python”, por cuestiones de compatibilidad, se realizó la implementación en el mismo lenguaje, los cálculos se realizaron con las configuraciones que tiene la instalación del módulo que se realizó en la infraestructura de la ANE, en cuanto a la asignación de canales a nivel nacional, parámetros de cálculo, entre otros. En esta implementación se realizaron algunos parámetros configurables como el nombre de la base de datos y las colecciones, con el fin de realizar pruebas con otras bases de datos.

Con el fin de asegurar el proceso, al finalizar cada etapa se realizó una copia de respaldo de la base de datos, para garantizar la integridad de los datos.

4.2.1. Cálculo del sitio

El sitio es calculado por el módulo de cálculo mediante una consulta al Shapefile que contiene la división política de Colombia, de esta manera ubica el punto geográfico consultado con el objetivo de verificar si está habilitado para la tecnología TVWS, además de obtener información acerca del tipo de entorno, se extraen los códigos DANE y la “Clase”, este dato nos indica si la coordenada se encuentra en un entorno rural o urbano, en la figura 2.7 podemos ver la totalidad de los atributos que contiene el Shapefile, a partir de los datos obtenidos, se construye el objeto “place” cuya estructura se detalla en el código 3.3.

Al comenzar esta etapa, se completa el objetivo de la sección 2.4, el cual es descartar los puntos fuera de Colombia, para crear la cola de cálculo de sitio se realiza un filtro sobre los datos de la colección que contiene el ráster completo (“locations”), esto deja como resultado una colección con **28 826 716** puntos como se muestra en la figura 4.5, lo cual es una reducción considerable con respecto al valor inicial de los datos, esta cifra representa los puntos del ráster que se encuentran dentro de Colombia, los demás puntos son simplemente el relleno necesario para que el ráster conserve la forma rectangular, este procedimiento descartó más de 30 millones de puntos que se encontraban fuera de Colombia.

```
place_queue  
> db.place_queue.count()  
28826716
```

Figura 4.5: Cantidad de documentos de la colección “place_queue”.

El filtro que se aplicó se hizo sobre el dato de la banda (elevación), únicamente se encuentran dentro de Colombia los datos de elevación iguales o mayores a cero, ya que es una magnitud, esto se debe al recorte que se realizó previamente. Con los datos resultantes se crea una colección en la base de datos llamada “place_queue” con la estructura que se observa en la figura 4.6.

```
> db.place_queue.find().pretty()  
{  
  "_id" : ObjectId("5f61ab12fd47ea155dfc3d13"),  
  "longitude" : -71.67152544633902,  
  "latitude" : 12.456413142988314,  
  "band" : 0  
}
```

Figura 4.6: Estructura de la cola de sitio.

Posterior a la creación de la cola “place_queue”, el proceso principal iniciará los procesos secundarios para comenzar con el cálculo de los datos que contiene dicha colección, esto da como resultado la creación de la colección de salida (“calculated_locations”), que contiene los puntos a los que se les realizó el cálculo de sitio como se puede ver en la figura 4.7.

```
db.calculated_locations.find().limit(1).pretty()

  "_id" : ObjectId("5f61ab12fd47ea155dfc3d16"),
  "latitude" : 12.456413142988314,
  "longitude" : -71.66613907833901,
  "place" : {
    "is_valid" : true,
    "city" : {
      "cod_dane" : "4444847",
      "cod_dep" : "44",
      "cod_mun" : "44847",
      "nom_dep" : "LA GUAJIRA",
      "nom_mun" : "Uribia",
      "clase" : "3",
      "verify" : true
    }
  },
```

Figura 4.7: Estructura de la colección de salida etapa 1.

4.2.2. Cálculo del HAAT

El HAAT o altura efectiva, se calcula dentro del módulo empleando el ráster de elevación, ubicando un anillo que tiene como centro la coordenada consultada, y promediando las elevaciones que se encuentran en la intersección del ráster y el anillo, para posteriormente hallar la diferencia entre el promedio de elevaciones y la elevación correspondiente al punto geográfico consultado, en la figura 4.8 se puede observar el anillo y de fondo el ráster de elevación. A partir de la información obtenida, se construye el objeto “haat”, su estructura se especifica en el código 3.4.

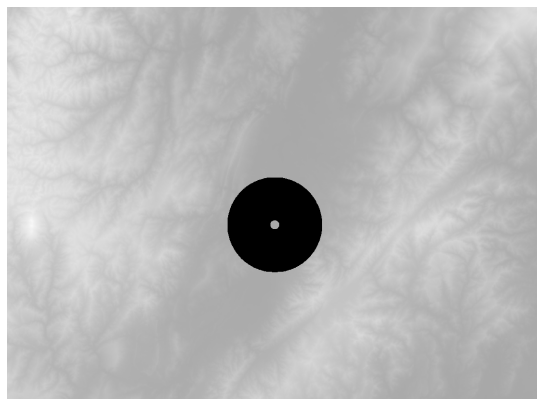


Figura 4.8: Anillo sobre raster.

En esta etapa se realiza el cálculo del HAAT para todos los puntos dentro de Colombia, para esto el proceso principal crea en la base de datos una colección de cola llamada “haat_queue”, esta colección de cola tiene la estructura final que tenía la colección de salida en la etapa anterior (figura 4.7) y **28 826 716** puntos como se observa en la figura 4.9. Al finalizar esta etapa, la colección de salida tiene la estructura que se observa en la figura 4.10.

```
> db.haat_queue.count()
28826716
```

Figura 4.9: Cantidad de documentos de la colección “haat_queue”.

```
> db.calculated_locations.find().limit(1).pretty()
{
  "_id" : ObjectId("5f61ab12fd47ea155dfc3d16"),
  "latitude" : 12.456413142988314,
  "longitude" : -71.66613907833901,
  "place" : {
    "is_valid" : true,
    "city" : {
      "cod_dane" : "4444847",
      "cod_dep" : "44",
      "cod_mun" : "44847",
      "nom_dep" : "LA GUAJIRA",
      "nom_mun" : "Uribia",
      "clase" : "3",
      "verify" : true
    }
  },
  "haat" : {
    "haat_value" : 1194.9423828125,
    "haat_altitude" : 12.042252540588379,
    "haat_mean" : -1182.900146484375,
    "haat_hatvws" : 1244.9423828125,
    "is_haat_valid" : false
  }
}
```

Figura 4.10: Estructura de la colección de salida etapa 2.

4.2.3. Cálculo del Radio Cocanal (CCR)

El radio cocanal se obtiene consultando una matriz que contiene datos referentes al estándar UIT-Rp 1546-6, dependiendo el método de cálculo, se consulta la figura 9 o la figura 11 de dicho estándar. Debido a que el módulo de cálculo cuenta con dos métodos de consulta, a partir de esta etapa los cálculos se deben realizar dos veces, ya que los datos cambian sustancialmente entre los dos métodos de cálculo. En consecuencia la información se guarda dentro del método correspondiente (código 3.5).

En esta etapa se realiza el cálculo del radio cocanal, para todos los puntos que tengan un “HAAT” válido, en este caso, se considera válido para valores por debajo de **800** metros, ya

que así está configurado el módulo de cálculo, para esta etapa se crea la colección “ccr_queue” y en esta clasificaron **28 646 411** puntos como se observa en la 4.11, la estructura que tiene dicha colección es la que se puede ver en la figura 4.10.

```
> db.ccr_queue.count()
28646411
```

Figura 4.11: Cantidad de documentos de la colección “ccr_queue”.

Al finalizar la etapa, la estructura de la colección de salida es la que se puede observar en la figura 4.12.

```
{
  "_id" : ObjectId("5f61ab13fd47ea155dfc7209"),
  "latitude" : 12.452822230988314,
  "longitude" : -71.64459360633901,
  "place" : {
    "is_valid" : true,
    "city" : {
      "cod_dane" : "4444847",
      "cod_dep" : "44",
      "cod_mun" : "44847",
      "nom_dep" : "LA GUAJIRA",
      "nom_mun" : "Uribia",
      "clase" : "3",
      "verify" : true
    }
  },
  "haat" : {
    "haat_value" : 796.1495971679688,
    "haat_altitude" : 0,
    "haat_mean" : -796.1495971679688,
    "haat_hatvws" : 846.1495971679688,
    "is_haat_valid" : true
  },
  "method_1" : {
    "ccr" : [
      115.7,
      106.8,
      104.3,
      104.3,
      106.4
    ],
    "is_valid_header" : true
  },
  "method_2" : {
    "ccr" : [
      88.5,
      83.5,
      82,
      82,
      83.3
    ],
    "is_valid_header" : true
  }
}
```

Figura 4.12: Estructura de la colección de salida etapa 3.

4.2.4. Cálculo de la lista de canales

La lista de canales se calcula a partir del vector cocanal y del vector adyacente, el módulo de cálculo construye esta lista con los canales del catorce al cincuenta y uno, asignando un cero o uno dentro del objeto para cada canal, al contrario de los datos booleanos en este caso el cero indica que un canal está disponible, y el uno indica que está ocupado, como se observa en la figura 4.13, la información es almacenada dentro del objeto “method_x” cuya estructura se puede observar en el código 3.5.

```
{"14":1,"15":1,"16":1,"17":1,"18":1,"19":1,"20":1,"21":0,"22":0,"23":0,"24":0,"25":0,"26":0,"27":1,
"28":1,"29":1,"30":1,"31":1,"32":1,"33":1,"34":0,"35":1,"36":0,"37":1,"38":0,"39":0,"40":0,"41":1,
"42":0,"43":0,"44":0,"45":1,"46":1,"47":1,"48":1,"49":1,"50":1,"51":1}
```

Figura 4.13: Ejemplo de lista de canales disponibles.

Al comenzar ésta etapa, se clasifican los puntos que no realicen interferencias dentro de la colección de cola “channels_queue”, para el calculo de esta etapa clasificaron **26 562 707** puntos como se observa en la figura 4.14, en esta etapa la colección de cola tendrá la estructura que se ve en la figura 4.12, al finalizar esta etapa, se completarán los datos en la colección de salida la cual tomará la estructura que se ve en la figura 4.15, para efectos de visualización solo se incluyó el canal catorce, pero esta lista abarca hasta el canal cincuenta y uno.

```
> db.channels_queue.count()
26562707
```

Figura 4.14: Cantidad de documentos de la colección “channels_queue”.

```
{
  "_id" : ObjectId("5f61ab13fd47ea155dfc7209"),
  "latitude" : 12.452822230988314,
  "longitude" : -71.64459360633901,
  "place" : {
    "is_valid" : true,
    "city" : {
      "cod_dane" : "4444847",
      "cod_dep" : "44",
      "cod_mun" : "44847",
      "nom_dep" : "LA GUAJIRA",
      "nom_mun" : "Uribia",
      "clase" : "3",
      "verify" : true
    }
  },
  "haat" : {
    "haat_value" : 796.1495971679688,
    "haat_altitude" : 0,
    "haat_mean" : -796.1495971679688,
    "haat_hatvws" : 846.1495971679688,
    "is_haat_valid" : true
  },
  "method_1" : {
    "ccr" : [
      115.7,
      106.8,
      104.3,
      104.3,
      106.4
    ],
    "is_valid_header" : true,
    "channels" : {
      "14" : 1
    },
    "all_channels" : 18
  },
  "method_2" : {
    "ccr" : [
      88.5,
      83.5,
      82,
      82,
      83.3
    ],
    "is_valid_header" : true,
    "channels" : {
      "14" : 1
    },
    "all_channels" : 18
  }
}
```

Figura 4.15: Estructura de la colección de salida etapa 4.

Capítulo 5

Construcción de los mapas

Una vez procesados todos los datos, para la presentación de los resultados, se requiere ver gráficamente la información procesada y confinada en el banco de datos. Como es sabido, los ráster son una muy buena alternativa en los casos en los que se necesita ver datos georeferenciados. Debido a que los datos extraídos hacen referencia a píxeles de un ráster previamente reproyectado, se realizará una reconstrucción reemplazando la banda por el dato que se quiera visualizar.

En este orden de ideas, se construirá un ráster georeferenciado, con un valor de banda basado en algún campo del banco de datos generado en el capítulo anterior. Para realizar esto, es necesario exportar algún campo de la base de datos a un archivo con algún sistema de coordenadas. Como vimos anteriormente, el SRC más sencillo se denomina XYZ, como característica principal este sistema de coordenadas se puede construir en un archivo CSV, siempre y cuando tenga la estructura adecuada. Dicha estructura debe ser la de un archivo CSV que tiene como separador el carácter espacio, no debe tener encabezados y debe incluir los datos longitud, latitud y valor de banda en este orden. La estructura mencionada se puede ver en la figura 5.1.

```
-76.2660973503390096 8.94988757498831333 -32767  
-76.2643018943390132 8.94988757498831333 -32767
```

Figura 5.1: Estructura CSV driver XYZ.

La estructura de datos correspondiente al sistema de coordenadas XYZ, a pesar de ser tan simple, tiene la particularidad de que los datos deben encontrarse ordenados en el archivo CSV dependiendo de la coordenada geográfica que represente. El primer criterio de ordenamiento es el campo latitud, este campo se debe ubicar en el archivo de manera descendente, el segundo criterio es el campo longitud, el cual debe estar dispuesto en el archivo en un orden ascendente. Al verlo de forma gráfica en un ráster, el orden enunciado anteriormente se ve como el de la figura 5.2.

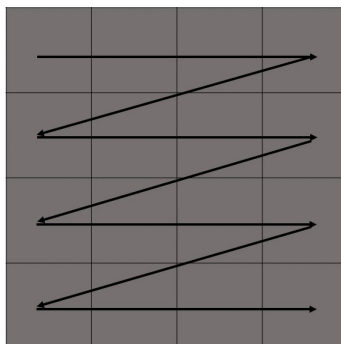


Figura 5.2: Sentido de ordenamiento de un archivo CSV driver XYZ.

Una vez exportados, ordenados y confinados en un archivo CSV en el sistema de coordenadas XYZ, el paso a seguir es el de cambiar la proyección de los datos, debido a que la proyección XYZ no es la adecuada para la zona geográfica en la que se encuentra ubicada Colombia. Como se vio en la sección 2.1 el sistema de coordenadas que corresponde a la zona de Colombia es “EPSG:4686 - MAGNA-SIRGAS” por tanto se debe realizar una reproyección a este SRC. Una vez realizados estos pasos, se debería disponer de un ráster listo para su visualización.

Se explicará el proceso únicamente para el ráster de “HAAT”, en el caso de los ráster de cantidad de canales en método uno y método dos, aunque el proceso es el mismo se hará uso de una automatización realizada por el autor para llegar rápidamente a los mapas.

Antes de comenzar con la exportación de los datos, se deben agrupar en una sola colección y ordenarlos, para esto se ejecutarán los comandos que se pueden ver en el código 5.1 dentro de la consola de MongoDB, esta ejecución puede tardar bastante ya que son millones de documentos.

```

1 db.locations.aggregate([{$out: 'raster1'}])
2 db.calculated_locations.find().forEach(
3     function(doc){
4         db.raster1.update({_id: doc._id},{$set: doc})
5     }
6 )
7 db.raster1.aggregate([{$sort: {'latitude': -1, 'longitude': 1}}, {$out: '
    raster1'}], {allowDiskUse: true})
8 db.raster1.drop()
```

Código 5.1: Comandos para agrupar y ordenar los datos

El procedimiento para agrupar los datos, consiste en crear una colección (“raster1”) igual a la

colección que contiene todos los datos del ráster incluyendo los datos de relleno (“locations”), luego actualizar los puntos que están dentro de Colombia con los datos calculados (colección “calculated_locations”). Con referencia al ordenamiento, ejecutar el algoritmo incluido en MongoDB y guardar la salida en otra colección (“raster”), la colección “raster1” se utilizó como colección temporal, por tanto se elimina al finalizar el procedimiento.

Luego de realizar este procedimiento, se debería tener una colección llamada “raster” (figura 5.3), a partir de esta colección, se exportarán los datos necesarios para construir el ráster de HAAT.

```
> show collections
calculated_locations
locations
raster
```

Figura 5.3: Colección de datos agrupados y ordenados.

A continuación se exportarán los datos correspondientes a latitud, longitud y valor del “HAAT”, como banda a visualizar. Para realizar esta tarea, se empleará la herramienta “Mongoexport” de MongoDB, en el código 5.2 se puede ver la instrucción con sus parámetros configurados, en la tabla 5.1 se observan los parámetros utilizados y la descripción de cada uno.

```
1 mongoexport -d col200 -c raster --type=csv --fields longitude,latitude,
  method_2.all_channels --out raster_haat.csv
```

Código 5.2: Instrucción para exportar los datos de MongoDB.

Parámetro	Descripción
d	Nombre de la base de datos.
c	Nombre de la colección.
type	Tipo de archivo.
fields	Nombres de los campos a exportar en orden.
out	Nombre del archivo de salida.

Tabla 5.1: Parámetros de entrada: Instrucción de exportación.

La salida de esta instrucción se puede ver en la figura 5.4. Una vez ejecutada la instrucción se obtiene un archivo CSV con la estructura que se observa en la figura 5.5.

```
2021-03-19T13:49:49.169-0500 [#####] col200.raster 62944007/62944007 (100.0%)
2021-03-19T13:49:49.169-0500 exported 62944007 records
```

Figura 5.4: Salida de mongoexport.

```
(platebco) juan@DESKTOP-RAKFQH0:/mnt/c/Use
longitude,latitude,haat.haat_value
-79.00775866233901,12.456413142988314,
-79.005963206339,12.456413142988314,
-79.00416775033901,12.456413142988314,
-79.00237229433901,12.456413142988314,
-79.00057683833901,12.456413142988314,
-78.998781382339,12.456413142988314,
-78.99698592633901,12.456413142988314,
-78.99519047033901,12.456413142988314,
-78.99339501433901,12.456413142988314,
```

Figura 5.5: Archivo CSV entregado por MongoDB.

Este archivo CSV incluye un encabezado en la primera línea, y su separador es el carácter coma, como se mencionó anteriormente, el driver XYZ solo admite archivos sin encabezado, y que tengan como separador el carácter espacio. Para esto se empleará el editor de flujo sed, la instrucción mostrada en el código 5.3. Esta instrucción tiene varios parámetros de ejecución, con estos parámetros se eliminan los encabezados, se rellenan los datos inexistentes, se cambia el separador y por último, en el caso de que el dato sea un booleano, lo reemplaza a cero y uno respectivamente, todo esto para que se pueda construir un ráster con cualquier campo que se encuentre en la base de datos.

```
1 sed -i -e '1d' -e 's/,$/, -32767/' -e 's/,/\ /g' -e 's/true/1/g' -e 's/
false/0/g' raster_haat.csv
```

Código 5.3: Instrucción sed con múltiples parámetros de ejecución.

Después de ejecutar esta instrucción, el archivo CSV ya está listo para ser reproyectado, para esto se utilizará de nuevo la herramienta “gdal_translate” para cambiar el driver a “.TIFF”, con la proyección “EPSG:4686 - MAGNA-SIRGAS” que corresponde a Colombia. La instrucción que se observa en el código 5.4.

```
1 gdal_translate -of GTiff -a_srs EPSG:4686 -a_nodata -32767 raster_haat.
csv haat_value.tif
```

Código 5.4: Instrucción de reproyección.

En la tabla 5.2 se pueden ver a detalle los parámetros configurados para esta ejecución con sus respectivas descripciones.

Parámetro	Descripción
of	Formato de salida.
a_srs	Sistema de referencia de salida.
a_nodata	Valor de dato nulo.

Tabla 5.2: Parámetros de entrada: Instrucción de reproyección.

La salida de esta instrucción se puede ver en la figura 5.6, allí se evidencia que la herramienta reconoce el tamaño del ráster de entrada, también se muestra al avance del procesamiento progresivamente.

```
Input file size is 6774, 9292
0...10...20...30...40...50...60...70...80...90...100 - done.
```

Figura 5.6: Salida de la instrucción translate.

Al terminar el procedimiento enunciado anteriormente, se debería tener un ráster como el de la figura 5.7, cabe aclarar que la visualización se configura en la herramienta que se esté trabajando para este fin, sus propiedades se pueden observar en la figura 5.8, como se puede ver son muy similares a las del ráster de elevación remuestreado y recortado (figura 2.26).

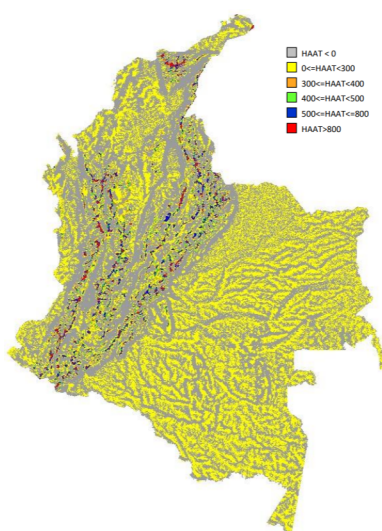


Figura 5.7: Raster de HAAT de Colombia.

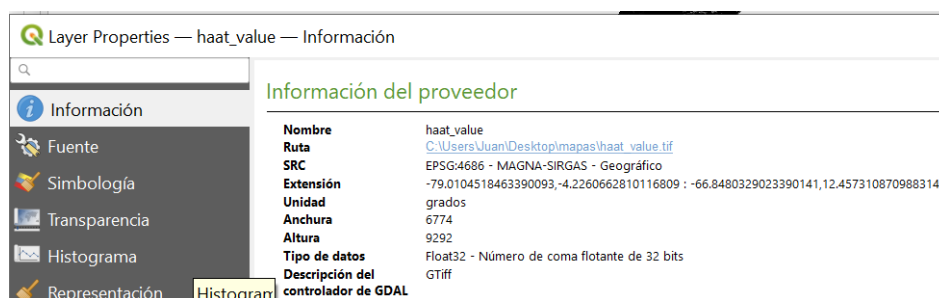


Figura 5.8: Propiedades del Ráster de HAAT de Colombia.

Teniendo claro este proceso, fue automatizado por el autor para la construcción de los mapas de método uno y método dos, la herramienta escrita en bash se encarga de construir un ráster con el campo de una colección en MongoDB indicado por el usuario. Los parámetros se toman por posición, en la tabla 5.3 se pueden ver los parámetros correspondientes a las cuatro posiciones.

Parámetro	Descripción
Posición 1	Nombre de la base de datos.
Posición 2	Nombre de la colección.
Posición 3	Campo de la colección.
Posición 4	Nombre del ráster de salida.

Tabla 5.3: Parámetros de entrada: Instrucción de conversión de MongoDB a Tif.

La instrucción ya preparada que permite construir el ráster se puede ver en el código 5.5.

```
1 ./mongo_to_tif col200 raster method_1.all_channels
   method_1_all_channels.tif
```

Código 5.5: Herramienta mongo_to_tif.

Al ejecutar la instrucción la consola mostrará una salida como la que se observa en la figura 5.9.

```
2021-03-19T10:14:56.901-0500 [#####] col200.raster 62944007/62944007 (100.0%)
2021-03-19T10:14:56.901-0500 exported 62944007 records

Input file size is 6774, 9292
0...10...20...30...40...50...60...70...80...90...100 - done.
```

Figura 5.9: Salida de la herramienta mongo_to_tif.

Como resultado de lo anterior, se obtiene el ráster de cantidad de canales para método uno que se muestra en la figura 5.10, sus propiedades se pueden ver en la figura 5.11.

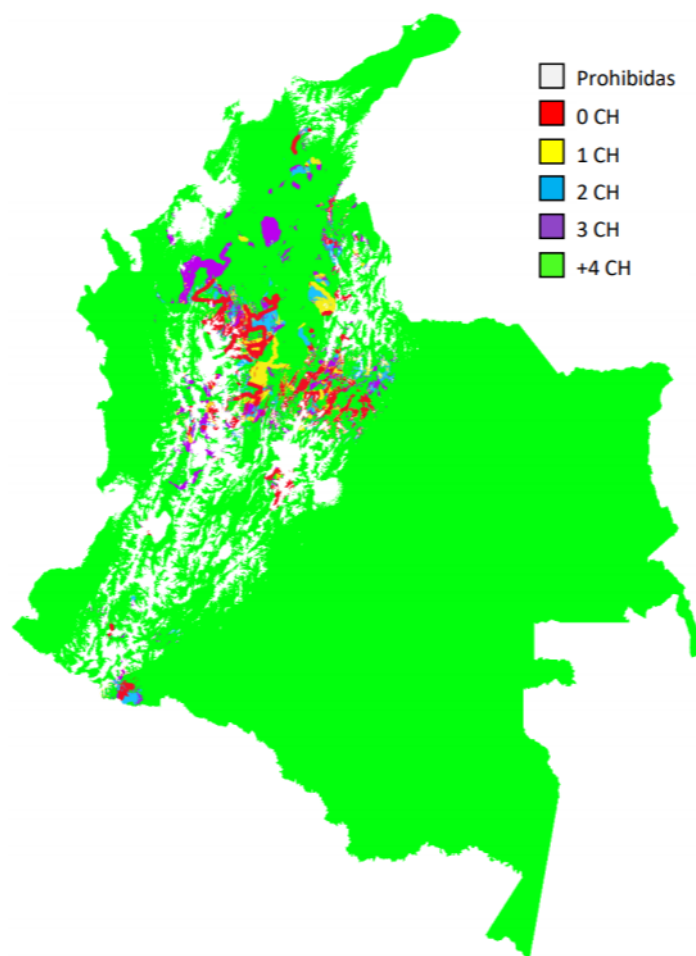


Figura 5.10: Ráster de cantidad de canales para el método uno de Colombia.



Figura 5.11: Propiedades del Ráster de cantidad de canales para método uno de Colombia.

Al replicar el proceso para el método dos se obtuvo el ráster de cantidad de canales para el método dos que se observa en la figura 5.12, sus propiedades se pueden ver en la figura 5.13.

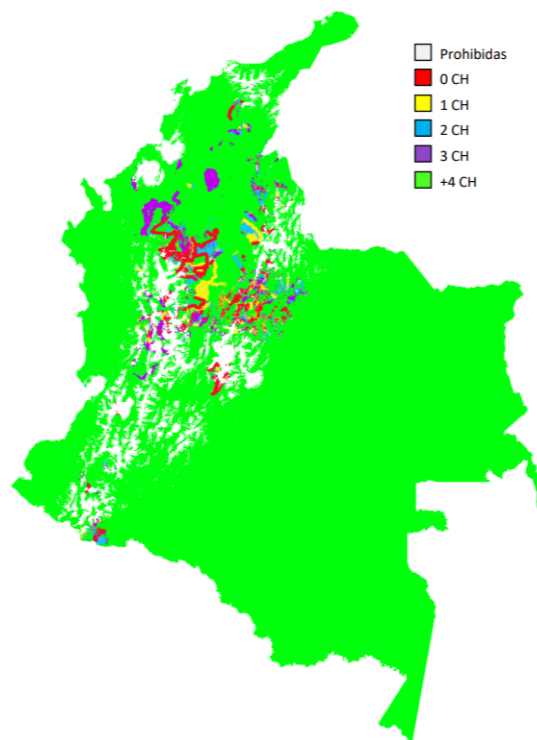


Figura 5.12: Ráster de cantidad de canales para el método dos de Colombia.

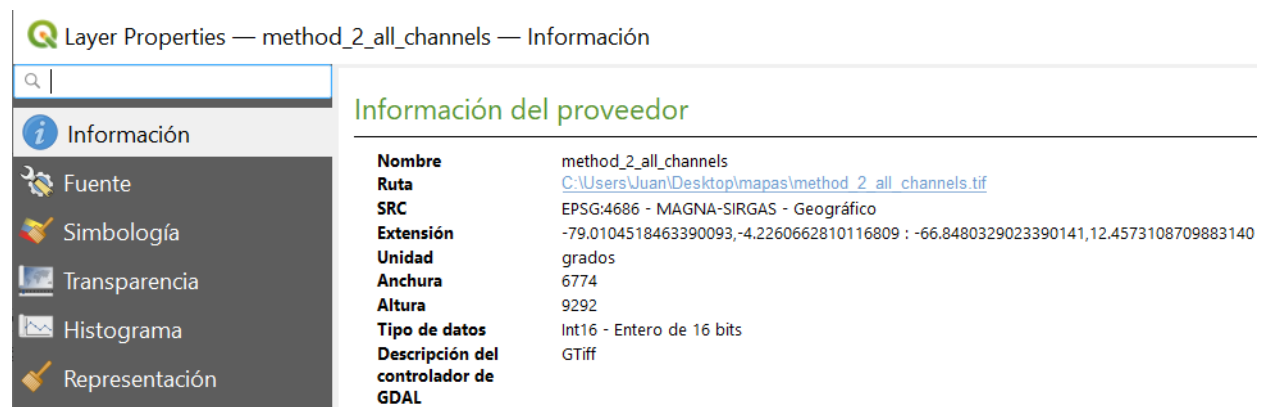


Figura 5.13: Propiedades del Ráster de cantidad de canales para método dos de Colombia.

Capítulo 6

Elaboración de los conceptos técnicos

Con el fin de realizar un análisis un poco más exhaustivo del estado actual de la tecnología TVWS en Colombia, el equipo de trabajo de la Universidad Santo Tomás, llegó a un acuerdo con la Agencia Nacional del Espectro, en el cual se pactó la entrega de tres informes de alcance nacional (Conceptos Técnicos). Esto con la intención de tener información exacta de cada uno de los departamentos y municipios de Colombia en caso de necesitar una consulta más específica.

En el marco dado, un concepto técnico se compone de tres apartados, los cuales son: la introducción, el apartado de información por departamentos/municipios y por último las conclusiones/recomendaciones.

Este capítulo abarcará el apartado de información por departamentos/municipios de Colombia, debido a que este proceso es el más extenso en cuanto a cantidad de páginas, y debe ser uniforme para todo el territorio nacional. Al momento de realización de este procedimiento aún no se conocía el contenido del documento, por tanto se diseñó un modelo bastante flexible con el fin de facilitar la adaptación del mismo a las necesidades que pudieran ir surgiendo sobre la marcha.

Se seleccionó la herramienta Microsoft Office para la gestión de informes, esta implementación está basada en una ejecución de “Macros” anidadas las cuales realizan una función específica definida por el usuario, son bastante flexibles, y permiten la comunicación entre los diferentes productos de Office para utilizar algunas de las funcionalidades que ofrecen, dichas macros se programan en el lenguaje Visual Basic for Applications(VBA)[8].

Se mostrará el procedimiento realizado para el informe de HAAT, el proceso es replicado de la misma manera para los informes de método uno y método dos. Los insumos requeridos para elaborar un concepto técnico se definieron como:

- Archivo de Introducción.

- Cálculos de cantidad de puntos para alimentar los informes departamento/municipio.
- Plantilla base de informe para departamento.
- Plantilla base de informe para municipio.
- Archivo de Conclusiones/Recomendaciones.

Los archivos de introducción y de conclusiones/recomendaciones se separaron del proceso puesto que estas partes del concepto técnico no se componen de un proceso repetitivo, y por lo tanto se deben realizar de manera independiente, para posteriormente ser adicionados al resto del documento.

Lo primero es obtener los datos que alimentarán los informes realizados posteriormente, para ello se implementó un programa escrito en el lenguaje de programación python, que se conecta a la base de datos y realiza cálculos porcentuales para cada municipio y cada departamento, dicho programa genera un archivo tipo CSV para cada municipio y otro para cada departamento, este procedimiento se realiza una vez por cada concepto técnico, porque cada uno de ellos se basa en un dato diferente contenido en la base de datos.

Para el concepto técnico de HAAT los archivos tienen las estructuras que se observan en las figuras 6.1 y 6.2.

cod_dep	name_dep	no_points	allowed	not_allowed	haat<0	0<=haat<300	300<=haat<400	400<=haat<500	500<=haat<800
11	SANTA FE DE	41330	41303	27	17544	13744	4946	3141	1928
13	BOLIVAR	679347	679172	175	429728	227779	12142	5882	3641
urban_no_points	urban_allowed	urban_not_allowed	urban_haat<0	urban_0<=haat<300	urban_300<=haat<400	urban_400<=haat<500	urban_500<=haat<800	rural_no_points	rural_allowed
9611	9611	0	9270	331	10	0	0	31719	31692
2672	2672	0	2165	507	0	0	0	676675	676500
urban_500<=haat<800	rural_no_points	rural_allowed	rural_not_allowed	rural_haat<0	rural_0<=haat<300	rural_300<=haat<400	rural_400<=haat<500	rural_500<=haat<800	
0	31719	31692	27	8274	13413	4936	3141	1928	
0	676675	676500	175	427563	227272	12142	5882	3641	

Figura 6.1: Estructura CSV haat_stats departamento.

cod_mun	name_mun	name_dep	no_points	allowed	not_allowed	haat<0	0<=haat<300	300<=haat<400	400<=haat<500	500<=haat<800	urban_no_points	urban_allowed	urban_not_allowed	urban_haat<0	urban_0<=haat<300	urban_300<=haat<400	urban_400<=haat<500	urban_500<=haat<800	rural_no_points	rural_allowed	rural_not_allowed	rural_haat<0	rural_0<=haat<300	rural_300<=haat<400	rural_400<=haat<500	rural_500<=haat<800
11001	Santafe de Bc	SANTA FE DE	41330	41303	27	17544	13744	4946	3141	1928	9611	9611	0	9270	331	10	0	0	31719	31692	27	8274	13413	4936	3141	1928
13001	Cartagena	BOLIVAR	16123	16123	0	10658	5465	0	0	0	0	0	0	0	0	0	0	0	14811	14811	0	9526	5285	0	0	0
400<=haat<500 500<=haat<800 urban_no_points urban_allowed urban_not_allowed urban_haat<0 urban_0<=haat<300 urban_300<=haat<400 urban_400<=haat<500 urban_500<=haat<800 rural_no_points rural_allowed rural_not_allowed rural_haat<0 rural_0<=haat<300 rural_300<=haat<400 rural_400<=haat<500 rural_500<=haat<800																										

Figura 6.2: Estructura CSV haat_stats municipio.

Una vez obtenidos los archivos CSV, se debe revisar la plantilla dada para departamento y municipio, dependiendo de la estructura, se deben realizar las macros conseguir la estructura planteada.

Para el concepto técnico de HAAT, en el caso del departamento se definió una plantilla compuesta de tres páginas, la primera corresponde a una página de presentación la cual contiene el nombre del departamento (figura 6.3).



Figura 6.3: Plantilla para departamento página 1.

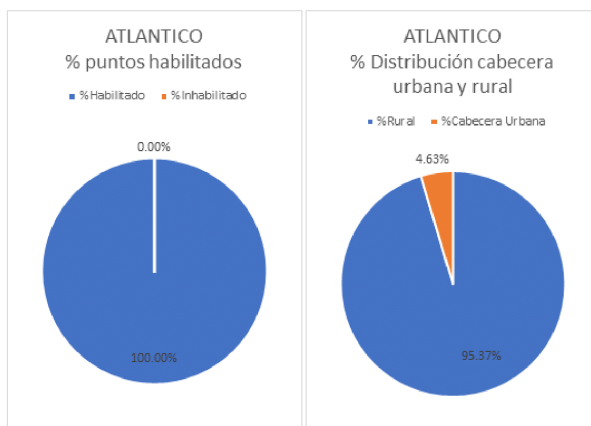
La segunda contiene dos gráficas, la primera muestra información acerca de la cantidad de

puntos habilitados por HAAT en el departamento y la segunda información de la distribución del departamento entre zona rural y urbana, luego algunas conclusiones dependiendo de los datos obtenidos del archivo CSV.

ANÁLISIS DE HAAT COLOMBIA

ANÁLISIS DEL HAAT PARA EL DEPARTAMENTO

«nom_dep»



Porcentaje de puntos dentro del departamento de «nom_dep» distribuidos en cabecera urbana y rural, habilitados y no habilitados.

De los «NoPuntos» puntos que contiene el departamento de «nom_dep», el **!Error de sintaxis**, « está ubicado en zona rural y el **!Error de sintaxis**, « se encuentran ubicados en una cabecera urbana municipal. De los «NoPuntos_rural» ubicados en zona rural están habilitados por HAAT para el uso de espacios en blanco en TV el **!Error de sintaxis**, «, y de los «NoPuntos_urb» de las cabeceras urbanas del departamento están habilitados por HAAT el **!Error de sintaxis**, « para el uso de espacios en blanco en TV.

En general dentro del departamento de «nom_dep», el **!Error de sintaxis**, « de los puntos están habilitados para el uso de espacios en blanco en TV el **!Error de sintaxis**, « se encuentran inhabilitados por HAAT para el uso de espacios en blanco en TV.

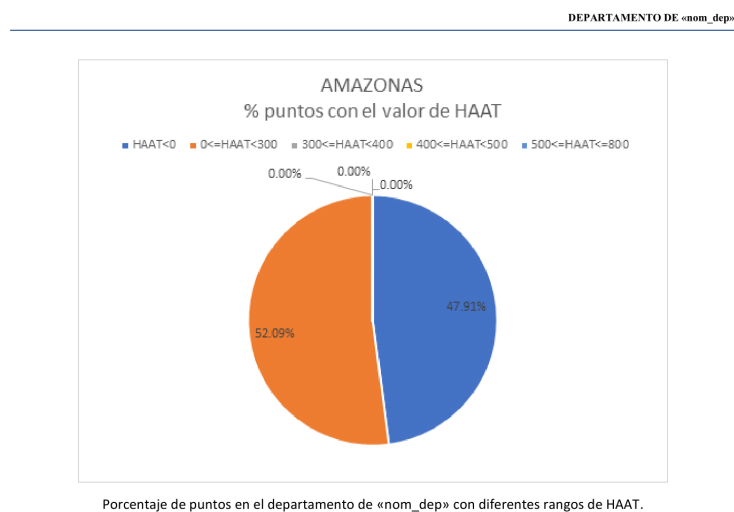
En cuanto a los valores de HAAT para los «Habilitado» puntos habilitados dentro de departamento, estos se encuentran en los siguientes rangos:

- El **!Error de sintaxis**, « de los puntos habilitados del municipio tienen un HAAT superior o igual a 400m. Se debe recordar que valores superiores a 800m no están habilitados.
- El **!Error de sintaxis**, « se encuentra en el rango entre 300m y 400m.
- El **!Error de sintaxis**, « de los HAAT en el municipio están en el rango de 0 a 300m.
- El **!Error de sintaxis**, « de los HAAT son inferiores a 0m.

2

Figura 6.4: Plantilla departamento página 2.

La tercera página contiene una gráfica que muestra la distribución del HAAT a nivel departamental.



A continuación, se presenta la tabla resumen de los municipios que contiene el departamento de «nom_dep». La columna de porcentaje de puntos representa la contribución porcentual del municipio al departamento en cuanto a número de puntos, en las demás columnas de la tabla el porcentaje están referido al número de puntos del propio municipio, de manera que se puede establecer la distribución en zona rural y cabecera urbana, al igual que el porcentaje de puntos habilitados por HAAT dentro del municipio.

3

Figura 6.5: Plantilla departamento página 3.

Adicionalmente se debe adicionar una tabla que contenga un resumen de todos los municipios del departamento en cuestión, calculando los porcentajes basados en la cantidad de puntos del departamento, dicha tabla no se puede incluir en la plantilla, debido a que la longitud de esta es variable, ya que cada departamento tiene una cantidad diferente de municipios. Se puede observar la tabla en la figura 6.6.

Municipio	% Puntos	% Rural	% Cabecera Urbana	% Habilitados Municipio
El Encanto	10,17%	100,00%	0,00%	100,00%
La Chorrera	11,45%	100,00%	0,00%	100,00%
La Pedrera	12,54%	100,00%	0,00%	100,00%
La Victoria	1,36%	100,00%	0,00%	100,00%
Leticia	5,60%	99,88%	0,12%	100,00%
Miriti-Parana	15,40%	100,00%	0,00%	100,00%
Puerto Alegria	7,92%	100,00%	0,00%	100,00%
Puerto Arica	12,43%	100,00%	0,00%	100,00%
Puerto Narino	1,42%	99,97%	0,03%	100,00%
Puerto Santander	13,56%	100,00%	0,00%	100,00%
Tarapaca	8,16%	100,00%	0,00%	100,00%

Figura 6.6: Ejemplo de tabla de departamento.

Para generar los informes con los datos del archivo CSV se empleó la herramienta de correspondencia de Word, que se encarga precisamente de tomar datos de un archivo de Excel y generar informes basados en una plantilla preexistente. Según la configuración de la correspondencia, es posible generar todo en un archivo o en archivos diferentes, en este caso se utilizó la configuración de archivos independientes para adicionar la tabla y los municipios más adelante. Para cada departamento se creó un archivo basado en la plantilla mostrada anteriormente aunque inicialmente se realizó todo el procedimiento con un solo departamento, en la figura 6.7 se puede ver el directorio con el archivos generados por la herramienta de correspondencia.

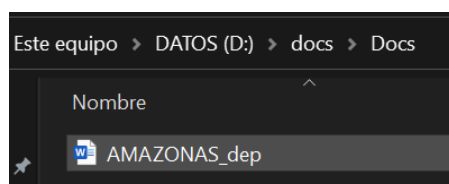


Figura 6.7: Carpeta con informe de un departamento.

Debido a que las gráficas fueron producidas en Excel, se realizó una macro que permite exportarlas y guarda el directorio de la imagen para que la correspondencia las ubique satisfactoriamente. En la figura 6.8 se puede ver el directorio con las respectivas gráficas.

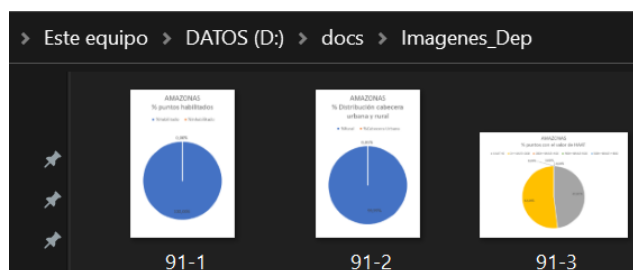


Figura 6.8: Carpeta con las gráficas de un departamento.

Una vez realizada la correspondencia, se procedió a adicionar la tabla, a cada archivo de departamento. Con esto en mente se creó una macro que trabaja con los archivos de departamento y el archivo CSV de municipios. La macro mencionada realiza un filtro sobre el archivo CSV de municipio y la copia al archivo del departamento en cuestión, en la figura 6.9 se puede ver un ejemplo de una tabla de salida.

Municipio	% Puntos	% Rural	% Cabecera Urbana	% Habilitados Municipio
El Encanto	10.17%	100.00%	0.00%	100.00%
La Chorrera	11.45%	100.00%	0.00%	100.00%
La Pedrera	12.54%	100.00%	0.00%	100.00%
La Victoria	1.36%	100.00%	0.00%	100.00%
Leticia	5.60%	99.88%	0.12%	100.00%
Miriti-Parana	15.40%	100.00%	0.00%	100.00%
Puerto Alegria	7.92%	100.00%	0.00%	100.00%
Puerto Arica	12.43%	100.00%	0.00%	100.00%
Puerto Narino	1.42%	99.97%	0.03%	100.00%
Puerto Santander	13.56%	100.00%	0.00%	100.00%
Tarapaca	8.16%	100.00%	0.00%	100.00%

Figura 6.9: Tabla del departamento Amazonas.

En cuanto a los municipios, el equipo de trabajo produjo una plantilla de una página para cada municipio, como la que se observa en la figura 6.10.

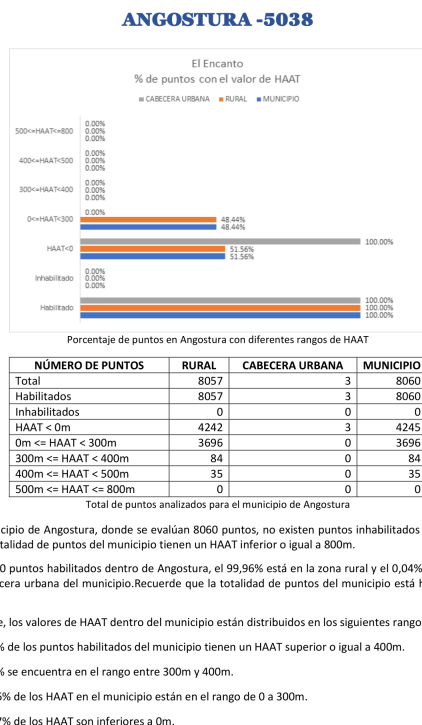


Figura 6.10: Plantilla de municipio.

Para generar el reporte de un municipio, se siguió el mismo patrón, con la salvedad de que en este caso la tabla es del mismo tamaño en todos los casos, por tanto se incluyó en la plantilla, en consecuencia, para este caso no es necesario utilizar la macro de la tabla que se usó anteriormente. Aunque la primera prueba se realizó con un municipio, posteriormente la correspondencia se configuró de manera en que todos los municipios se guardaran en el mismo archivo. En la figura 6.11 podemos ver un ejemplo del reporte de municipio creado.

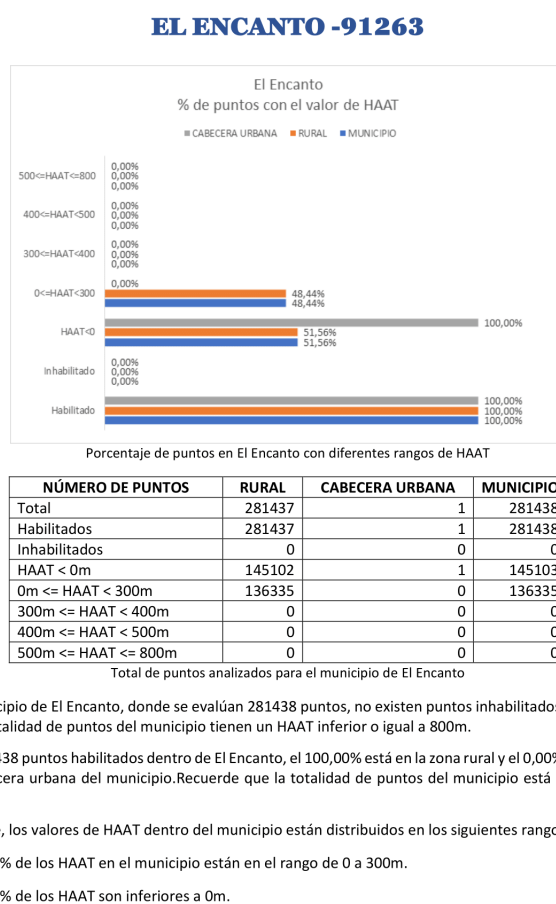


Figura 6.11: Reporte de municipio El Encanto.

Una vez probados los procedimientos para generar los informes de departamento y municipio independientemente, estos procedimientos se agruparon en una sola macro que recibe el nom-

bre del departamento y genera los dos archivos correspondientes (el reporte de departamento y el de los municipios correspondientes). Luego, en excel se creó una macro que recorre el archivo de departamentos, y por cada uno genera los dos archivos, además de ejecutar la macro que agrega la tabla al reporte de cada departamento, por último se creó una macro que concatena los archivos, este flujo general se puede observar en la figura 6.12.

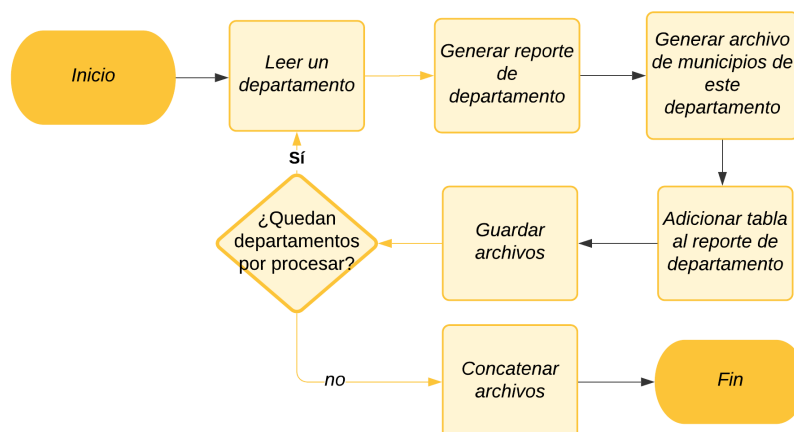


Figura 6.12: Flujo macro principal.

La macro que concatena los archivos lo hace por orden alfabético, para adicional la introducción y las conclusiones/recomendaciones, simplemente se renombran como “A” y “Z” respectivamente, para que se ubiquen correctamente en el documento. Los últimos detalles fueron pulidos a mano, y se obtuvo un concepto técnico con **1 270** páginas como se observa en la figura 6.13.

Concepto Técnico análisis de HAAT (BDEB de la ANE).pdf

1 / 1270

Figura 6.13: Cantidad de paginas concepto técnico HAAT.

A pesar de que las plantillas cambiaron ligeramente para los métodos uno y dos, siguiendo este mismo procedimiento, ligeramente adaptado a las necesidades del reporte, se consiguieron conceptos técnicos con **2 198** (figura 6.14) y **2 383** (figura 6.15) páginas respectivamente.

ANÁLISIS DE CANALES DISPONIBLES TVWS COLOMBIA (Método 1 BDEB de la ANE))....

1 / 2198

Figura 6.14: Cantidad de paginas concepto técnico método uno.

ANÁLISIS DE CANALES DISPONIBLES TVWS COLOMBIA (Método 2 BDEB de la ANE))....

1 / 2383

Figura 6.15: Cantidad de paginas concepto técnico método dos.

Capítulo 7

Conclusiones y Producción

En el marco de desarrollo del presente proyecto se llegó a las siguientes conclusiones, específicamente en relación con los objetivos específicos:

- Se implementó el procedimiento de acondicionamiento de los datos, con el cual se logró reducir el número de consultas de 1 278 364 968 a 28 826 716. Estos 28 826 716 de puntos están dentro de Colombia y se les debe calcular la lista de canales disponibles, sin embargo, los ráster que acompañan los conceptos técnicos tienen un total de 62 944 008 píxeles. Cabe aclarar que este procedimiento no generó cambios en el funcionamiento normal del módulo de cálculo, y la lista de canales disponibles fue calculada utilizando la cartografía de 50m.
- El modelo de datos diseñado se basó en un modelo de base de datos no relacional, debido a la facilidad de la implementación y a la flexibilidad en cuanto a la cantidad de campos. Esto facilitó tareas como la construcción de consultas, la importación y exportación de datos, el ordenamiento de los datos, la interacción y el acceso a los datos.
- El procesamiento fue implementado mediante un esquema multiproceso, el cual sirvió como mecanismo de consulta al módulo de cálculo. Esto permitió reducir el tiempo de obtención de la información para los 29 millones de puntos en aproximadamente siete días tan sólo haciendo modificaciones al flujo general del módulo de cálculo.
- Se construyeron tres ráster, uno por cada concepto técnico. Se realizó el ordenamiento de los puntos en la base de datos, con el objetivo de lograr una correcta georreferenciación en el momento de visualizar cada ráster. Este ordenamiento contempló los datos faltantes en las consultas y fueron rellenados de forma correcta para la visualización. Además, se tuvo en cuenta el sistema de coordenadas para que las proyecciones permitieran una visualización correcta, teniendo en cuenta la concavidad de la tierra en la zona geográfica que abarca la extensión del ráster.
- La generación de los documentos de conceptos técnicos fue automatizada de manera satisfactoria, permitiendo la construcción de cada concepto técnico en una semana. Aunque el equipo de trabajo tuvo que pulir ciertos detalles del documento una vez concatenadas las estadísticas de departamento, municipio y de país, esta concatenación fue el insumo

sólido que permitió obtener los documentos de concepto técnico de más de mil páginas de una forma rápida y sistemática.

El desarrollo de este proyecto, permitió afianzar y ampliar los conocimientos en bases de datos, lenguajes de programación como python, Visual Basic for Application (VBA), multiprocesos. Y desde el punto de vista de ingeniería se demostró la capacidad de adquirir nuevos conocimientos, como fue el manejo de datos georreferenciados, específicamente en estructuras de datos como los son ráster y shapefile.

Como resultado de este trabajo se llevaron a cabo ante la Dirección Nacional de derechos de Autor (DNA) los siguientes registros de software:

Código del registro	Título del registro	Descripción
13-83-499	Construcción de raster para la visualización de HAAT y número de canales disponibles	Se presenta el script en bash que permite la construcción de un raster con una resolución de 200m, basado en la información de una base de datos en mongoDB que contiene los datos de HAAT, y número de canales disponibles, para los puntos de Colombia.
13-84-1	Generación de la Base de datos con lista de canales disponibles TVWS y valor de HAAT para todos los puntos de Colombia	Se presenta el script en python que permita la consulta al módulo de cálculo de la plataforma de espacios en blanco en Colombia fase 2, para obtener los datos de HAAT, Lista de canales disponibles para ambos métodos, radios cocanales y vector adyacente, basados en los puntos de un raster de Colombia con una resolución de 200m.

Continúa en la página siguiente.

Código del registro	Título del registro	Descripción
13-84-2	Generación de tablas por municipio y departamento para la construcción de conceptos técnico para el análisis de disponibilidad de TVWS	Se presenta el script en python que permita la consulta a la base de datos en mongoDB para obtener los archivos CSV que muestran por municipio y departamento la totalidad de canales disponibles en TVWS para la zona urbana y rural de todos los municipios y departamentos de Colombia.
13-83-500	Construcción de un documento en word basado en tablas de excel y archivos por municipios y departamentos	Se presenta una macro en VBA que permite generar y concatenar archivos de Word que responden a una plantilla programada para cada municipio y departamento de Colombia, con los archivos de introducción y conclusiones, generando un documento que presenta el análisis de la totalidad del territorio nacional para número de canales disponibles en ambos métodos y haat utilizando la plataforma de espacios en blanco de Colombia.

Tabla 7.1: Registros de software realizados.

Referencias

- [1] ANE. *RESOLUCIÓN 105 DE 2020*. URL: https://normograma.mintic.gov.co/mintic/docs/resolucion_ane_0105_2020.htm. (accessed: 22.03.2021).
- [2] Baas B. *NoSQL spatial: Neo4j versus PostGIS*. URL: <https://repository.tudelft.nl/islandora/object/uuid:a47d3b8e-650a-4152-a310-366db0773848/?collection=research>. (accessed: 22.03.2021).
- [3] Ann Blyth y col. *A Relational Database Management System*. URL: http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/NoSQL/Home%5C%20Page. (accessed: 22.03.2021).
- [4] Ann Blyth y col. *SPATIAL ANALYSIS AND MODELING*. URL: https://www.geoportal.lt/geoportal/documents/18923/19607/GII-07_training_material.pdf/2352100c-43b7-4ae4-829a-a885625caf34. (accessed: 22.03.2021).
- [5] Mónica Espinosa Buitrago [y otros dieciséis]. *El reino digital. Transformaciones y aplicaciones multidisciplinares*. Colección Iuris et realitas; N.º 2. Universidad Santo Tomás, 2019.
- [6] DNP. *Plan Nacional de Desarrollo 2018-2022 Pacto por Colombia, pacto por la equidad*. URL: <https://colaboracion.dnp.gov.co/CDT/Prensa/Resumen-PND2018-2022-final.pdf>. (accessed: 22.03.2021).
- [7] MongoDB Inc. *MongoDB Documentation*. URL: <https://docs.mongodb.com>. (accessed: 22.03.2021).
- [8] Microsoft. *Office VBA Reference*. URL: <https://docs.microsoft.com/en-us/office/vba/api/overview/>. (accessed: 22.03.2021).
- [9] PORTAFOLIO. «Seis de cada 10 colombianos tienen acceso a internet móvil». En: *Portafolio* (2020), pág. 1.
- [10] Frank Warmerdam, Even Rouault y others. *GDAL documentation*. URL: <https://gdal.org/index.html>. (accessed: 22.03.2021).
- [11] Angela Tatiana Zona-Ortiz y col. *E5 - Documento con la lógica del primer método del modelo de cálculo implementado en la fase I, incluyendo la condición que no permita la utilización de TVWS en las zonas urbanas definidas por la ANE*. Universidad Santo Tomás, 2020.

- [12] Angela Tatiana Zona-Ortiz y col. *E6 - Descripción teórica del segundo método del modelo de Cálculo*. Universidad Santo Tomás, 2020.