

**VISOR GEOGRÁFICO RASTER Y PERFIL PARA LA CORPORACIÓN
AUTÓNOMA REGIONAL DEL VALLE DEL CAUCA**

**Leonardo Espinosa Pachón
Jorge Enrique Hernández Ojeda**



**UNIVERSIDAD DE MANIZALES
FACULTAD DE CIENCIAS E INGENIERÍA
ESPECIALIZACIÓN EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA
MANIZALES
2018**

**VISOR GEOGRÁFICO RASTER Y PERFIL PARA LA CORPORACIÓN
AUTÓNOMA REGIONAL DEL VALLE DEL CAUCA**

**Leonardo Espinosa Pachón
Jorge Enrique Hernández Ojeda**

Trabajo de Grado presentado como opción parcial para optar
al título de Especialista en Información Geográfica

**UNIVERSIDAD DE MANIZALES
FACULTAD DE CIENCIAS E INGENIERÍA
ESPECIALIZACIÓN EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA
MANIZALES
2018**

AGRADECIMIENTOS

Los autores expresan sus agradecimientos a:

Al Grupo SIA de la Corporación Autónoma Regional del Valle del Cauca – CVC por facilitarnos la información espacial.

Al señor Comunicador Social - Periodista Diego Alejandro Guerrero por la revisión del documento

Contenido

LISTA DE FIGURAS.....	7
LISTA DE TABLAS.....	9
LISTA DE ANEXOS.....	9
GLOSARIO.....	10
RESUMEN.....	11
ABSTRACT.....	11
INTRODUCCIÓN.....	12
1. ÁREA PROBLEMÁTICA.....	13
2. OBJETIVOS.....	14
2.1 OBJETIVO GENERAL.....	14
2.2 OBJETIVOS ESPECÍFICOS.....	14
3. JUSTIFICACIÓN.....	15
4. MARCO TEÓRICO.....	16
4.1 MODELOS DE DATOS.....	16
4.1.1 Modelo vectorial.....	16
4.1.2 Modelo ráster.....	17
4.1.3 Ventajas y desventajas de los modelos vectorial y ráster.....	20
4.1.4 Conjunto de datos de mosaico (mosaic dataset).....	21
4.2 MODELOS DIGITALES DE TERRENO Y SUPERFICIE.....	22
4.2.1 Estructura de los modelos digitales.....	22
4.2.2 Modelos digitales de elevación MDE.....	24
4.2.3 Modelos digitales de Superficie MDS.....	24
4.3 PERFILES DE TERRENO.....	25
4.4 SISTEMA DE INFORMACIÓN GEOGRÁFICA - SIG.....	27
4.4.1 Concepto de dato e información.....	28
4.4.2 Componentes de la información geográfica.....	28
4.4.3 Funcionalidades de un sistema de información geográfica.....	29
4.5 PROTOCOLOS Y LENGUAJES.....	30
4.5.1 Python.....	30
4.5.2 XML (Extension Markup Language).....	30

4.5.3	SOAP (Simple Object Access Protocol).....	30
4.5.4	HTTP (Hiper Text Transfer Protocol)	30
4.5.5	REST (REpresentational State Transfer).....	30
4.5.6	JavaScript	31
4.6	SERVICIOS GEOGRAFICOS	31
4.6.1	Clasificación general de servicios SIG	32
4.6.2	Servicio de imágenes dinámico.....	32
4.6.3	Servicios de geoprociamiento	32
4.7	SIG WEB.....	33
4.7.1	SIG corporativo	33
4.7.2	SIG para comunidades	33
4.7.3	SIG para la participación ciudadana.....	33
4.7.4	SIG móviles (Mobile GIS).....	34
4.8	PORTALES GEOGRÁFICOS	37
4.8.1	Funciones de los Geoportales.....	37
4.9	ANTECEDENTES.....	39
4.9.1	Marco normativo del Proyecto	39
4.9.2	Sistema de Información Geo-referenciado Corporativo GeoCVC.....	40
4.9.3	Proyectos LiDAR y radar.....	40
5	METODOLOGÍA	43
5.1	TIPO DE TRABAJO.....	43
5.2	LEVANTAMIENTO DE REQUERIMIENTOS.....	43
5.2.1	Definición del entorno gráfico	43
5.2.2	Alcances de la aplicación geográfica.....	45
5.2.3	Limitaciones de la aplicación geográfica.....	45
5.2.4	Definición detallada de la información	45
5.3	MODELO CONCEPTUAL.....	46
5.4	MODELO LÓGICO	47
5.4.1	Entorno operativo.....	47
5.4.2	Definición de relaciones y procesos.....	47
5.4.3	Identificación de las entidades	47

5.4.4	Clases de entidad (tabla espacial).....	49
5.5	MODELO FISICO	49
5.6	MODELO CARTOGRÁFICO DEL SISTEMA.....	50
5.6.1	Simbología del servicio de MDT.....	50
5.6.2	Simbología de los modelos derivados.....	57
5.6.3	Simbología del servicio de fotografías aéreas	58
5.7	MODELO DE GEOPROCESAMIENTO.....	59
5.7.1	Preparación de los datos.....	59
5.7.2	Modificación del Script	59
5.7.3	Publicación del servicio.....	59
5.8	VISOR GEOGRÁFICO	¡Error! Marcador no definido.
6.	RESULTADOS	61
7.	CONCLUSIONES	69
8.	RECOMENDACIONES.....	70
	BIBLIOGRAFÍA.....	71
	ANEXO A PASOS PARA MODIFICAR SCRIPT PYTHON PERFIL	73
	ANEXO B SCRIPT PERFIL	75

LISTA DE FIGURAS

Fig. 1. Representación de la realidad geográfica a través de dos modelos.	16
Fig. 2. Modelo de datos vector	17
Fig. 3. Modelo de datos ráster	17
Fig. 4. Estructura del modelo ráster.	18
Fig. 5. Información con valores cero en las celdas	19
Fig. 6. Rásters con diferentes resoluciones espaciales	19
Fig. 7. Comparación de los modelos ráster y vector	20
Fig. 8. Esquema de un conjunto de datos de mosaico	21
Fig. 9. Modelos TIN tomado de (Environmental Systems Research Institute, 2016)	22
Fig. 10. Modelo de contornos generado a partir de ráster tomado de (Environmental Systems Research Institute, 2016)	23
Fig. 11. Modelo ráster de matrices, tomado de (ZENIT, 2018)	24
Fig. 12. Representación MDE tomado de (Inegi, 2018)	24
Fig. 13. Representación MDE y MDS tomado de (Geoimage, 2018)	25
Fig. 14. Perfil sobre un MDT tomado de (Carlos Gómez, 2008)	25
Fig. 15. Perfil sobre el río Cauca Cortesía Ingetop Ltda.	26
Fig. 16. Componentes de un SIG, tomado de (Instituto Geográfico Nacional de Argentina, s.f.)	28
Fig. 17. Relaciones espaciales y atributos, tomado del Instituto Geográfico Nacional de Argentina (Instituto Geográfico Nacional de Argentina, s.f.)	29
Fig. 18. Proceso de consulta WEB	31
Fig. 19. Portal ciudadano Idas Cali, tomado de (Geovisor IDESC, 2018)	34
Fig. 20. Aplicación GisData	35
Fig. 21. Plataforma GisData	36
Fig. 22. Plataforma CARTO DB	36
Fig. 23. Geoportal GEO	37
Fig. 24. Modelo de sombras generado a partir de LiDAR del corredor Río Cauca, Zona Juanchito	41
Fig. 25. Orto imagen en banda X	42
Fig. 26. Extensión geográfica de los Datos	44
Fig. 27. Modelo conceptual	46
Fig. 28. Inicio del proceso de creación de plantilla función	51
Fig. 29. Editor de plantillas de funciones, función identidad	51
Fig. 30. Parámetros de la función extender (stretch)	52
Fig. 31. Parámetros de la función colormap	53
Fig. 32. Funciones encadenadas y exportación de plantilla de funciones	53
Fig. 33. Inicio del proceso de publicación del servicio de imágenes	54
Fig. 34. Asistente de publicación del servicio de imágenes	54
Fig. 35. Implementación de plantilla en servicio	55
Fig. 36. Servicio MDT corredor Río Cauca, simbología dinámica en base a rango de elevaciones	56
Fig. 37. Servicio de sombras del corredor del Río Cauca	57
Fig. 38. Servicio de fotografías aéreas del corredor del Río Cauca	58
Fig. 39. Inicio del proceso de publicación servicio geoprocetamiento	60
Fig. 40. Parámetros que requiere el servicio tomado de Esri	60
Fig. 41. Requerimientos obtenidos	61
Fig. 42. Estructura de carpetas en el servidor	62
Fig. 43. Estructura de los datos que emplea el servicio	62
Fig. 44. Carpeta con los principales servicios de visor	63
Fig. 45. Servicios de modelos derivados a partir de una fuente MDT única	64
Fig. 46. Mosaic dataset de aerofotografías y Oris	64

<i>Fig. 47. Servicios de imágenes de aerofotografías y Oris</i>	65
<i>Fig. 48. Servicios de imágenes de aerofotografías y Oris</i>	65
<i>Fig. 49. Elementos del visor perfil</i>	66
<i>Fig. 50. Barra de Herramientas</i>	67
<i>Fig. 51. Herramienta de perfil</i>	67
<i>Fig. 52. Digitalización polilínea y mano alzada</i>	67
<i>Fig. 53. Botones de exportación</i>	68
<i>Fig. 54. Perfil exportado en formato PNG</i>	68
<i>Fig. 55. Datos del perfil exportados en formato XLS</i>	68

LISTA DE TABLAS

<i>Tabla 1 Ventajas y desventajas más relevantes de las estructuras de datos</i>	20
<i>Tabla 2 Requerimientos sintetizados y su tipo</i>	43
<i>Tabla 3 Datos suministrados para la construcción del visor</i>	48
<i>Tabla 4 Esquema de la clase de entidad Demboundary</i>	49
<i>Tabla 5 Esquema de la clase de entidad Profileschema</i>	49

LISTA DE ANEXOS

Anexo A Script perfil

Anexo B Pasos para modificar script Python perfil

GLOSARIO

Add-in: es una personalización, un conjunto de herramientas que se integra en una aplicación con el fin de proporcionar funciones complementarias para la realización de tareas personalizadas.

CVC: (Corporación Autónoma Regional del Valle del Cauca) la entidad se define en su misión corporativa (Corporación Autónoma Regional del Valle del Cauca, 2018) “Somos la entidad encargada de administrar los recursos naturales renovables y el medio ambiente del Valle del Cauca, que como máxima autoridad ambiental y en alianza con actores sociales propende por un ambiente sano, contribuyendo al mejoramiento de la calidad de vida de la población y la competitividad de la región en el marco del desarrollo sostenible.”

Esri: (Environmental Systems Research Institute) empresa estadounidense creada en 1969, desarrolla y comercializa el software ArcGIS.

Formato ECW: (Enhanced Compression Wavelet) es un formato de archivo propietario para imágenes ráster, desarrollado por la empresa Earth Resource Mapping y presenta unas escalas muy altas de compresión que van desde 10:1 a 50:1.

GDB: (GeoDataBase) formato de almacenamiento espacial y alfanumérico creado por esri

LiDAR: (Light, Detection And Ranging) Esri (Environmental Systems Research Institute, 2016) lo define como “una técnica de teledetección óptica que utiliza luz láser para obtener una muestra densa de la superficie de la tierra, obteniendo nubes de puntos masivos que se pueden administrar, visualizar, analizar y compartir; se utiliza principalmente en aplicaciones de representación cartográfica”.

Píxel: Acrónimo de Picture element. Es la mínima unidad de un ráster.

Radar: (Radio Detection And Ranging) tecnología que emplea ondas electromagnéticas para medir distancias, altitudes, direcciones y velocidades de objetos estáticos o móviles, usualmente se utiliza como un sensor activo para generar imágenes de la superficie terrestre a través de las nubes.

SRTM: (Shuttle Radar Topography Mission) Proyecto internacional cuyo fin fue obtener un modelo digital de elevación de la superficie terrestre.

Teselación: Es el cubrimiento de una superficie plana por medio de una figura geometría (regularmente un cuadrado o rectángulo) donde las figuras no deben superponerse o dejar huecos entre ellas.

RESUMEN

La Corporación Autónoma Regional del Valle del Cauca - CVC contrató la obtención de datos a partir de radar y LiDAR los cuales proporcionan información de gran detalle del departamento del Valle del Cauca. Los raster obtenidos suelen ser de gran tamaño y es necesario software especializado para visualizarlos y analizarlos, lo cual dificulta el acceso a dichos datos. El desarrollo de un visor geográfico permite la centralización de los datos y al mismo tiempo es un medio para divulgar esta información al público interesado. A continuación, se presenta como se llevó a cabo el proyecto, desde la estructuración de los datos pasando por los servicios de visualización dinámicos, así como los servicios necesarios para el funcionamiento de su herramienta principal de perfil topográfico, hasta llegar a la construcción del visor web.

PALABRAS CLAVES: Ráster, SIG Web, Visor geográfico, Perfil

ABSTRACT

The Regional Autonomous Corporation of the Valle del Cauca contract the obtaining of data from radar and LiDAR which provide information of great detail of the department of Valle del Cauca, the raster obtained are usually of great size and specialized software is necessary to visualize and analyze them, this hinders access to data; the development of a geographic viewer allows centralization of the data and at the same time it is a medium to disseminate this data to the interested public, the following is presented as the project was carried out, from the structuring of the data through the services of dynamic visualization as well as the necessary services for the operation of the main topographic profile tool, until reach the construction of the web viewer

KEY WORDS: Raster, GIS Web, Geographic Viewer, Profile

INTRODUCCIÓN

Los sistemas de información geográfica son un campo de estudio muy apasionante y extenso. Cada día nos encontramos con nuevos conceptos, páginas con información enriquecida, mapas que interactúan con nosotros de forma cada vez más amigable.

En años anteriores, el tratamiento dado a los datos por las instituciones del Estado y privadas era salvaguardar sus datos, sus mapas y sus bases de datos, pues eran procesos de captura muy costosos que no podían ser compartidos. La masificación del internet, de los equipos y de la tecnología como tal, y el cambio en la legislación han permitido que muchos datos, que eran restringidos o de difícil acceso, hoy estén disponibles para su consulta y uso público, lo que representa grandes beneficios para el desarrollo de los proyectos y la interoperabilidad entre instituciones y resalta la importancia de instituciones generadoras de datos oficiales en temáticas específicas.

La Ley 1712 sobre transparencia y derecho al acceso de la información pública que promulgó el gobierno colombiano en el año 2014 sentó las bases para que instituciones o personas externas a las entidades gubernamentales pudieran consultar los datos de dominio público, un medio para lograr esto, en relación a datos geográficos, es mediante el desarrollo de portales web geográficos, los cuales son cada vez más especializados e interactivos.

Este documento pretende dar una visión general del entorno de los Sistemas de información geográfica en la web y adentrarse en la potencialidad de los servicios web de imágenes para los usuarios internos y externos de la Corporación Autónoma Regional del Valle del Cauca – CVC.

1. ÁREA PROBLEMÁTICA

La Corporación Autónoma Regional del Valle del Cauca – CVC es una entidad pública, encargada de la administración de los recursos naturales y su protección en el departamento del Valle del Cauca; muchas de las actividades llevadas a cabo en este ámbito requieren datos geoespaciales: cartografía y datos vectoriales en distintos formatos, o los comúnmente denominados datos ráster, el cual es un arreglo matricial de celdas utilizado para el almacenamiento de fotografías, imágenes de satélites, radar o modelos de elevación, pendiente, temperatura, precipitación, entre otros.

En años recientes, la CVC ha encargado la adquisición de un modelo de elevación de 2,5m a partir de radar para todo el Valle del Cauca y la parte norte del Cauca y otros modelos de elevación de 1m a partir de tecnología LiDAR y fotografías aéreas de alta resolución de algunas cabeceras municipales y del corredor del Río Cauca, desde la represa de Salvajina hasta el municipio de Cartago en el norte del departamento; para la visualización, análisis y obtención de los modelos derivados (como los de pendiente o aspecto) es necesario software especializado, adicionalmente es necesario conocer cómo emplearlo y esto representa una barrera al acceso a estos datos.

Se requiere dar conocer estos datos para que sean utilizados por los funcionarios, dado que están empleando el modelo de elevación SRTM, que tiene una resolución de 30m y varias inconsistencias, de igual forma se hace necesario la difusión al público en general que requiera dicha información.

Se necesita una aplicación que permita generar perfiles de terreno a partir de los modelos digitales de elevación obtenidos; también se necesita que sea posible el despliegue de las fotografías y que adicionalmente se cuente con los modelos derivados como el de pendiente y aspecto.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar un visor geográfico que permita consultar la información ráster LiDAR y radar que posee la Corporación Autónoma Regional del Valle del Cauca – CVC.

2.2 OBJETIVOS ESPECÍFICOS

- Consolidar y estructurar la información ráster resultado de los proyectos de radar y LiDAR que posee la CVC con el objeto de dar respuesta a las consultas a dicha información.
- Crear servicios geográficos de imágenes y geoprocesamiento eficientes para el despliegue de los raster y permitan la creación de perfiles topográficos.
- Construir una aplicación web que permita la difusión de la información a los funcionarios y al público en general.

3. JUSTIFICACIÓN

Con el desarrollo de este proyecto se busca dar cumplimiento a los principios de facilitación, transparencia, celeridad, eficacia y de divulgación proactiva de la información, consagrado en la Ley 1712 de 6 marzo de 2014 (Congreso de la Republica de Colombia, 2014) “derecho al acceso a la información pública nacional”, la cual determina que se debe garantizar el acceso a toda información capturada o producida por entidades públicas a excepción de información clasificada o reservada.

Además de dar cumplimiento a la ley anteriormente descrita, la difusión de los datos ráster obtenidos de los proyectos de radar y LiDAR, ayudarán a los funcionarios en el cumplimiento de las actividades propias de su cargo o en el desarrollo de proyectos de diversa índole, facilitando la búsqueda de estos datos, haciendo un uso adecuado de los recursos y evitando la compra de información duplicada o innecesaria y proporcionando un mayor grado de detalle, sobre todo en aquellas actividades que requieren información de alturas sobre el terreno.

La organización de forma centralizada en un servidor, tanto de fotografías aéreas como de los modelos digitales de elevación, ahorrará espacio de almacenamiento en las estaciones de trabajo de los funcionarios, adicionalmente el visor web geográfico servirá de medio de difusión para el público en general y se implementará aprovechando la plataforma tecnológica existente, servidores y software de la plataforma ArcGIS, así como el portal geográfico de la CVC llamado GeoCVC lo cual ahorrará costos en la implementación.

4. MARCO TEÓRICO

La creación del visor ráster y su integración al portal GeoCVC requiere entender conceptos de los sistemas de información y de los portales geográficos, así como un conocimiento del dato ráster, fundamentado en la apreciación de la realidad y abstracción de esta, también se deben comprender las herramientas analíticas aplicadas; estos fundamentos serán explicados a continuación.

4.1 MODELOS DE DATOS

Los modelos de datos son la representación del mundo real en dos tipos de formatos de igual importancia: modelo vector y modelo ráster.

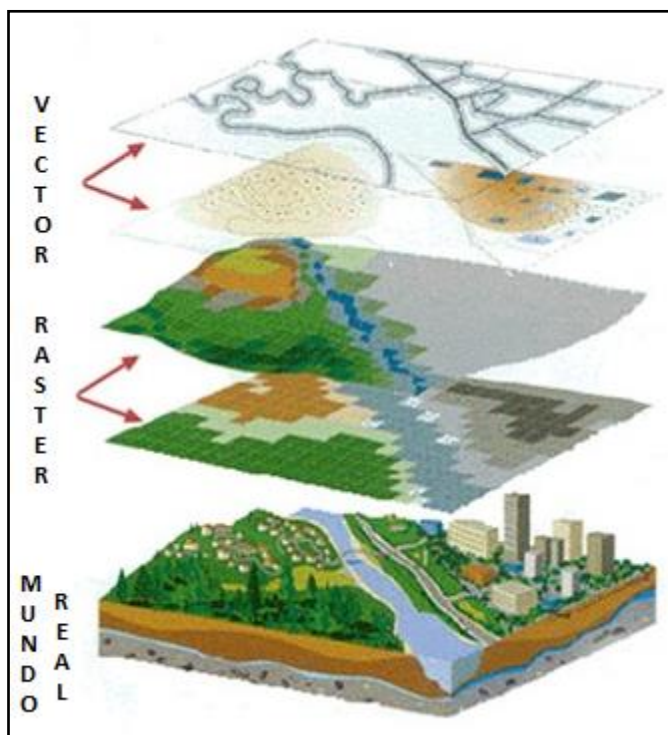


Fig. 1. Representación de la realidad geográfica a través de dos modelos.

4.1.1 Modelo vectorial

Este modelo se rige por las entidades geométricas que lo conforman, el modelo vector tiene tres tipos de geometría que son puntos, líneas y polígonos. Los puntos son representados por un par de coordenadas x, y . Las líneas son representadas por un par de coordenadas de inicio y un par de coordenadas final $x_1, y_1; x_2, y_2$. Los polígonos se representan por un conjunto de coordenadas $x_1, y_1; x_i, y_i; x_n, y_n$. Con inicio y fin común. Los detalles se interrelacionan geoméricamente y están representados en su topología (vecindad, inclusión, contención,

continuidad, entre otros). Los atributos temáticos están representados en tablas alfanuméricas, conteniendo mecanismos para ser entrelazados con sus respectivas entidades cartográficas.


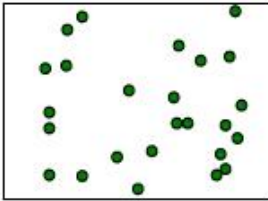

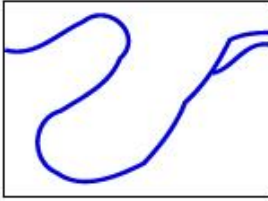

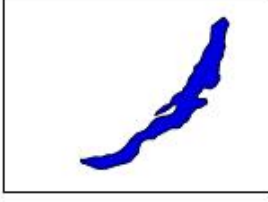
Primitiva	Entidad espacial	Representación	Atributos																		
Puntos			<table border="1"> <thead> <tr> <th>ID</th> <th>Alura</th> <th>Diámetro Normal</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>17,5</td> <td>35</td> </tr> <tr> <td>2</td> <td>22</td> <td>45,6</td> </tr> <tr> <td>3</td> <td>15</td> <td>27,2</td> </tr> <tr> <td>4</td> <td>19,7</td> <td>36,1</td> </tr> <tr> <td>...</td> <td></td> <td></td> </tr> </tbody> </table>	ID	Alura	Diámetro Normal	1	17,5	35	2	22	45,6	3	15	27,2	4	19,7	36,1	...		
ID	Alura	Diámetro Normal																			
1	17,5	35																			
2	22	45,6																			
3	15	27,2																			
4	19,7	36,1																			
...																					
Líneas			<table border="1"> <thead> <tr> <th>Ancho máx(m)</th> <th>Calado máx(m)</th> <th>Longitud(km)</th> </tr> </thead> <tbody> <tr> <td>15</td> <td>4,3</td> <td>35</td> </tr> <tr> <td>6,3</td> <td>3,9</td> <td>6,2</td> </tr> </tbody> </table>	Ancho máx(m)	Calado máx(m)	Longitud(km)	15	4,3	35	6,3	3,9	6,2									
Ancho máx(m)	Calado máx(m)	Longitud(km)																			
15	4,3	35																			
6,3	3,9	6,2																			
Polígonos			<table border="1"> <thead> <tr> <th>Superficie(km)²</th> <th>Profundidad máx(m)</th> </tr> </thead> <tbody> <tr> <td>31494</td> <td>1637</td> </tr> </tbody> </table>	Superficie(km) ²	Profundidad máx(m)	31494	1637														
Superficie(km) ²	Profundidad máx(m)																				
31494	1637																				

Fig. 2. Modelo de datos vector

4.1.2 Modelo ráster

Es un arreglo de filas y columnas del mismo tamaño. La intersección de fila-columna es llamada celda y para cada una de las celdas existe información que la describe.

Referencia:	1	1	3	3	3	3	4
1. Unidad Espacial A	1	1	1	3	3	4	4
2. Unidad Espacial B	1	1	2	3	3	4	4
3. Unidad Espacial C	2	2	2	2	4	4	4
4. Unidad Espacial D	2	2	2	2	4	4	4

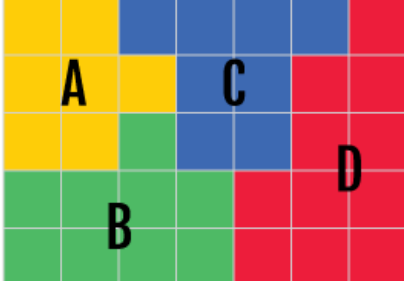


Fig. 3. Modelo de datos ráster

En este modelo, no se conocen las coordenadas de cada celda, tan solo los valores de ellas. El tamaño de las celdas da la resolución del ráster, la información de cada celda es contigua, cubren un área determinada y no se superponen entre sí, formando una malla regular de igual tamaño de celda y de forma cuadrada.

(Olaya, 2014) construyó el siguiente esquema que ilustra cómo se manejan las coordenadas en base a la localización de la celda (x,y) y su tamaño (d).

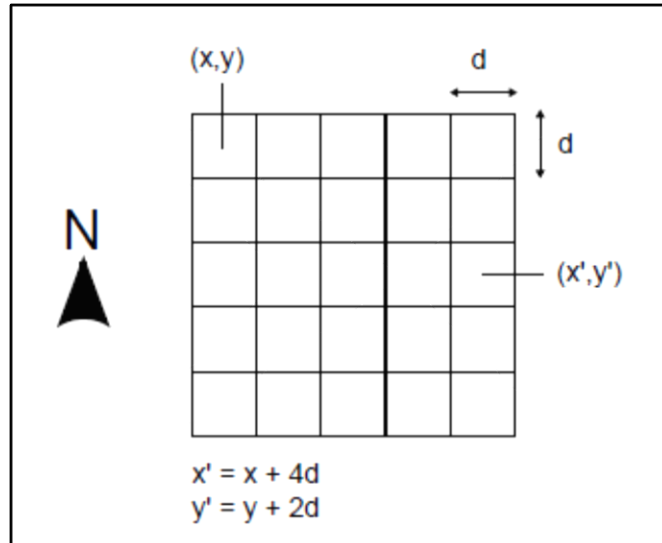


Fig. 4. Estructura del modelo ráster.

Para el modelo ráster, la topología viene implícita en el propio modelo y no depende de los valores de las celdas. Solo depende de la estructura de la malla de datos en sí.

Bueno ¿Y qué sucede con las imágenes y fotografías aéreas que se toman en sentido Sur-Este o Sur-Oeste? La información se presenta como un arreglo de celdas de igual tamaño, pero con celdas con valores de información igual a cero.

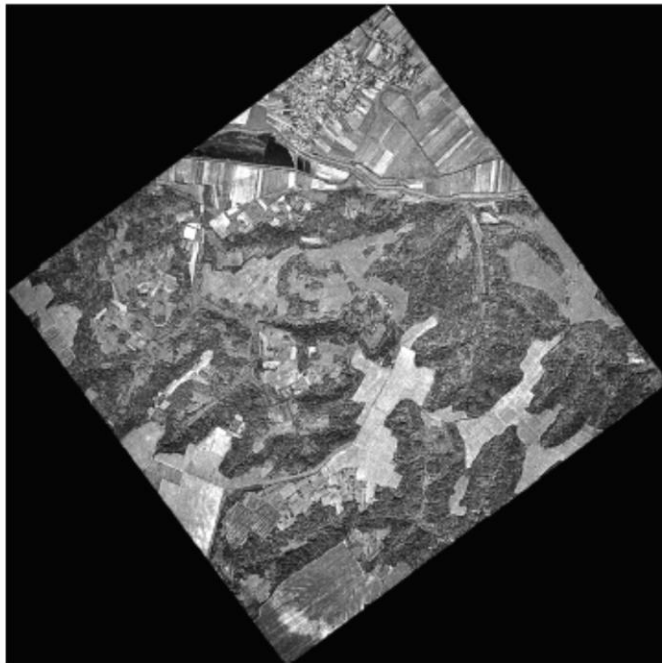


Fig. 5. Información con valores cero en las celdas

Otro parámetro importante en el ráster es el tamaño de la celda o tamaño de pixel y esto se denomina resolución espacial. Este valor define la escala o calidad de la imagen y oscila desde valores de centímetros hasta cientos de metros. Un valor pequeño de pixel implica una mayor resolución; un valor grande de pixel, una menor resolución.

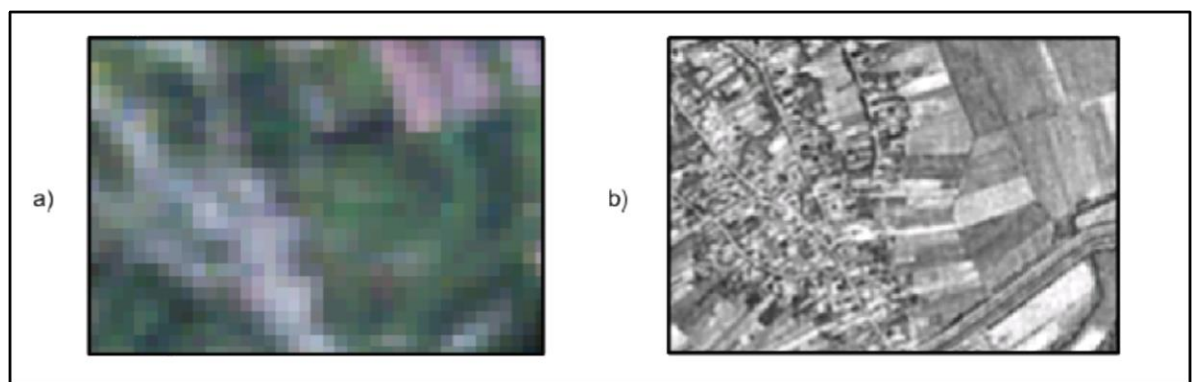


Fig. 6. Rásters con diferentes resoluciones espaciales

4.1.3 Ventajas y desventajas de los modelos vectorial y ráster

Cada una de estas estructuras tienen ventajas y desventajas que deben ser consideradas al momento de ser utilizadas para la representación de algún elemento, en la Tabla 1 se presentan las más relevantes.

Tabla 1 Ventajas y desventajas más relevantes de las estructuras de datos

	Ventajas	Desventajas
V E C T O R	Menor requerimiento en almacenamiento y memoria.	Estructura de datos más compleja.
	Codificación eficiente de la topología.	Las operaciones de superposición son más difíciles de implementar.
	Mejor compatibilidad con entornos de base de datos.	Formato más laborioso de mantener actualizado.
R A S T E R	Estructura de los datos muy simple.	Mayor requerimiento en almacenamiento y memoria.
	Las operaciones de superposición son muy sencillas.	Dependiendo de la resolución espacial, los elementos pueden tener sus límites no muy bien definidos.
	Formato óptimo para variaciones altas de datos.	

Una comparación de forma con el modelo de representación ráster y el modelo vector es la división del espacio en unidades cuadradas que impide la representación fiel de entidades tales como curvas como la mostrada en trazo rojo en la figura 9.

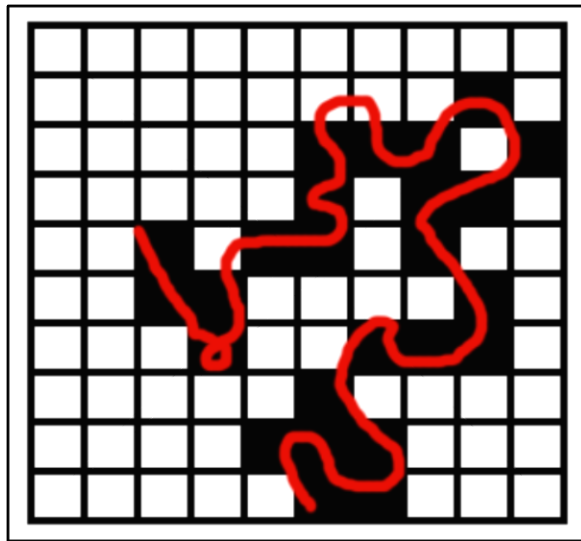


Fig. 7. Comparación de los modelos ráster y vector

4.1.4 Conjunto de datos de mosaico (mosaic dataset)

Es una estructura de datos creada por Esri para el manejo de información ráster indistinta, dentro de una GDB, tiene ventajas frente a ráster individuales, tales como la configuración de parámetros como el orden de dibujo de los ráster participantes, personalización de huellas entre otras y en la visualización se despliega como si fuera un elemento único. Los componentes del conjunto de datos de mosaico, la realiza Esri (Environmental Systems Research Institute, 2016) y es la siguiente

“Un conjunto de datos de mosaico consta de:

- Un catálogo que proporciona la fuente de los píxeles y las huellas de los raster
- Una clase de entidad que define el límite
- Un conjunto de reglas de creación de mosaicos que se utilizan para crear dinámicamente el mosaico de ráster
- Un conjunto de propiedades que se utiliza para controlar la creación de mosaicos y la extracción de imágenes
- Una tabla para el registro durante la carga de datos y otras operaciones
- Opcionalmente, una clase de entidad de línea de unión para la creación de mosaicos de línea de unión
- Opcionalmente, una tabla de corrección de colores que defina la asignación de colores para cada ráster del catálogo de ráster”

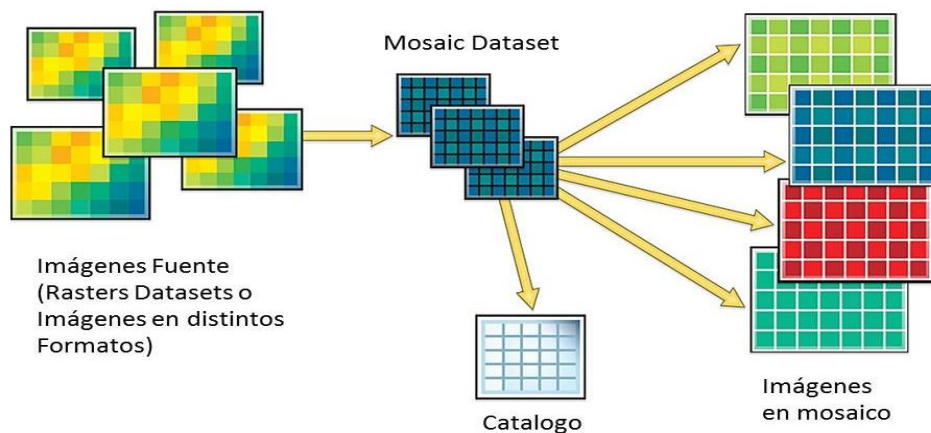


Fig. 8. Esquema de un conjunto de datos de mosaico

4.2 MODELOS DIGITALES DE TERRENO Y SUPERFICIE

Los modelos digitales se han definido como “un conjunto de datos numéricos que describe la distribución espacial de una característica del territorio” (Doyle, 1978).

Esta definición describe en parte al modelo, pero no describe la relación de los datos con la variable que se va a modelar. Es decir, podemos hacer un modelo digital de elevación, de temperatura, de pendientes, etc. Y todos cumplen la premisa de distribución espacial sobre un territorio. Existe una definición más elaborada que involucra el tipo de dato que debe ser cuantitativo y continuo y define un modelo como “Una estructura numérica de datos que representa la distribución espacial de una variable cuantitativa y continua” (Felicísimo, 2004).

Existen dos diferenciaciones sobre los modelos de terreno, estos pueden ser análogos (maquetas a escala) o digitales (archivos de datos). Nos centraremos en los modelos digitales.

El término de modelo digital de terreno nació en el laboratorio de fotogrametría de la Universidad de Massachussetts en el año de 1958 cuando se trataba de dar nombre a una representación estadística de un gran número de puntos con coordenadas x,y,z distribuidos sobre una superficie continua del terreno.

4.2.1 Estructura de los modelos digitales

Según su estructura los Modelos digitales se dividen en Vectoriales y Ráster, siendo los primeros basados en líneas y puntos, y el segundo en un arreglo matricial que teselan por completo la superficie del terreno en estudio.

Bajo esta estructura se encuentran los modelos TIN, contornos y los modelos ráster de matrices regulares.

- Modelo TIN: es una red de triángulos irregulares creados a partir de 3 puntos cercanos no alineados y se pegan al terreno formando un mosaico para adaptarse a la superficie.

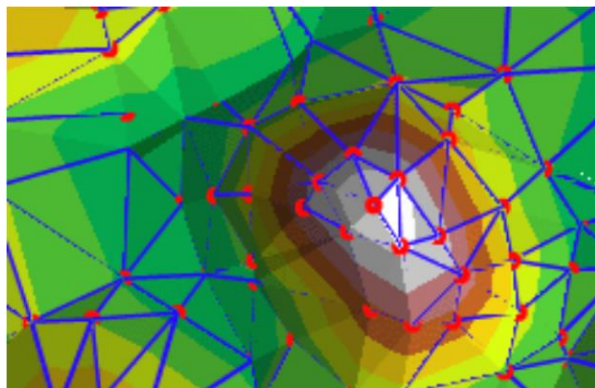


Fig. 9. Modelos TIN tomado de (Environmental Systems Research Institute, 2016)

- **Modelo de Contornos:** son vectores alineados que poseen la misma elevación en su trayectoria, generando el mapa de curvas de nivel. El valor de elevación o altura se denomina cota y puede tomar diferentes valores dependiendo del terreno. Las curvas de nivel espaciadas entre sí determinan una zona plana, las curvas cercanas entre sí pueden determinar una elevación o una depresión.

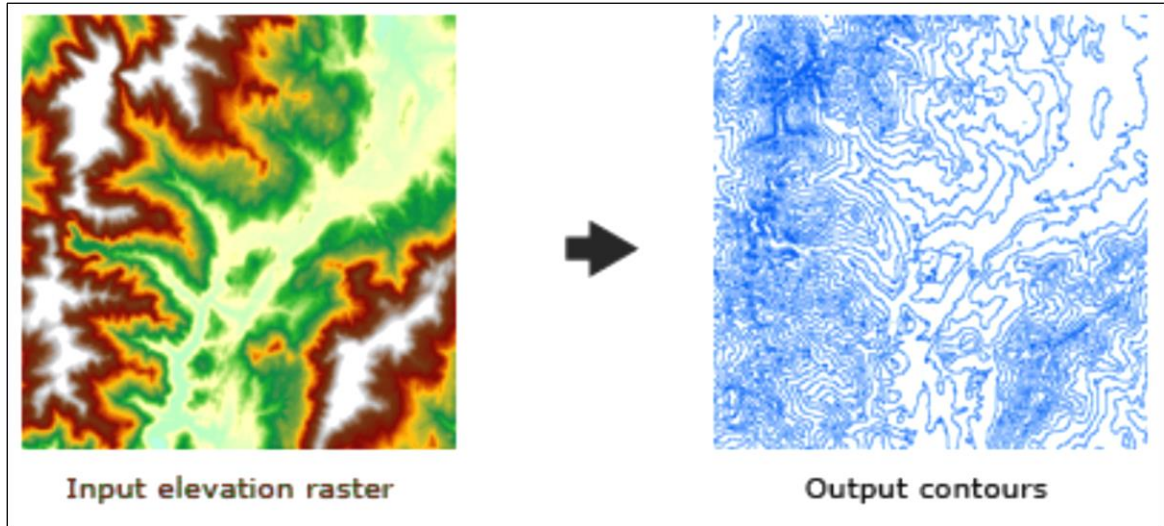


Fig. 10. Modelo de contornos generado a partir de ráster tomado de (Environmental Systems Research Institute, 2016)

- **Modelos ráster de matrices:** es el resultado de superponer una cuadrícula o malla regular sobre el terreno y extraer la altura media de cada celda.

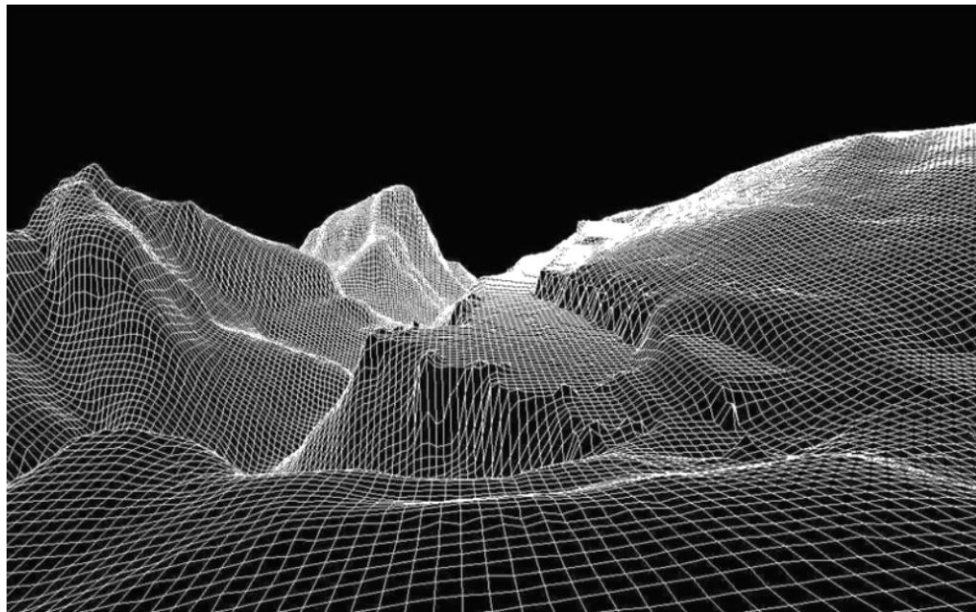


Fig. 11. Modelo ráster de matrices, tomado de (ZENIT, 2018)

4.2.2 Modelos digitales de elevación MDE

Continuando con la definición de modelo digital, un modelo de elevación es una distribución numérica de datos que representan la distribución espacial del terreno en alturas o elevaciones llamadas también cotas.

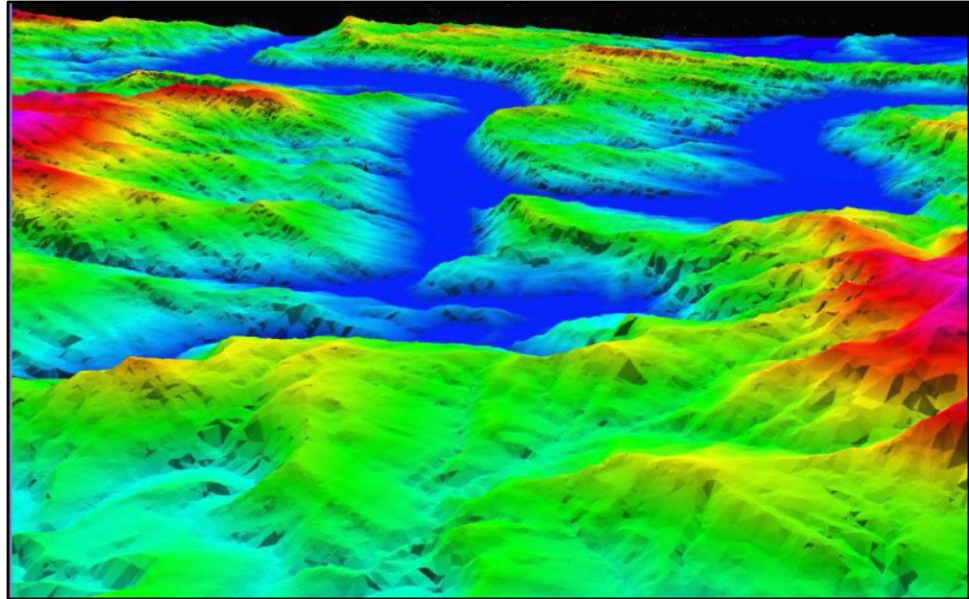


Fig. 12. Representación MDE tomado de (Inegi, 2018)

La elección de la estructura del modelo depende de la cantidad de datos fuente, el tipo de almacenamiento y la representación o salida gráfica al final del análisis. Todas las estructuras tienen sus ventajas y desventajas y ellas se dejan de diferenciar si la precisión y el volumen de los datos son parecidos entre sí.

4.2.3 Modelos digitales de Superficie MDS

El modelo digital de superficie hace referencia a describir todos los objetos sobre la superficie terrestre incluyendo construcciones, copas de árboles, puentes, vías, infraestructura y demás elementos ubicados encima de la tierra.

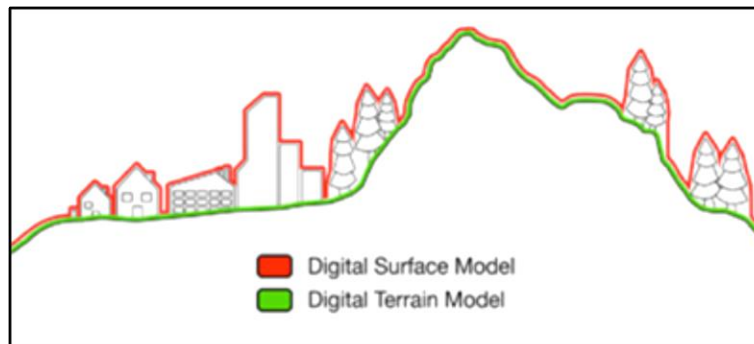


Fig. 13. Representación MDE y MDS tomado de (Geoimage, 2018)

Las tecnologías que pueden generar estos modelos son la fotogrametría, el LIDAR y la tecnología SAR (Radar).

4.3 PERFILES DE TERRENO

Se define un perfil como (Mantelo, 1979) "la operación producto de la nivelación de puntos entre sí a lo largo de una alineación determinada".

La información de alturas se registra sobre el eje y, el abscisado o desplazamiento horizontal se registra sobre el eje x. Existen 2 tipos de perfiles topográficos: los perfiles sobre el terreno y los perfiles o secciones sobre un río (sección batimétrica).

Los perfiles son muy utilizados en el campo de vías, geología, batimetría y donde se requiera conocer la diferencia de niveles entre 2 o más puntos.

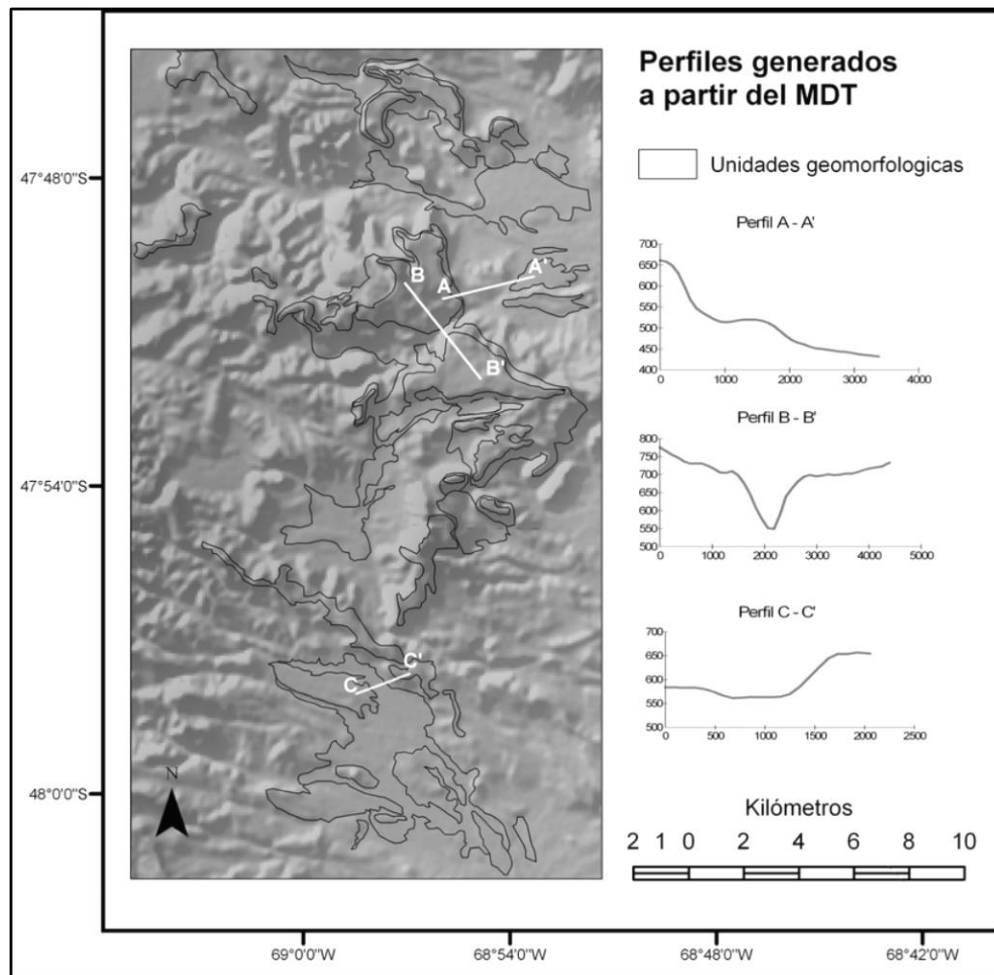


Fig. 14. Perfil sobre un MDT tomado de (Carlos Gómez, 2008)

Para realizar un perfil de terreno se debe tener un mapa de curvas de nivel o un modelo de elevación. Con las curvas de nivel podemos generar un trazado donde se intersecan las curvas con el alineamiento y determinar pendientes y calcular volúmenes de corte y de relleno.

Como se mencionó, existe otro tipo de perfiles que se realizan sobre los ríos y cuerpos de agua, el procedimiento se llama batimetría y en él se destacan 2 categorías. La sección transversal al río y el perfil longitudinal o Talweg que se define como los valores mínimos de cada sección, unidos en forma longitudinal al río.

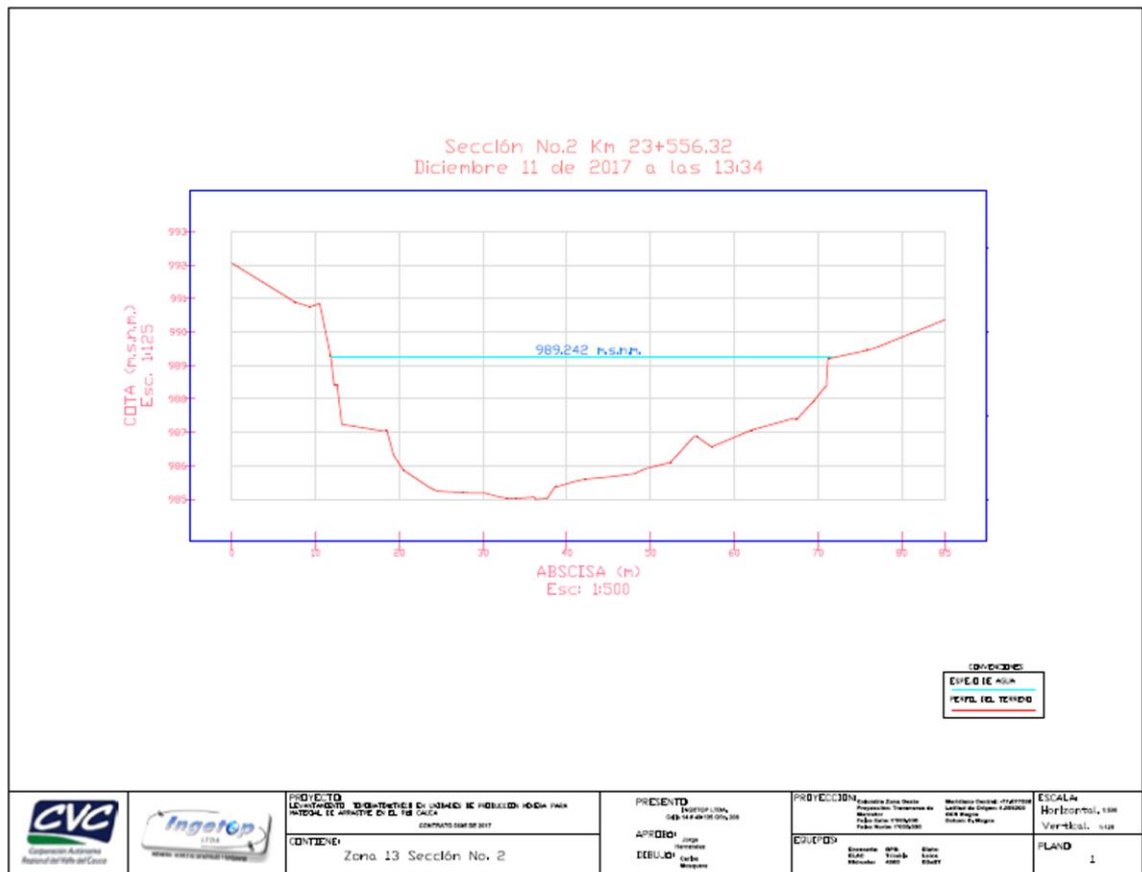


Fig. 15. Perfil sobre el río Cauca Cortesía Ingetop Ltda.

4.4 SISTEMA DE INFORMACIÓN GEOGRÁFICA - SIG

Existen muchas definiciones sobre SIG, unas más profundas que otras. Podemos decir que un Sistema de Información Geográfica es un sistema complejo, global e independiente que permite modelar información georreferenciada espacialmente para la toma de decisiones que afectan directamente su entorno.

Una definición más clásica nos la presenta Tomlin (Tomlin, 1990) para quien un SIG es:

“Un elemento que permite analizar, presentar e interpretar hechos relativos a la superficie terrestre” complementando más adelante como “un conjunto de software y hardware diseñado específicamente para la adquisición, mantenimiento y uso de datos cartográficos.

Partiendo de esta descripción, podemos dar una definición más precisa de lo que un SIG puede ser.

Básicamente un SIG ha de permitir la realización de las siguientes operaciones:

- Lectura, edición, almacenamiento y, en términos generales, gestión de datos espaciales.
- Análisis de dichos datos. Esto puede incluir desde consultas sencillas a la elaboración de complejos modelos, y puede llevarse a cabo tanto sobre la componente espacial de los datos como sobre la componente temática (el valor del elemento en sí).

En la actualidad, los avances tecnológicos permiten usar diferentes herramientas para el manejo o administración de un SIG, convirtiéndose este en un sistema integrador de tecnología, personas e información geográfica.”

Un SIG como todo sistema, está formado por una serie de componentes o subsistemas encargadas de diferentes funcionalidades, las cuales son:

- **Datos:** es la información tanto espacial como alfanumérica.
- **Métodos:** son los procesos, algoritmos, modelos que se aplican a la información.
- **Software:** son los programas y aplicaciones usadas para presentar los datos.
- **Hardware:** es el equipo y periféricos necesarios para la ejecución de los programas.
- **Personas:** son las encargadas de interactuar con los componentes del SIG.



Fig. 16. Componentes de un SIG, tomado de (Instituto Geográfico Nacional de Argentina, s.f.)

4.4.1 Concepto de dato e información

La definición más sencilla de dato es un valor usado para representar algo. Y la información es el resultado de la interpretación de ese dato. Es decir, el dato tiene valor solo si su significado o valor tiene importancia o representa un significado propio para el usuario. Un ejemplo claro es un dato de altura, este se vuelve información cuando conocemos que el valor del dato es la altura.

4.4.2 Componentes de la información geográfica

Existen dos componentes que representan la información geográfica:

- **Componente espacial** hace alusión a la localización geográfica dentro de un sistema de coordenadas definido.
- **Componente alfanumérico** es la información almacenada dentro de la componente espacial, también llamada atributo.

El vínculo entre estos dos componentes es de carácter relacional, es decir, uno le da sentido al otro. Los tipos de geometría que componen la información espacial son puntos, líneas, polígonos y superficies.

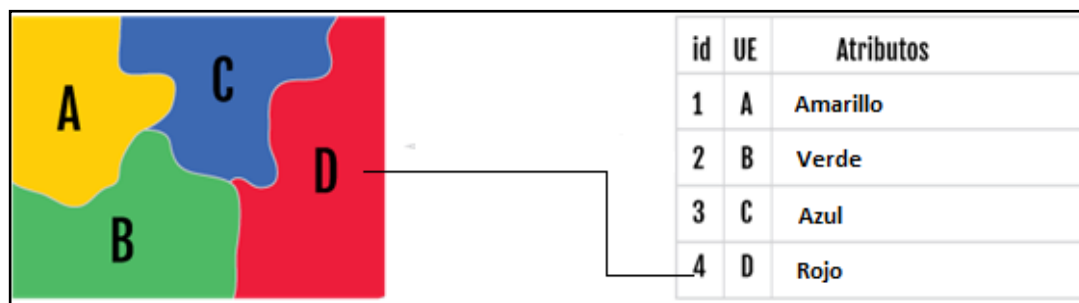


Fig. 17. Relaciones espaciales y atributos, tomado del Instituto Geográfico Nacional de Argentina (Instituto Geográfico Nacional de Argentina, s.f.)

4.4.3 Funcionalidades de un sistema de información geográfica

Un SIG debe cumplir con funciones básicas para facilitar el trabajo de los usuarios. Estas funciones son:

- **Entrada y salida de datos:** es la capacidad de “leer” datos espaciales y alfanuméricos de manera conjunta y/o separada. Debe tener las herramientas para importar/exportar datos desde y hacia otros formatos a través de comandos o conversiones. Más adelante retomaremos el tema de esta funcionalidad en el capítulo de portales geográficos en la web.
- **Visualización:** Un SIG posee las herramientas necesarias para representar la información geográfica, explorar los datos, herramientas de visualización de información a diferentes escalas y la representación gráfica y de colorimetría necesarios para la correcta visualización de la información espacial.
- **Análisis:** se basa en la entrada de datos, un modelo o proceso que se ejecuta y una salida o resultado producto del proceso. Generalmente esta funcionalidad puede estar ubicada en módulos, herramientas o modelos. Por ejemplo, el geo-proceso llamado área de influencia (buffer) permitirá crear una nueva capa con base en una distancia al eje de una línea. En la actualidad se pueden usar funcionalidades de análisis en la web donde el cliente solicita un proceso y el servidor web está en capacidad de generar el procesamiento y entregar el resultado en muy corto tiempo.
- **Edición:** La información espacial y alfanumérica sufre cambios en los procesos de análisis y de entrada de datos. Una funcionalidad importante es la de poder editar la información para actualizarla, eliminarla o modificarla. La edición puede ser en la tabla de atributos, en la parte espacial o en los valores de un ráster.

4.5 PROTOCOLOS Y LENGUAJES

Varios factores han influido en el avance tecnológico de los SIG y su implementación en la web, por nombrar los más notables están: la reducción en el costo de los servidores y su gran incremento en el desempeño, la implementación de la computación en la nube, el aumento de la velocidad en la transmisión y recepción de los datos, el desarrollo de lenguajes y protocolos más eficientes, estos últimos son empleados por los servicios geográficos que usa el visor. A continuación, presentamos los conceptos de estos lenguajes y protocolos.

4.5.1 Python

Lenguaje creado por Guido van Rossum. La definición es realizada por Python Software Foundation (Python Software Foundation, 2018) “Es un lenguaje de programación interpretado, interactivo y orientado a objetos, que incorpora módulos, excepciones, tipado dinámico, con tipo de datos de alto nivel y clases. Python combina un poder notable con una sintaxis muy clara. Tiene interfaces para muchas bibliotecas y llamadas de sistema, así como para varios sistemas de ventanas, y es extensible en C o C ++. También se puede usar como un lenguaje de extensión para aplicaciones que necesitan una interfaz programable. Finalmente, Python es portátil: se ejecuta en muchas variantes de Unix, en Mac y en PC bajo MS-DOS, Windows, Windows NT y OS / 2”.

4.5.2 XML (Extension Markup Language)

Es un lenguaje de marcado utilizado para almacenar datos de forma legible, creado en 1998 por la World Wide Web Consortium (W3C) (W3C World Wide Web Consortium, 2016), deriva del lenguaje SGML (ISO 8879).

4.5.3 SOAP (Simple Object Access Protocol)

Es un protocolo que define cómo dos objetos pueden comunicarse en diferentes procesos, utiliza el intercambio de datos XML y son utilizados por los servicios Web; este protocolo fue creado por Microsoft con base a uno originalmente creado por Dave Winer en el año 1998 llamado XML-RPC.

4.5.4 HTTP (Hiper Text Transfer Protocol)

Se trata de un conjunto de pautas que posibilitan la comunicación entre las distintas partes del sistema cliente - servidor para que puedan intercambiar información. Se trata de reglas o estándares que definen la sintaxis, semántica y sincronización de la comunicación, así como posibles métodos de recuperación de errores. Pueden ser implementados por hardware, software o una combinación de ambos.

4.5.5 REST (REpresentational State Transfer)

Es un estilo de arquitectura creado en el año 2000 que es frecuentemente usado para permitir la comunicación entre software distribuido a lo largo de una red, también desarrolla una interfaz web simple utilizando XML y HTTP, el término

proviene de la tesis doctoral (Fielding, 2000) de Roy Fielding; este protocolo es preferido en vez de SOAP debido a su bajo peso y por ende requerir menos ancho de banda.

4.5.6 JavaScript

El Java es un lenguaje de programación que se usa en el desarrollo de páginas web dinámicas. Podemos decir que es un lenguaje de programación interpretado, es decir, no requiere de un compilador o software especial para correrlo.

Todos los programas escritos bajo Java pueden correr en los diferentes navegadores como Chrome, Internet Explorer, Mozilla, entre otros, sin necesidad de hacer procesos intermedios.

Su desarrollo se inició en el año de 1995 con el navegador Netscape, que permitía ver páginas web un poco más rápido que los navegadores anteriores. Netscape hizo alianza con Sun microsystems y JavaScript se convirtió en el lenguaje de preferencia de los programadores a nivel mundial (Sun microsystems, 2018).

El esquema de funcionamiento del proceso de consulta de una página web es el siguiente, las flechas azules representan el uso de algún protocolo o lenguaje.



Fig. 18. Proceso de consulta WEB

Inicia en 1. El usuario solicita la página, 2. El servidor recibe la petición, 3. El servidor envía la página, 4. El usuario recibe la página.

4.6 SERVICIOS GEOGRAFICOS

Los servicios SIG en la Web son servicios web automatizados de información geográfica expuestos por medio de los protocolos SOAP y REST, siendo el último el más utilizado para aplicaciones web geográficas, permitiendo exponer recursos geográficos tales como mapas, globos, localizadores de direcciones, redes geométricas, geoprosesamiento, imágenes entre otros.

Este mismo procedimiento se hace para una solicitud a un servidor web que aloja datos geográficos. El objetivo de un SIG en la WEB es facilitar a usuarios el uso

de las funcionalidades de un SIG de escritorio sin la necesidad de adquirir un hardware y un software especializado para tal fin.

4.6.1 Clasificación general de servicios SIG

Los servicios se pueden clasificar en dinámico o estático. Los cuales tienen las siguientes características.

- **Servicios Estáticos:** Generalmente son vistas o imágenes en formato JPEG, TIFF de un área determinada y sirven para representar una información que no cambia con el tiempo. Y sobre ella superponer información vectorial de interés. Un ejemplo claro de este tipo de mapas es el sistema de emergencia de una ciudad. La información del mapa permanece estática mientras que la información útil al usuario se presenta en otro formato.
- **Servicios Dinámicos:** Estos mapas se actualizan casi en línea y permiten visualizar información geográfica de manera ágil y efectiva. Además, el acceso a la información se da a diferentes escalas y acercamientos y el usuario puede acercarse y alejarse sobre la cartografía. Un ejemplo de este tipo de mapas son las aplicaciones de navegación móvil que continuamente están cambiando la información a medida que el usuario se mueve dentro del mapa.

4.6.2 Servicio de imágenes dinámico

La definición es realizada por Esri (Environmental Systems Research Institute, 2018) “ArcGIS Image Server sirve imágenes y colecciones de datos ráster como servicios de imágenes dinámicos con procesamiento y creación de mosaicos al vuelo. Use ArcGIS Image Server para:

- Imágenes de satélite, áreas y de drones
- Ortofotos y raster de categorías
- Modelos de elevación y datos LIDAR

Estos servicios de imágenes usan data sets de mosaico para administrar imágenes superpuestas, temporales y multirresolución provenientes de varios formatos, fuentes y proyecciones. Las imágenes resultantes se pueden compartir fácilmente con una amplia variedad de aplicaciones de escritorio, web y móviles. Los colaboradores pueden acceder a productos predefinidos o interactuar con los servicios para acotar la representación y el procesamiento que se devuelven. Con los servicios de imágenes dinámicas, el contenido completo de la información de las imágenes está accesible más rápido sin necesidad de un procesamiento previo”.

4.6.3 Servicios de geoprocésamiento

Los servicios de geoprocésamiento representan el modo en que expone la capacidad analítica de ArcGIS en la Web. Los servicios de geoprocésamiento se crean a partir de modelos o scripts Python que contienen herramientas de

geoprocesamiento encadenadas, la ejecución se realiza mediante tareas, esta toma datos simples capturados en una aplicación web, los procesa y devuelve resultados significativos y útiles en forma de características, mapas, informes y archivos. Las posibilidades de estos servicios son ilimitadas.

4.7 SIG WEB

Con el avance de la tecnología computacional y de redes, el avance en el “cloud Computing” y las aplicaciones móviles, los SIG ven un panorama amplio para ser más colaborativo y eficiente. La funcionalidad de administración, ingreso de datos y exportación o visualización de datos cambia hacia un panorama a nivel mundial. La posibilidad de compartir o recibir datos de servicios de mapas del clima desde cualquier estación del mundo y utilizarlos como entrada de datos para generar mapas más regionales es un ejemplo claro de la funcionalidad de compartir información a través del internet.

Los usuarios SIG se han vuelto más especialistas en las diferentes áreas del conocimiento espacial gracias a la masificación de dicha tecnología en la red. Expertos de todas partes del planeta pueden intervenir, opinar, apoyar con ideas, conceptos o datos fuente que enriquezcan nuestro sistema de información geográfica y esto se debe gracias a la colaboración en la nube.

Existen diferentes tipos de SIG en la Web: el SIG corporativo, el SIG para comunidades y el SIG para participación ciudadana.

4.7.1 SIG corporativo

En este tipo de SIG, tanto los mapas y los datos son creados o publicados teniendo en cuenta los objetivos de la corporación o entidad. Destinados a dar cumplimiento a cada uno de los procesos del Core de negocios y diferenciando la información que llega a directores, gerentes y otros funcionarios que no les interesa los pormenores de la creación, edición de la información sino solamente el resultado final.

4.7.2 SIG para comunidades

Es un tipo de SIG más colaborativo, generalmente se agrupan por necesidades en común o características geográficas parecidas. Un ejemplo claro es el SIG web para la (Observatorio de Territorios Etnicos y Campesinos, s.f.) defensa de los territorios étnicos y campesinos, que es un SIG destinado a la divulgación y promoción de la cartografía social y colaborativa entre las comunidades afrodescendientes, indígenas y campesinas.

4.7.3 SIG para la participación ciudadana

Es un SIG destinado a la función pública. Presentado a usuarios internos o externos que desean información relevante a su ciudad. Un ejemplo interesante lo podemos ver en el geo portal (Geovisor IDESC, 2018) **IDESC** de la Alcaldía de

Santiago de Cali, un portal donde se pueden descargar información cartográfica, mapas de ordenamiento territorial de la ciudad, geo servicios de puntos de control GPS de la ciudad con sus certificados y ubicación y otras utilidades que le son útiles al ciudadano.

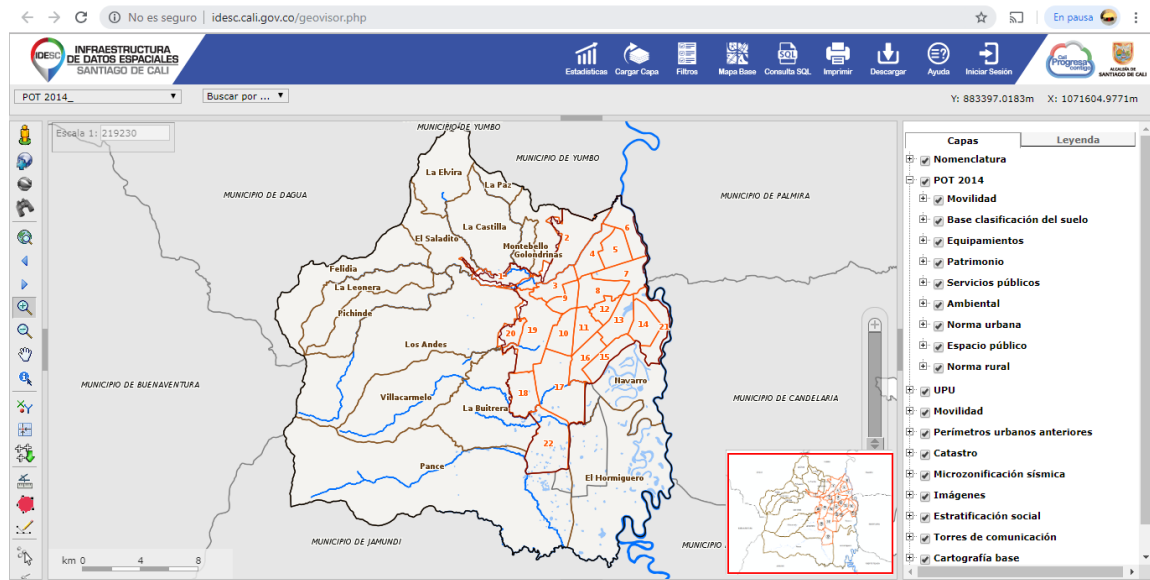


Fig. 19. Portal ciudadano Idas Cali, tomado de (Geovisor IDESC, 2018)

Otro de los puntos a tener en cuenta es la estructura del SIG Web, es decir, cómo se relacionan con el trabajo diario, la corporación y los mapas para la organización. Para identificar esta estructura se tienen los siguientes roles del SIG web:

Rol administrador: Es la persona que administra el SIG, crea a los usuarios y le asigna los roles a cada uno.

Rol publicador: son los encargados de crear los mapas y las aplicaciones para que sean usadas por los usuarios de la organización.

Rol usuario: son los que consultan los mapas y las aplicaciones y en muchos casos pueden compartir la información con otros usuarios internos y externos de la organización.

4.7.4 SIG móviles (Mobile GIS)

Gracias a los SIG móviles, los mapas y las aplicaciones van a donde vaya el usuario.

Esri en su libro ArcGIS book realiza la siguiente definición (Environmental Systems Research Institute, 2015) “La integración de los Smart phones y los SIG tiene muchas aplicaciones, puedes utilizar tu teléfono para capturar fotos y vídeos geotiquetados y después, utilizarlos para narrar y compartir historias. Es posible recopilar datos sobre el terreno y actualizar la información de tu empresa. El teléfono también se puede usar para acceder a la información empresarial correspondiente a la ubicación actual, para disponer de más información y mejorar la concienciación”.

Un ejemplo práctico de los SIG móviles y de las plataformas geográficas en la web es la aplicación llamada (Geoprocess SAS, 2018) GisData. Esta aplicación ha sido desarrollada y diseñada para captura e inventario de información de diversa índole. Como por ejemplo inventario de especies arbóreas, censos de viviendas, inventario de postes y apoyos de energía, censos para geo mercadeo, etc.



Fig. 20. Aplicación GisData

La aplicación se instala en un Smartphone con sistema operativo Android, preferiblemente con pantalla mayor a 5”. Además de la instalación existe una plataforma web que administra, visualiza y crea los formularios para luego ser enviados a los celulares de los colaboradores o censadores.

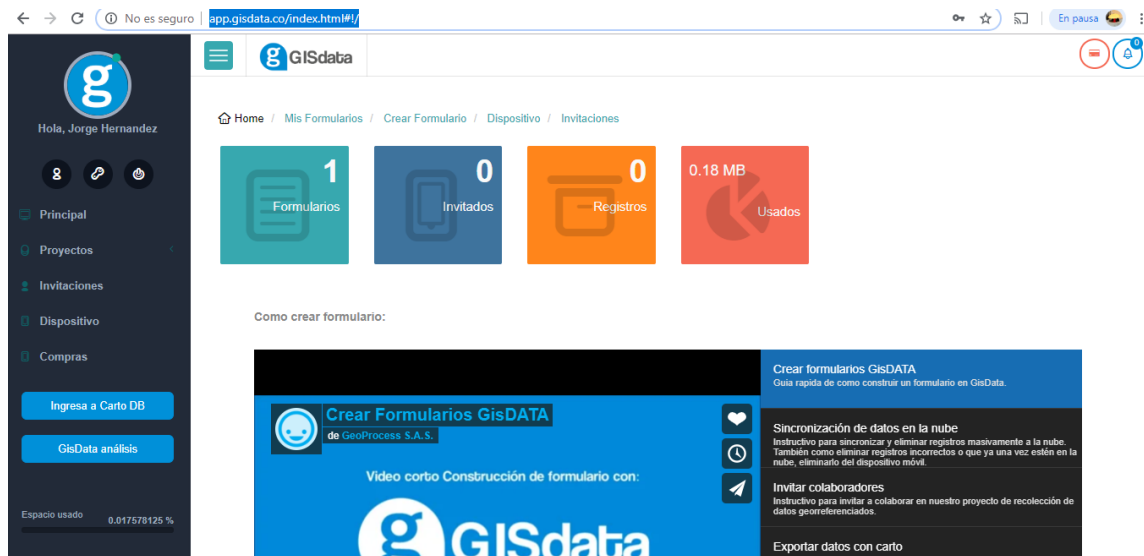


Fig. 21. Plataforma GisData

La interfase cartográfica es realizada a través de (CARTO DB, s.f.) CARTO, que es una plataforma de cartografía base que está embebida dentro de la plataforma web de GisData y permite recibir la información espacial proveniente de la sincronización de los formularios y le da un marco de referencia espacial a nuestros datos capturados con la aplicación GisData.

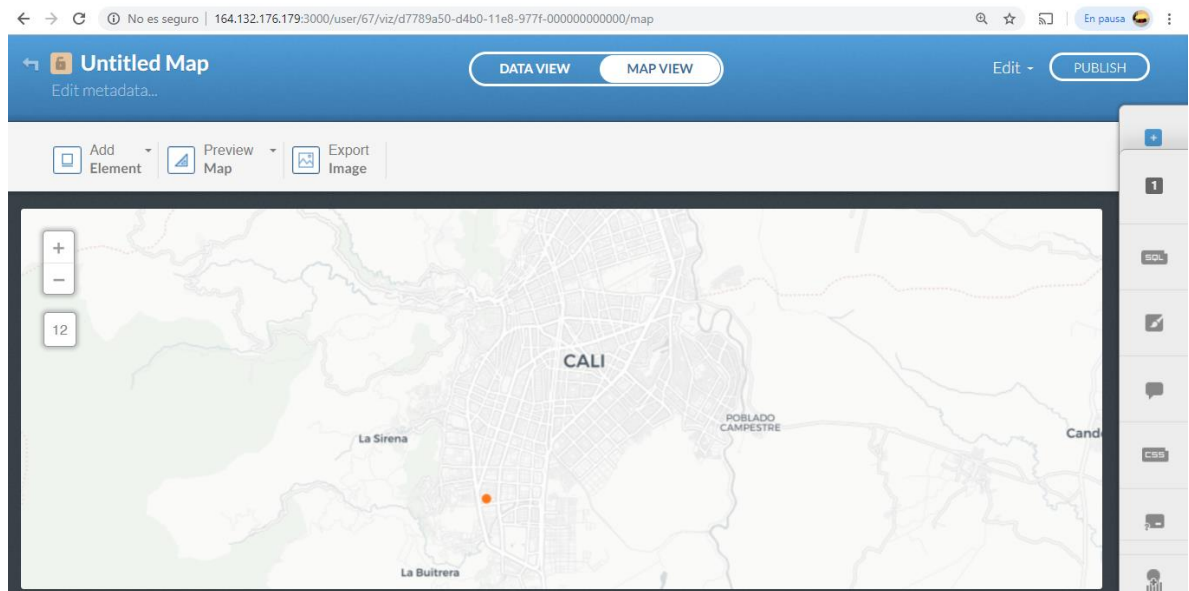


Fig. 22. Plataforma CARTO DB

4.8 PORTALES GEOGRÁFICOS

La palabra portal viene del latín *porta* que significa *Puerta o portón*, en términos nuestros podemos decir que un portal es un sitio web con acceso al World Wide Web. Una definición de portal geográfico la realiza *Tait* (Tait, 2005) “un sitio web que provee un punto de acceso simple a los datos geoespaciales, servicios web y otros recursos geoespaciales. Puesto más simple, un geo portal es un sitio web donde los recursos geoespaciales pueden ser descubiertos”.

Las organizaciones que más se han beneficiado de los portales geográficos son las fundaciones y comunidades de protección del medio ambiente. Ellos han sacado gran provecho de recursos geográficos y han generado cartografía propia dentro de sus portales para crear conciencia ambiental a nivel mundial.

Pero el servicio que ha masificado más el uso de este tipo de herramientas es Google Maps. Este portal junto con su servicio de aplicaciones para la Web y para el celular, hacen de esta aplicación una de las usadas a nivel mundial.

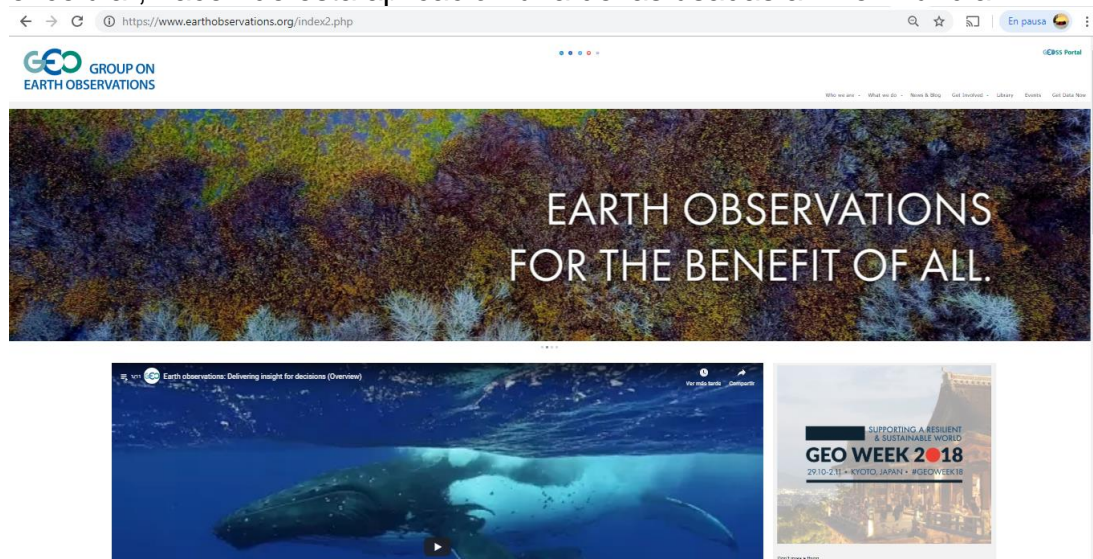


Fig. 23. Geoportal GEO

4.8.1 Funciones de los Geoportales

Al igual que los roles de los servicios SIG, los geoportales también tienen las funciones de administración, publicación y de usuario.

- **Administración:** Son los encargados de administrar las cuentas de usuario, los metadatos, la seguridad del portal, administran la información que se puede visualizar y la que no.

- **Publicación:** son los encargados de publicar la información, crear y editar los metadatos, publicar los metadatos a los usuarios. También pueden unirse a nuevos usuarios o grupos para compartir información.
- **Usuario:** consultan los servicios de mapas, bajan datos y los comparten, consultan información espacial y de atributos, visualizan los metadatos y buscan nueva información de otras fuentes de metadatos. También pueden abrir otros portales y combinar información local con información de la página.

4.9 ANTECEDENTES

A continuación, se listan el marco normativo más relevante y, los planes de acción trienal que dieron origen al sistema de información geo-referenciado corporativo que más adelante sería el portal geográfico GeoCVC.

4.9.1 Marco normativo del Proyecto

Las siguientes leyes son las más relevantes que justifican la realización del proyecto:

Ley 99 de 1993 (Congreso de la Republica de Colombia, 2018) la que señala las guías de acción del Ministerio del Medio Ambiente. (...) Ejercer las funciones de evaluación, control y seguimiento ambiental de los usos del agua, el suelo, el aire y los demás recursos naturales renovables.

- Artículo 31 Funciones de las Corporaciones Autónomas numeral 22 “implantar y operar el sistema de información ambiental en el área de su jurisdicción, de acuerdo con las directrices trazadas por el Ministerio del Medio Ambiente”.

Ley 1712 de 6 marzo de 2014 (Congreso de la Republica de Colombia, 2014) “ley de transparencia y del derecho al acceso a la información pública nacional”, la cual determina que se debe garantizar el acceso a toda información capturada o producida por entidades públicas a excepción de información clasificada o reservada. Los artículos más relevantes para el proyecto con los siguientes:

- “Artículo 2. PRINCIPIO DE MÁXIMA PUBLICIDAD PARA TITULAR UNIVERSAL. Toda información en posesión, bajo control o custodia de un sujeto obligado es pública y no podrá ser reservada o limitada sino por disposición constitucional o legal, de conformidad con la presente ley”.
- Artículo 3. OTROS PRINCIPIOS DE LA TRANSPARENCIA Y ACCESO A LA INFORMACIÓN PÚBLICA. En la interpretación del derecho de acceso a la información se deberá adoptar un criterio de razonabilidad y proporcionalidad, así como aplicar los siguientes principios:
 - Principio de transparencia. Principio conforme al cual toda la información en poder de los sujetos obligados definidos en esta ley se presume pública, en consecuencia, de lo cual dichos sujetos están en el deber de proporcionar y facilitar el acceso a la misma en los términos más amplios posibles y a través de los medios y procedimientos que al efecto establezca la ley, excluyendo solo aquello que esté sujeto a las excepciones constitucionales y legales y bajo el cumplimiento de los requisitos establecidos en esta ley.

- Principio de facilitación. En virtud de este principio los sujetos obligados deberán facilitar el ejercicio del derecho de acceso a la información pública, excluyendo exigencias o requisitos que puedan obstruirlo o impedirlo.
- Principio de celeridad. Con este principio se busca la agilidad en el trámite y la gestión administrativa. Comporta la indispensable agilidad en el cumplimiento de las tareas a cargo de entidades y servidores públicos.
- Principio de eficacia. El principio impone el logro de resultados mínimos en relación con las responsabilidades confiadas a los organismos estatales, con miras a la efectividad de los derechos colectivos e individuales.
- Principio de la divulgación proactiva de la información. El derecho de acceso a la información no radica únicamente en la obligación de dar respuesta a las peticiones de la sociedad, sino también en el deber de los sujetos obligados de promover y generar una cultura de transparencia, lo que conlleva la obligación de publicar y divulgar documentos y archivos que plasman la actividad estatal y de interés público, de forma rutinaria y proactiva, actualizada, accesible y comprensible, atendiendo a límites razonables del talento humano y recursos físicos y financieros”.

4.9.2 Sistema de Información Geo-referenciado Corporativo GeoCVC

En el año 2004, la CVC inició la implementación del Sistema de Información Geo-referenciado Corporativo, el cual fue planeado en varias fases.

La primera se formuló en el Plan de Acción Trienal (PAT) 2004-2007, mediante el sub-proyecto No 1081, en el que se efectuó el Análisis y Diseño. La segunda fase se desarrolló en el PAT 2007 - 2009, mediante el proyecto 1524 se complementó el diseño de la fase uno y se desarrolló un visor web para la consulta de datos geográficos. Es por ello que, a partir del segundo semestre del año 2009, la CVC cuenta con un visor geográfico que permite consultar la información espacial ambiental.

En las fases 1, 2 y 3 del Sistema de Información Geográfico permitió diseñar y crear una base de datos geográfica para el almacenamiento de datos organizados bajo un enfoque ecosistémico, la creación de un visor (aplicativo) para la consulta de datos geográficos y un subsistema para el manejo de datos de la red hidro-climatológica.

4.9.3 Proyectos LiDAR y radar

En el año 2012 la CVC publicó la licitación para el servicio de restitución cartográfica para el ordenamiento territorial y la zonificación de amenazas en las cabeceras municipales del Valle mediante técnicas de fotogrametría y LiDAR. La unión temporal **Ingeovista - FIT Conseil** fue la empresa que realizó dicho proyecto en un término de 16 meses.

El producto principal del contrato 0403 del 2012 era “programar y tomar las fotografías aéreas, digitales y a color de 15 cm de resolución, de 30 centros poblados del departamento del Valle del Cauca y el corredor del río Cauca”



Fig. 24. Modelo de sombras generado a partir de LiDAR del corredor Río Cauca, Zona Juanchito

Posteriormente en el 2014, la CVC solicitó el servicio de toma de imágenes empleando técnicas de radar. El objeto del contrato era “Generar cartografía básica (escala 1:10.000), un modelo digital de terreno (MDT), un modelo digital de superficie (MDS) e imágenes de radar orto rectificadas a partir de datos obtenidos mediante un radar aerotransportado para diferentes cuencas del Valle del Cauca y del Cauca”.

El producto principal del contrato 0485 del 2014 era obtener Imágenes de Radar Orto rectificadas (ORI) de banda X y P en 8 y 16 bits, e imagen a color en 24 bits, dividida por cuencas, correspondiente al elemento J) Imágenes de Radar Orto rectificadas (ORI) mínimo de 16 bits de toda el área de estudio. La duración del proyecto fue de 12 meses.



Fig. 25. Orto imagen en banda X

Una orto-imagen de banda X es una imagen orto rectificadas obtenida por la amplitud en la faja de frecuencia de la microondas conocida como Banda X cuya frecuencia central varia de 9,35GHz e 9,75GHz y longitud de onda entre 3,75 a 2,5 cm.

5. METODOLOGÍA

5.1 TIPO DE TRABAJO

El presente trabajo corresponde a un desarrollo tecnológico en las áreas de las aplicaciones web geográficas y sistemas de información geográficos el cual busca integrar los datos ráster al portal geográfico de la CVC GeoCVC.

5.2 LEVANTAMIENTO DE REQUERIMIENTOS

Esta actividad se realizó mediante reuniones con los funcionarios del grupo sistema de información ambiental, en las cuales se escucharon las funcionalidades requeridas o deseadas en el visor geográfico ráster, incluido los procesos realizados por los funcionarios concernientes a los modelos digitales de elevación, las herramientas de geoprocésamiento empleadas o los modelos derivados. Como resultado se generó la Tabla 2 que sintetiza los requerimientos factibles de realizar en el proyecto.

Tabla 2 Requerimientos sintetizados y su tipo

#	Requerimiento	Tipo
1	Visualización en dispositivos móviles y computadores de escritorio	Visualización
2	Simbología dinámica dependiente del rango de alturas desplegada	Visualización
3	Capacidad para generar modelos derivados a partir de MDE o MDT	Análisis
4	Capacidad para crear un perfil topográfico desde la aplicación web geográfica	Análisis
5	Capacidad para exportar el abdicado y las cotas en formato tabular	Análisis
6	Capacidad para ingresar coordenadas planas magna	Análisis
7	Servicios integrables con software SIG	Funcional
8	Integrar el visor al portal geográfico GeoCVC	Funcional

5.2.1 Definición del entorno gráfico

Este se encuentra definido por la suma de las extensiones geográficas de los ráster que serán desplegados en la aplicación, la cual incluye principalmente al departamento del Valle del Cauca en su totalidad, y la parte norte del departamento del Cauca.

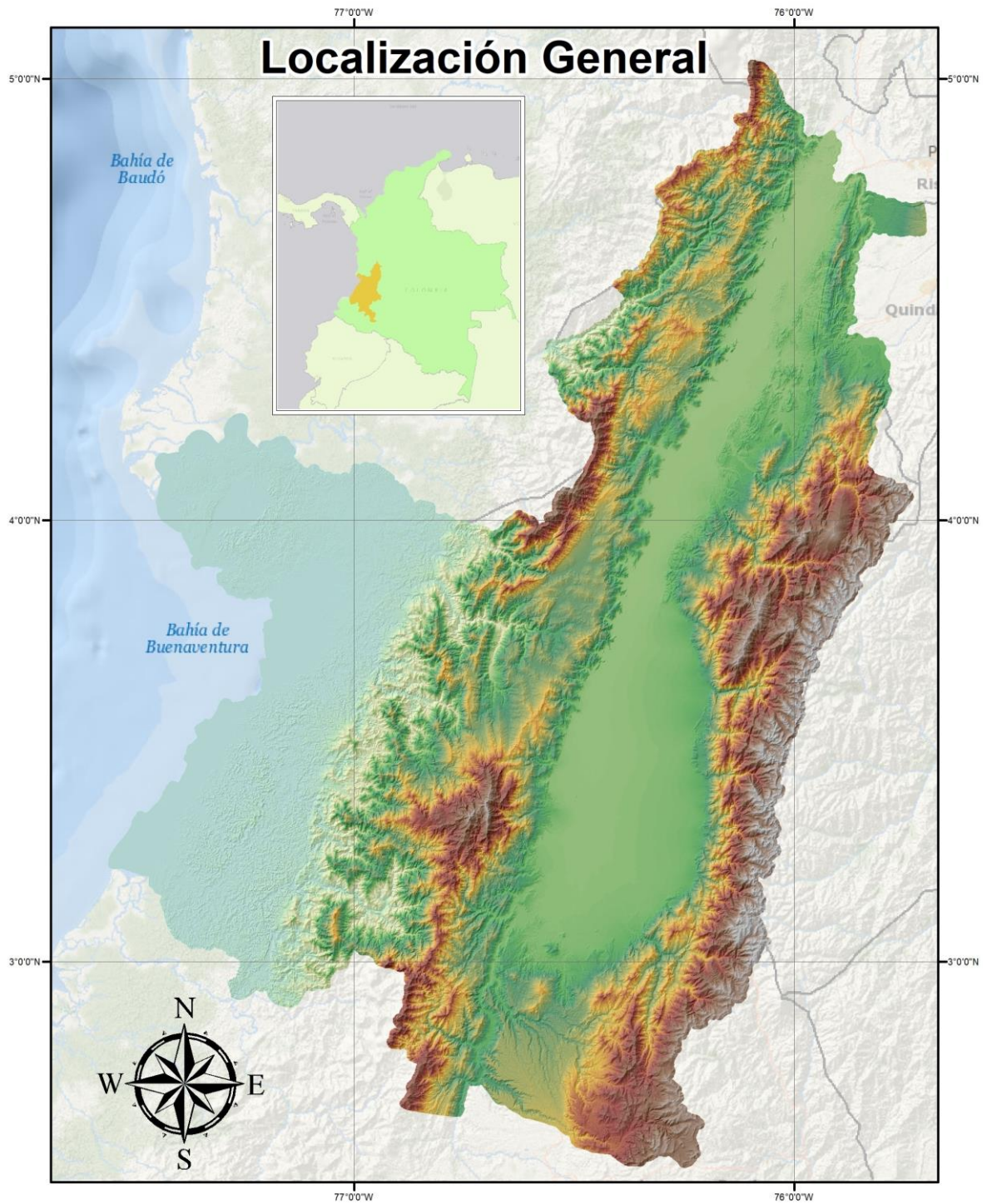


Fig. 26. Extensión geográfica de los Datos

5.2.2 Alcances de la aplicación geográfica

Para la realización de este proyecto se presentan 2 tipos de funciones, visualización cartográfica directa y análisis ráster.

- **Visualización cartográfica directa.** El usuario tendrá el control de visualización de las capas desplegadas en el mapa mediante controles para el encendido y apagado de las mismas, transparencia y también dispondrá de herramientas para la navegación dentro de la aplicación.
- **Análisis ráster.** Las funciones de análisis estarán en dos entornos: equipos con software geográfico ArcGIS Desktop el cual cuenta con un conjunto de funciones que se pueden aplicar a los servicios de imágenes publicados; la aplicación web la cual integrará herramientas de análisis específicos como generación de perfil topográfico y exportar a formato tabular los datos de abscisas y elevaciones realizadas con la herramienta perfil que estarán disponibles vía un botón en la interfaz.

5.2.3 Limitaciones de la aplicación geográfica

Toda la implementación se realizará utilizando la plataforma ArcGIS, debido a que los visores geográficos actuales se han creado utilizando varios de estos productos.

5.2.4 Definición detallada de la información

En este paso se listan aspectos específicos de la información requerida para el funcionamiento integral de la aplicación, basados en los requerimientos iniciales de los usuarios y en los requerimientos de las herramientas de análisis.

- **Servicios de mapas base.** La cartografía base utilizada serán los servicios de mapa base de ArcGIS online, se integrará un paquete de teselas vectoriales para clases de entidad de referencia que no se encuentran en los mapas base.
- **Servicios de imágenes.** Están basados en conjuntos de datos de mosaicos o ráster individuales y a partir de estos se publicarán los servicios imágenes, también se contará con un servicio con la clase de entidad de puntos, producto de la intersección de varias clases de entidad temáticas de la GDB.
- **Servicios de geoprocésamiento perfil.** Está basado en un script Python el cual toma los valores de los MDT de los proyectos de LiDAR y radar y crea el perfil topográfico del terreno en base a una línea dibujada por el usuario.

- **Respuestas esperadas.** Se esperan tres tipos de respuestas: grafica, la cual consiste en el despliegue de los dos tipos de servicios anteriormente descritos; tabular y análisis, son los resultados de la ejecución de funciones aplicadas a los servicios de imágenes.

5.3 MODELO CONCEPTUAL

Dado que el visor básicamente despliega datos ráster, el modelo conceptual no es muy complejo; sin embargo, para dar cumplimiento al proceso de diseño, a continuación, se presentan el modelo conceptual del visor ráster.

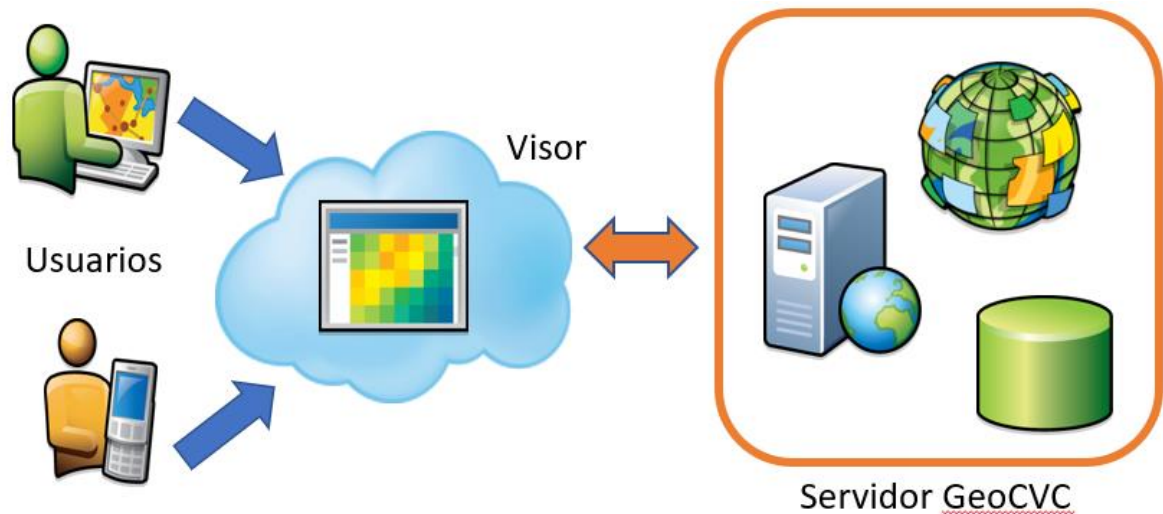


Fig. 27. Modelo conceptual

A partir del modelo conceptual se realizó la abstracción de los elementos del sistema.

- Usuarios (móviles y desktop)
- Conjuntos de datos ráster (MDE y fotografías aéreas)
- Servidor (SIG, WEB y de aplicaciones)
- Servicios de imágenes
- Mapas base
- Visor Geográfico

Con base a los requerimientos se determinó que la aplicación debe estar disponible via internet, permitir la visualización y creación de perfil topográfico a partir de los datos raster MDT radar, del corredor del Río Cauca, y de las cabeceras municipales contenidas de forma centralizada en el servidor GeoCVC.

5.4 MODELO LÓGICO

Dado que el visor básicamente despliega datos ráster, el modelo lógico no es muy complejo. Basta con describir los esquemas de los conjuntos de datos de mosaico o raster y las capas que utiliza el servicio de geoprocesamiento; los requisitos funcionales incluyen el entorno operativo, la definición de las relaciones, los procesos entre los componentes y la identificación de las entidades del visor.

5.4.1 Entorno operativo

La visualización de la aplicación se realiza en dispositivos móviles y equipos de escritorio, no se aplica restricción de acceso al visor como se realizó en otras aplicaciones que manejan información sensible destinada solo a funcionarios.

5.4.2 Definición de relaciones y procesos

No se presentan relaciones en el sentido estricto de una base de datos relacional, el visor tiene dos tipos de relaciones con los datos, una se presenta con los servicios de imágenes, los cuales leen de forma directa los datos ráster (fotografías áreas, MDTs y modelos derivados) alojados en el servidor, la segunda la realiza el servicio de geoprocesamiento, este emplea una estructura de datos definida, compuesta por los MDTs y dos capas vectoriales, una tipo línea y una capa de polígono. Otro de los procesos se lleva a cabo cuando se exporta las abscisas y cotas obtenidos por el perfil topográfico a un archivo tabular.

5.4.3 Identificación de las entidades

Los funcionarios del grupo SIA proporcionaron la información de los proyectos de LiDAR y radar los cuales se encuentran en el sistema de coordenadas MAGNA Colombia Oeste y se describen en la tabla 3.

Tabla 3 Datos suministrados para la construcción del visor

Nombre	Tipo	Descripción
MDT corredor Río Cauca	ráster	Modelo digital de elevación del corredor del Río Cauca de resolución espacial de 1m desde la represa de Salvajina hasta el municipio de Cartago
MDTs de los centros poblados	ráster	MDTs de las cabeceras urbanas de los municipios de Alcalá, Argelia, Caicedonia, Candelaria, Darién, El Águila, El Cerrito, El Dovio, Ginebra, Guacarí, La Cumbre, Restrepo, Sevilla, Trujillo, Ulloa y Versalles de resolución espacial de 1m
MDT radar	ráster	Modelo digital de elevación del departamento del Valle del Cauca más las cuencas de la parte norte del departamento del Cauca con resolución espacial de 2,5m
Fotografías aéreas del corredor del Río Cauca	ráster	Fotografías áreas del corredor del Río Cauca de resolución espacial de 15cm, desde la represa de salvajina hasta el municipio de Cartago y en formato ECW
Fotografías aéreas de los centros poblados	ráster	Fotografías áreas de las cabeceras urbanas de los municipios de Alcalá, Andalucía, Ansermanuevo, Argelia, Bolívar, Bugalagrande, Caicedonia, Candelaria, Darién, El Águila, El Cerrito, El Dovio, Ginebra, Guacarí, Jamundí, La Cumbre, La Unión, La Victoria, Obando, Restrepo, Riofrío, Roldanillo, Sevilla, Toro, Trujillo, Tuluá, Ulloa, Versalles, Vijes, Yotoco y Zarzal de resolución espacial de 15cm y en formato ECW
Radar Falso Color	ráster	Orto imagen a color generada mediante la fusión de las orto imágenes de la banda X y la banda P, en la cual la textura la aporta la banda X y el color la banda P. El cálculo de las sombras se obtiene del modelo de sombra del MDS

A partir de la información proporcionada se definirán las entidades necesarias para cumplir los requerimientos de la aplicación geográfica.

5.4.4 Clases de entidad (tabla espacial)

Los esquemas que se describen a continuación conforman los atributos de las clases de entidad dentro de una base de datos geográfica de archivos que fue alojada en el servidor.

- **Demboundary.** Capa vectorial con los límites de los MDTs en los que la herramienta puede generar perfiles.

Tabla 4 Esquema de la clase de entidad Demboundary

Nombre Campo	Tipo	Descripción
ObjectID	Object ID	Identificador creado por el sistema
Shape	Geometría	Campo de geometría
res	Entero Largo	Resolución espacial del MDT en esa zona
Shape_Length	Doble	Valor perímetro del polígono que representa la extensión del ráster
Shape_Area	Doble	Valor del área del polígono que representa la extensión del ráster

- **Profileschema.** Capa vectorial para alojar la línea dibujada (de forma temporal) por el usuario en la aplicación, con la cual se generará el perfil

Tabla 5 Esquema de la clase de entidad Profileschema

Nombre Campo	Tipo	Descripción
ObjectID	Object ID	Identificador creado por el sistema
Shape	Geometría	Campo de geometría
Name	Texto 60 caracteres	Campo para ingresar un nombre a la línea creada
Shape_Length	Doble	Longitud de la línea ingresada por el usuario

5.5 MODELO FISICO

El modelo físico para la implementación del visor ráster consiste en la plataforma del servidor SIG externo existente, la cual tiene las siguientes características.

Servidor de servicios SIG y portal geográfico. En este servidor se instalarán las aplicaciones de ArcGIS Enterprise: servidor de imágenes, servidor de mapas y portal las cuáles serán las que alojen los servicios geográficos (tanto de imágenes como de mapas), para el despliegue de los servicios de imágenes la información cartográfica base, la capa de puntos para consultas de atributos y la aplicación geográfica como tal; se realizó previamente un dimensionado para determinar los

recursos de hardware necesarios para soportar la cantidad de transacciones, redundancia y alta disponibilidad requeridas, las características del hardware son: servidor HP Proliant con procesador Intel Xeon con 16 Cores, memoria RAM de 64 Gb y un arreglo de discos de 15 Tb raid5 de los cuales se han destinado 5Tb para el almacenamiento de los datos ráster y 1Tb para los directorios de ArcSIG server y portal.

Estación de trabajo. Este equipo contará con un procesador de 8 Cores, 16 Gb en RAM y un 1Tb en almacenamiento en disco. Se establecerá una conexión al servidor SIG y portal geográfico para subir el conjunto de datos necesarios para su funcionamiento.

Esquema de software empleado. A continuación, se relaciona el software utilizado para la implementación del visor ráster:

- Sistema operativo Windows 10 (estación de trabajo)
- ArcGIS for Desktop nivel de licenciamiento Advanced versión 10.6
- Navegador web actualizado
- Software de procesamiento de imágenes ENVI
- Sistema operativo Windows server 2012 R2 (servidor SIG y portal geográfico) implementación de rol IIS para habilitar el servidor WEB
- ArcGIS Enterprise nivel de licenciamiento Advanced versión 10.6
- ArcGIS Image server versión 10.6
- ArcGIS WebAdaptor 10.6

5.6 MODELO CARTOGRÁFICO DEL SISTEMA

No fue necesario diseñar un mapa debido a que las fotografías aéreas no requieren que se les aplique simbología alguna, cuando se trata de MDTs la simbología es aplicada al servicio de imágenes directamente; el proceso de publicación de servicios de imágenes difieren de servicios de mapa tradicionales, los cuales parten de un documento mdx. En cambio, los servicios de imágenes se publican a partir de los conjuntos de datos de mosaicos o del ráster en sí y luego sobre el servicio son aplicadas plantillas de funciones que permiten simbolizar el servicio.

5.6.1 Simbología del servicio de MDT

La simbología de este servicio es dinámica, con cada operación de acercamiento, alejamiento o desplazamiento se extrae la extensión geográfica que será visualizada para obtener el valor máximo y el mínimo de elevación sobre el

terreno, y a partir de estos dos valores de elevación, se realiza un estiramiento de la rampa de colores; para realizar esto se creó una plantilla de funciones, el siguiente fue el proceso de creación de dicha plantilla.

Desde ArcMAP se carga el MDT o el *mosaic dataset*, luego se carga la ventana de análisis de imágenes, dentro de esta, se selecciona el MDT para activar las herramientas y luego se ejecuta la herramienta para adicionar funciones.

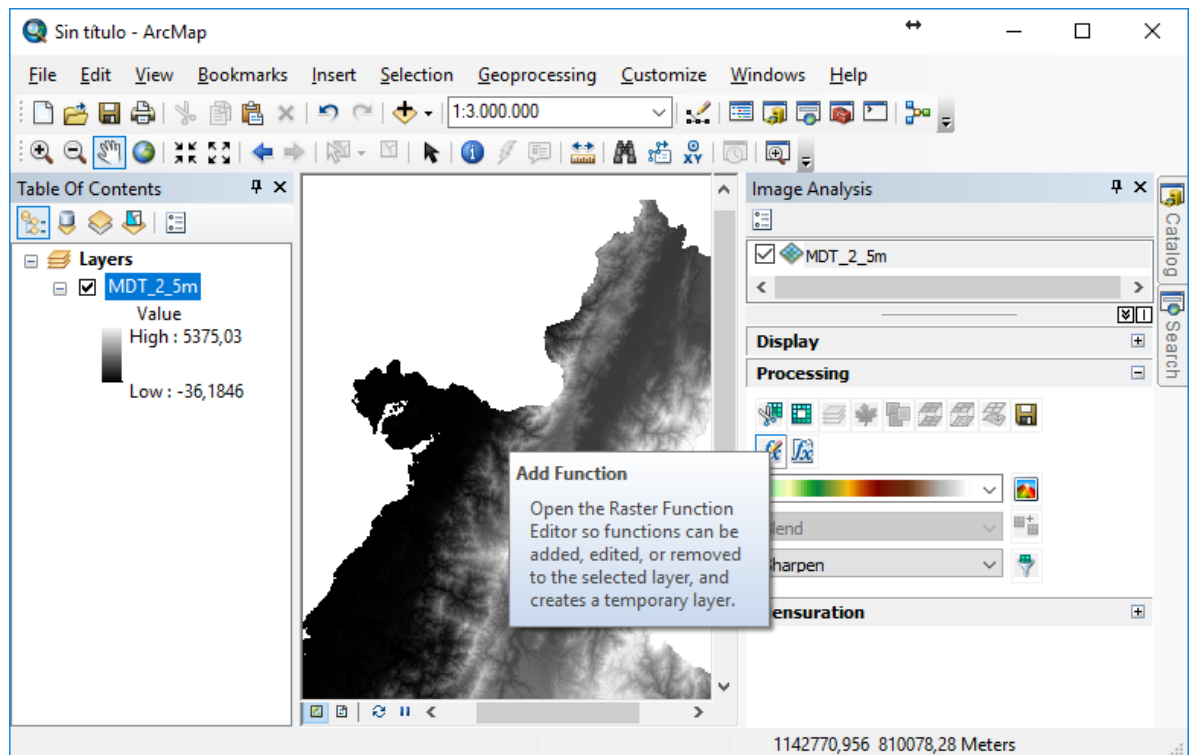


Fig. 28. Inicio del proceso de creación de plantilla función

Se despliega del editor de plantillas de funciones, con la función por defecto.

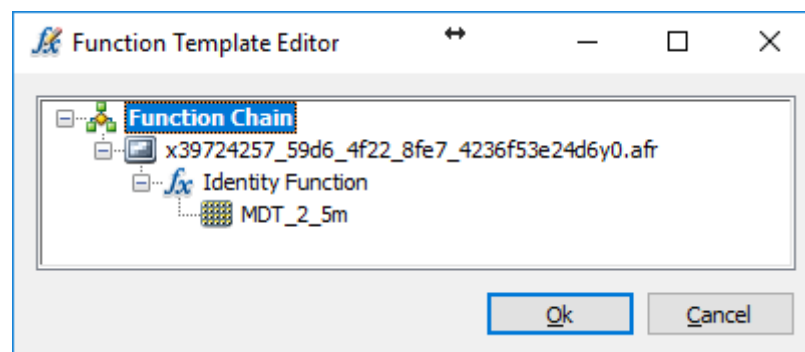


Fig. 29. Editor de plantillas de funciones, función identidad

Dentro del editor de plantillas de funciones, mediante el menú contextual sobre la función de identidad se empieza a insertar funciones. Primero se selecciona la función extender (stretch), en la cual selecciona el tipo en máximo – mínimo y el ajuste de rango dinámico (Dynamic Range Adjustment).

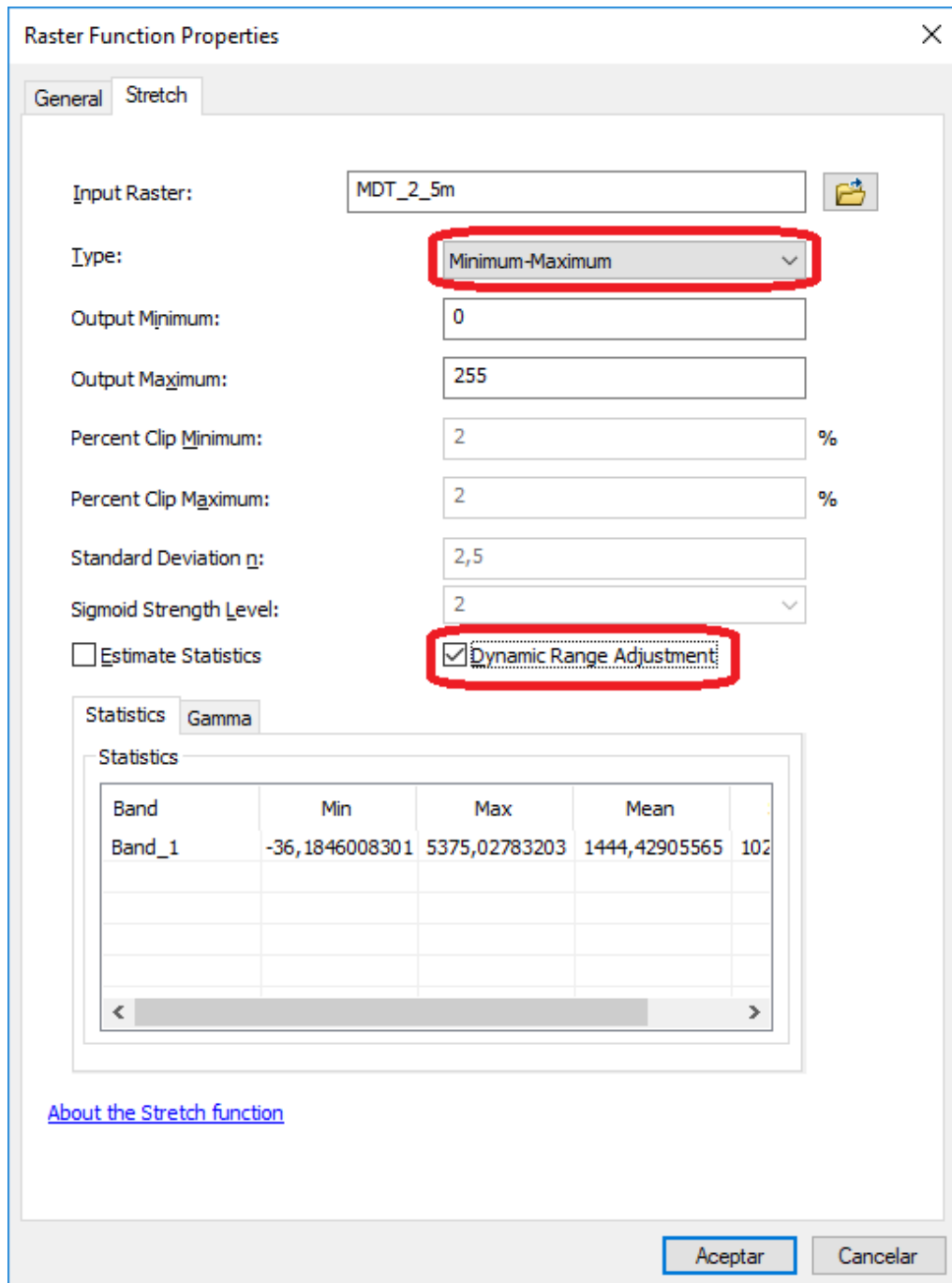


Fig. 30. Parámetros de la función extender (stretch)

Sobre la función Stretch se aplica la función colormap y se edita la rampla de color para quitar el valor extremo que representa las nieves perpetuas

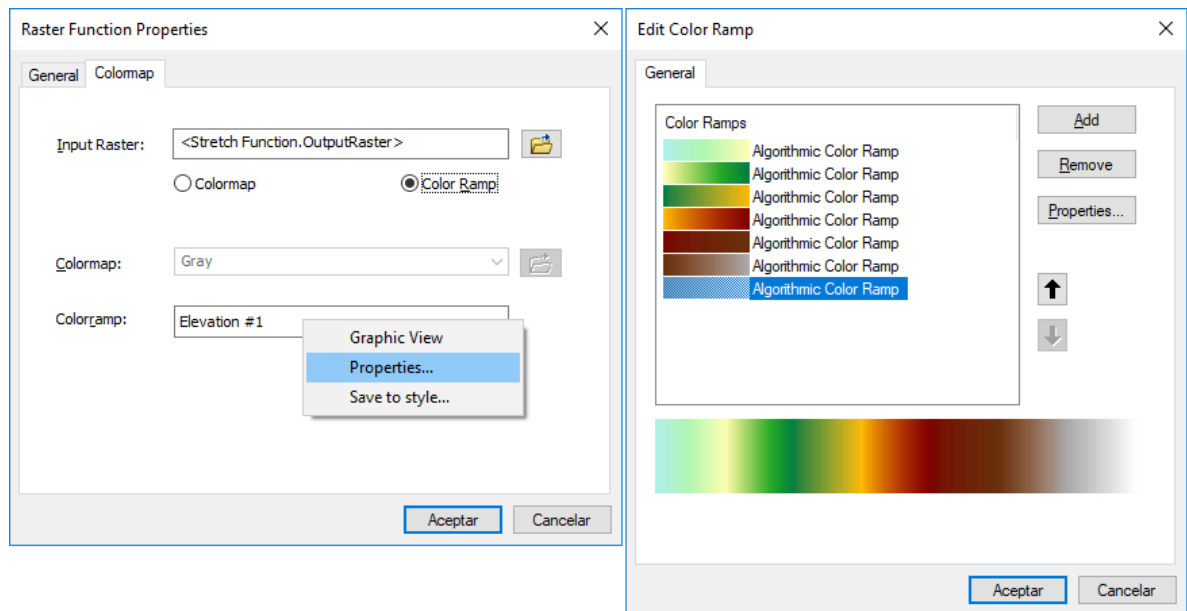


Fig. 31. Parámetros de la función colormap

El resultado son dos funciones encadenadas aplicadas al MDT, esta se exporta como una plantilla de funciones en un archivo XML con extensión RFT y después será aplicada al servicio.

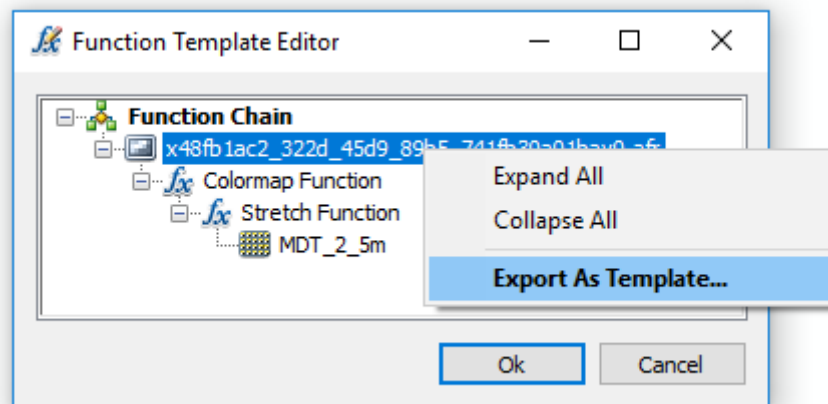


Fig. 32. Funciones encadenadas y exportación de plantilla de funciones

Después, a partir del MDT de radar y del Mosaic dataset de MDTs de LiDAR se publican los servicios. El proceso inicia desde ArcCatalogo, accediendo a la GDB y mediante el menú contextual se selecciona compartir como servicio de imágenes y este inicia el asistente.

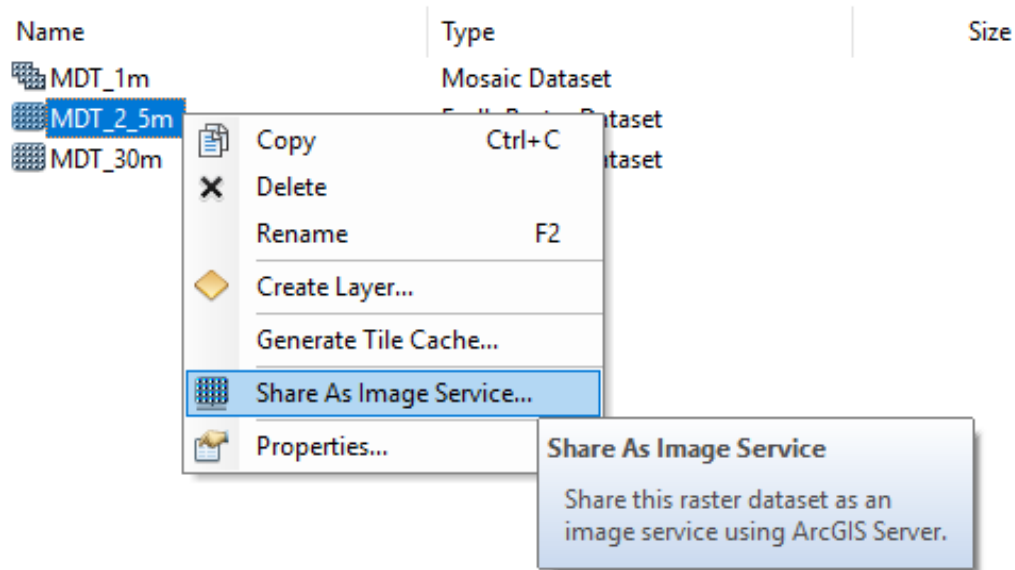


Fig. 33. Inicio del proceso de publicación del servicio de imágenes

En la primera ventana del asistente se selecciona publicar un servicio, en la segunda se selecciona la conexión al servidor *image* y el nombre del servicio, en la tercera ventana se crea o selecciona la carpeta donde se creará el servicio, en la siguiente figura se pueden apreciar las tres ventanas descritas.

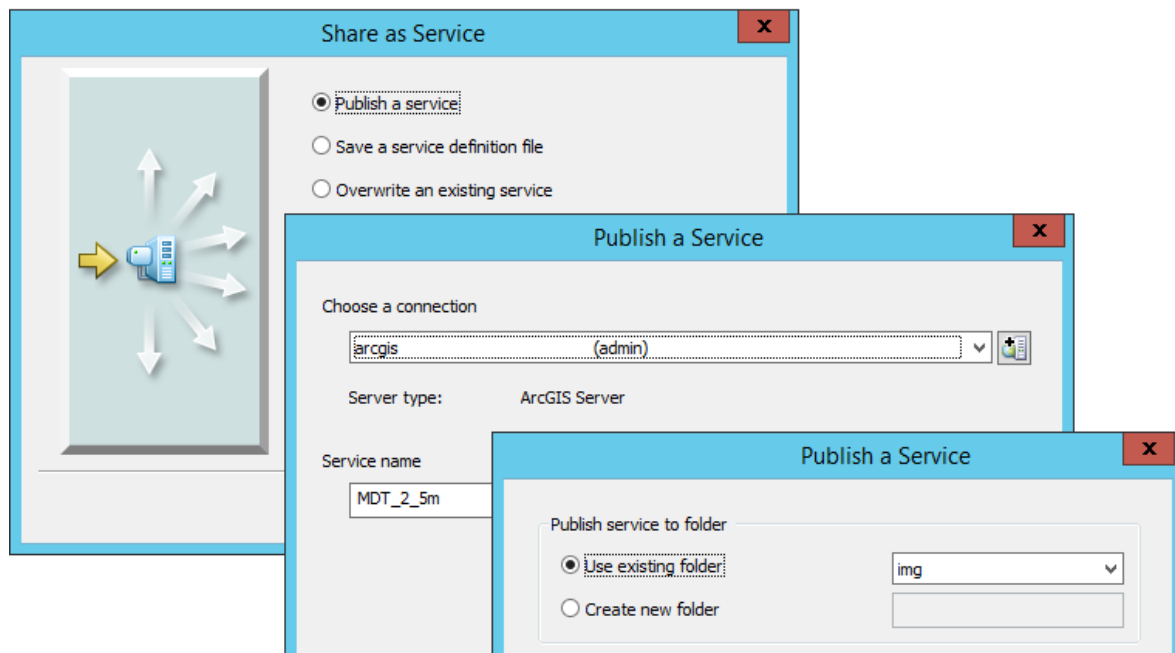


Fig. 34. Asistente de publicación del servicio de imágenes

En el editor de servicio, en parámetros – función (Function), al dar clic en el botón gestionar (1 manage...) se despliega el gestor de plantillas de funciones raster, mediante el boton 2, se selecciona el archivo RFT que fue creado en el editor de plantillas de funciones ilustrado en la figura 29, que se define como la plantilla por defecto 3, y por último se procede a publicar el servicio.

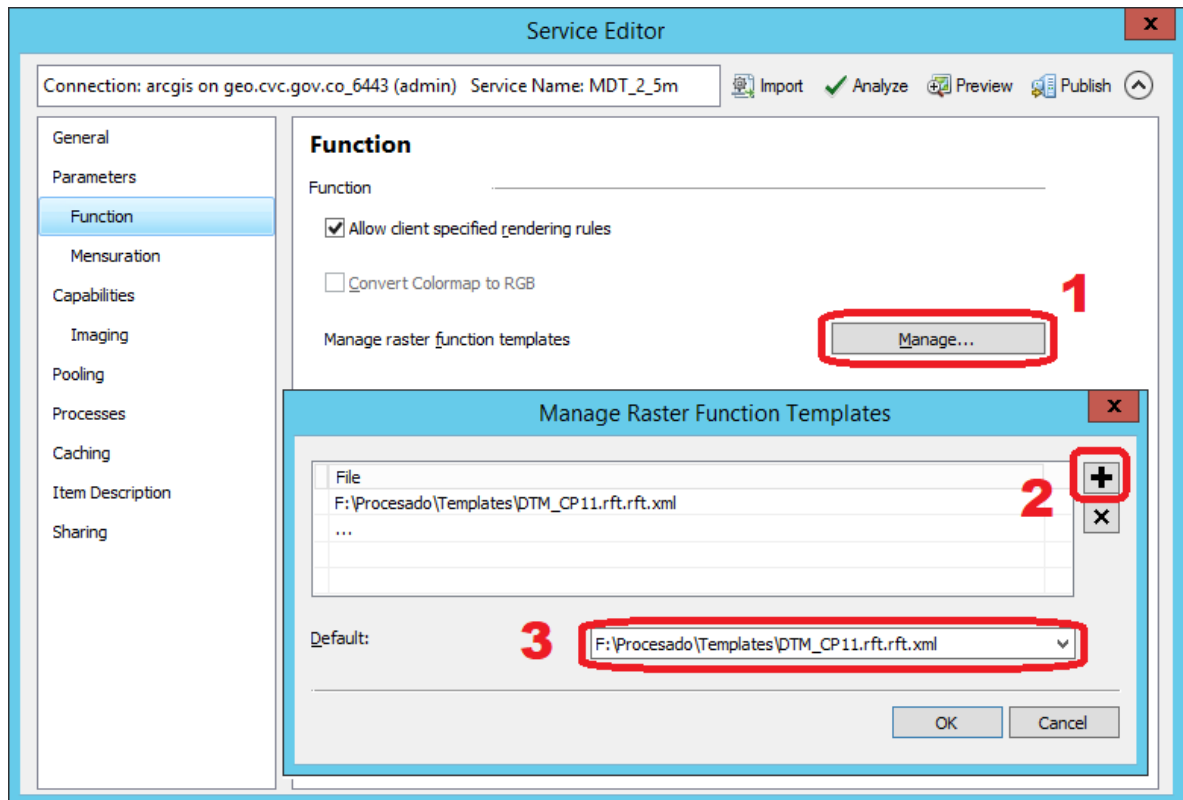


Fig. 35. Implementación de plantilla en servicio

Con este procedimiento queda simbolizado el servicio y su apariencia sería la siguiente.

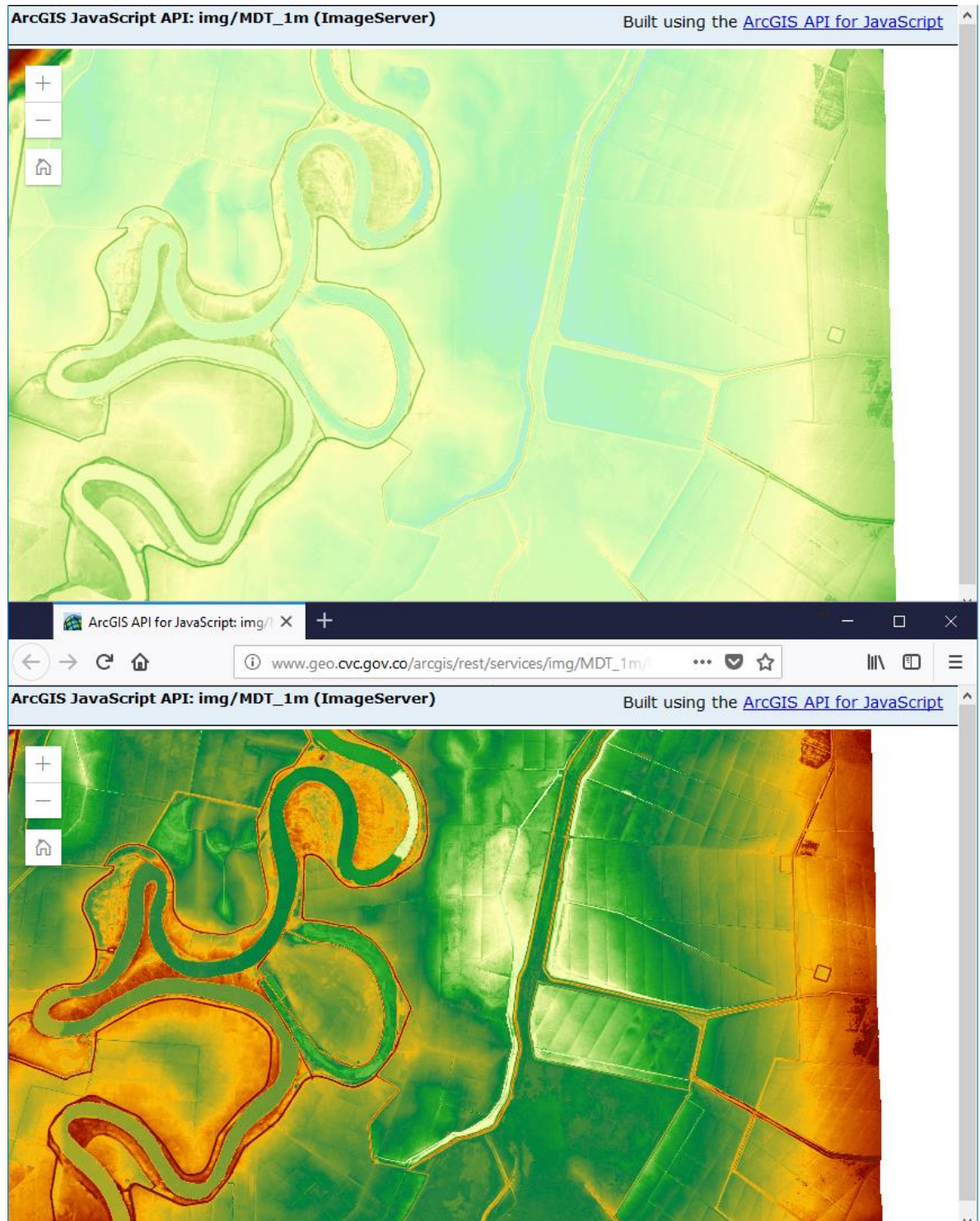


Fig. 36. Servicio MDT corredor Río Cauca, simbología dinámica en base a rango de elevaciones

5.6.2 Simbología de los modelos derivados

La simbología de los servicios de modelos derivados del MDT es el estándar. El procedimiento para publicar el servicio es el siguiente: se crea la plantilla correspondiente, solo aplicando la función de pendiente, aspecto o modelo de sombras; luego se publica el servicio con el nombre del modelo derivado pero empleando como fuente el MDT original y por último es aplicada la plantilla correspondiente, esto evita tener que generar los archivos a cada uno de estos modelos derivados y ahorra almacenamiento en disco.

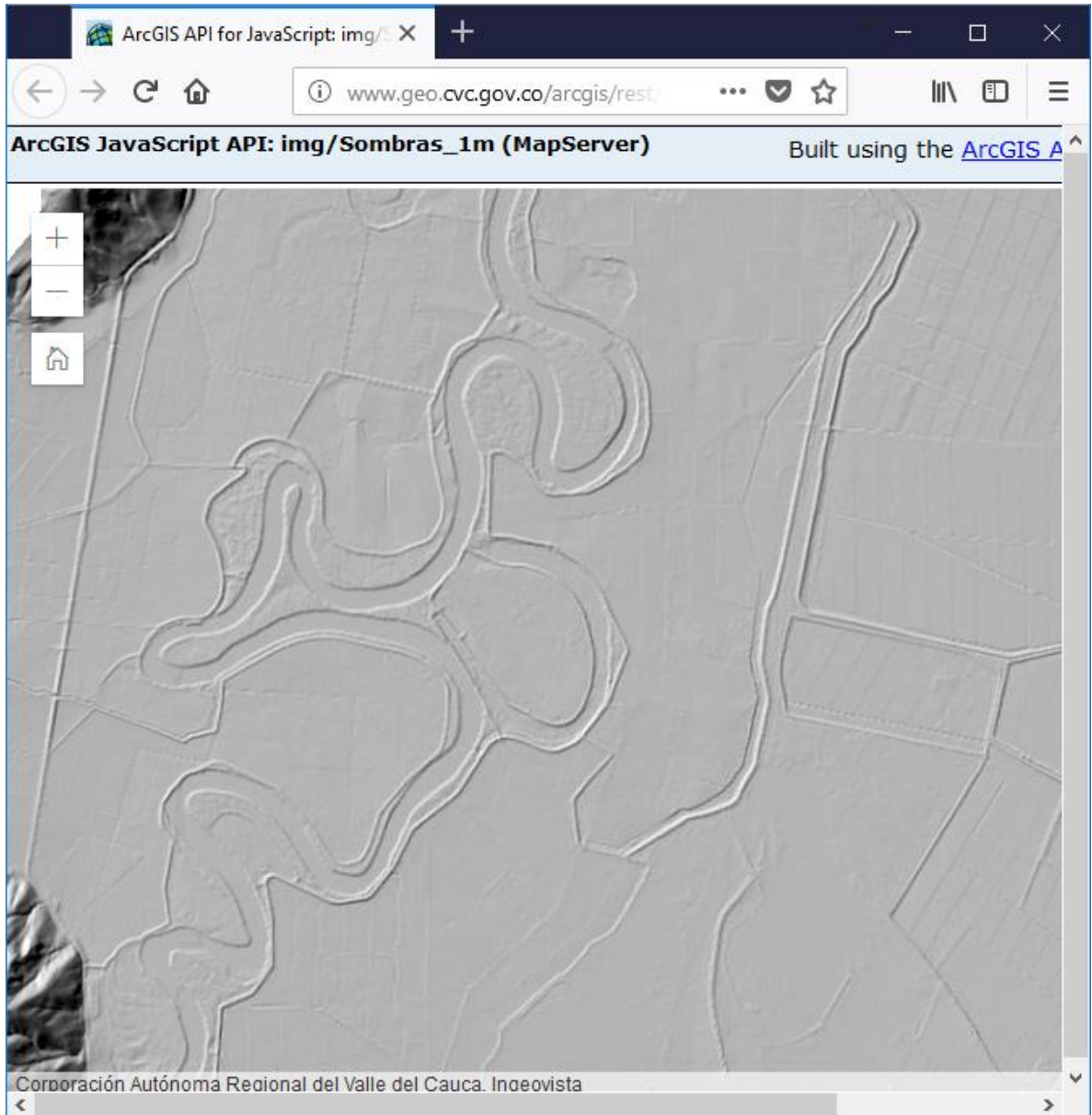


Fig. 37. Servicio de sombras del corredor del Río Cauca

5.6.3 Simbología del servicio de fotografías aéreas

Como se mencionaba anteriormente, las fotografías aéreas no requieren simbología, entonces no se hace necesario construir una plantilla, simplemente se realiza la publicación del servicio desde el mosaico dataset, empleando el mismo procedimiento descrito para un MDT, omitiendo la creación y aplicación de la plantilla.



Fig. 38. Servicio de fotografías aéreas del corredor del Río Cauca

5.7 MODELO DE GEOPROCESAMIENTO

El servicio de geoprocesamiento ejecuta un script Python creado por Esri y obtenido de la página web (Noman, 2014). El script fue modificado para trabajar con los MDTs de radar y LiDAR. El proceso de implementación fue el siguiente:

5.7.1 Preparación de los datos

Para el correcto funcionamiento del script se requiere dos GDB una denominada *dembnd* la cual almacenará las clases de entidad vectoriales *Demboundary* y *ProfileSchema* descritas en las tablas 4 y 5 del punto 5.4.4 y otra denominada *demdata* con los MDTs a partir de los cuales se obtendrán los perfiles topográficos.

5.7.2 Modificación del Script

Se modifica el script según el manual contenido en el paquete descargado al inicio de este proceso, el cual se puede observar en el Anexo 2

5.7.3 Publicación del servicio

Para la publicación del servicio es necesario ejecutarlo al menos una vez. La ejecución se realiza desde la aplicación ArcMap, los resultados en la ejecución del Script aparecen en la ventana de resultados, la cual puede llamarse desde el menú geoprocesamiento. A continuación, el proceso:

- Abrir el documento de mapa Profile.mxd se deben activar las extensiones 3D y spatial, se localiza el script y se ejecuta.
- Se abre la ventana de resultados (menú geoprocesamiento).
- Mediante el menú contextual se procede a la publicación del servicio como se puede observar en la figura 39.

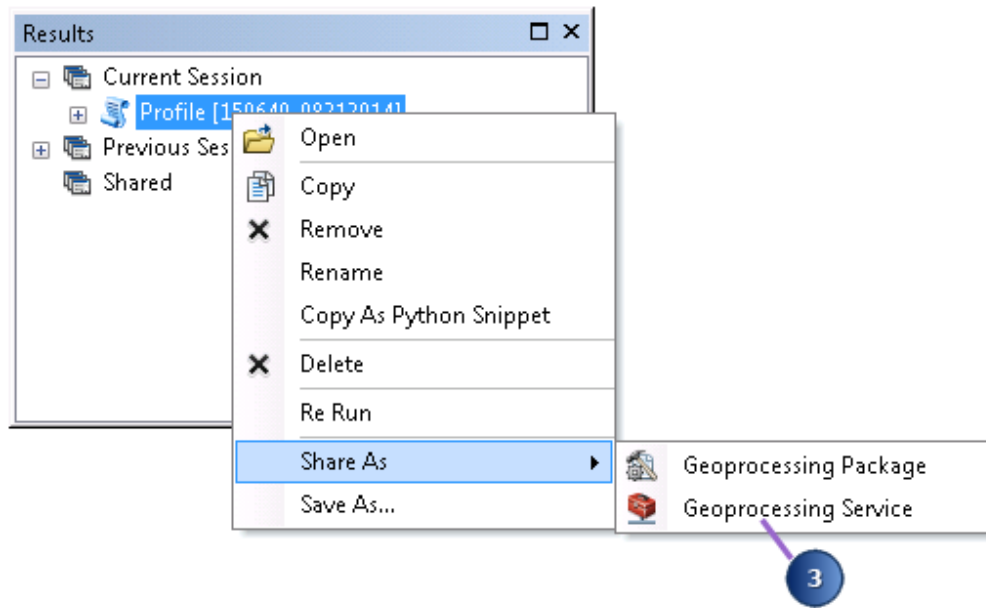


Fig. 39. Inicio del proceso de publicación servicio geoprocamiento

- Se sigue el asistente para nombre del servicio y conexión al servidor. Al final en el editor de servicio se deben determinar las siguientes propiedades como se muestra en la figura 40.

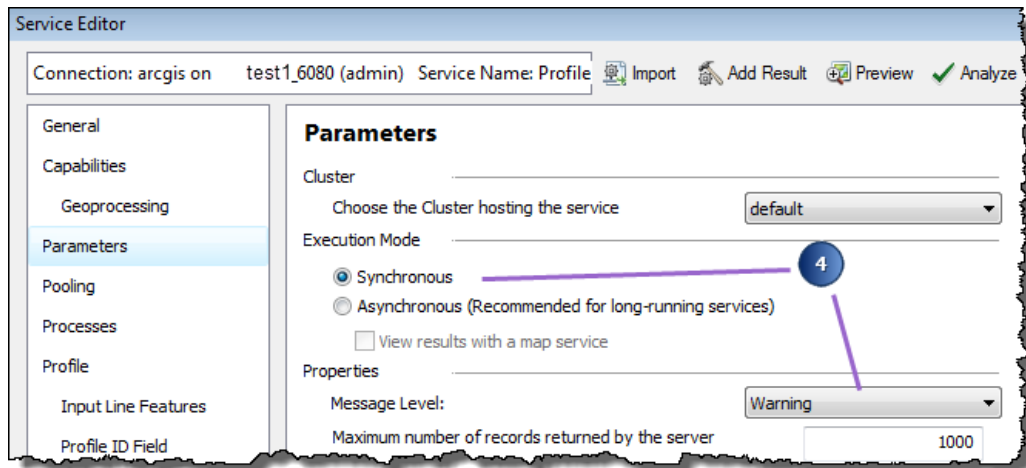


Fig. 40. Parámetros que requiere el servicio tomado de Esri

6. RESULTADOS

Los requerimientos son todas aquellas características y funcionalidades que un usuario desea disponer dentro de una aplicación o programa. Para atender las expectativas de los diferentes grupos de trabajo que posee la CVC se realizó un levantamiento de requerimientos identificando a tres grupos de trabajo que hacen parte del universo de posibles usuarios del visor geográfico.

Los grupos participantes fueron SIA (Sistema de Información ambiental), Grupo de usuarios no expertos de CVC y Grupo de administradores de cartografía de CVC, se les pidió que llenaran la encuesta consignada en el ANEXO C indicando en ella, el nombre, cargo, grupo al que pertenecía y seleccionando las funcionalidades que quisiera encontrar en el visor geográfico.

Una vez terminado el proceso de levantamiento de requerimientos se procede a generar la depuración de la información para su análisis. La figura 41, muestra un ejemplo del análisis de requerimientos de los usuarios en los tres grupos.

No.	Requerimientos de Usuario sobre la información Raster	Usuario			Dispositivo		Tipo de Requerimiento		
		Experto	Interno	Externo	PC	Móvil	Funcional	No Funcional	Organización
1	Consulta espacial	X	X	X	X	X	X		
2	Consulta por atributos raster	X	X	X	X	X	X		
3	Servicio para generar pendientes a partir de un DEM y exportar	X	X	X	X		X		
4	El Servicio de adición a un software SIG	X	X	X	X	X	X		
5	Estadísticas de los servicios raster	X	X		X	X	X		
6	Adicionar capas temáticas	X	X	X	X		X		
7	Generar salidas temáticas y su imagen.	X			X		X		
8	Sólo se pueden descargar o vender información propia de CVC							X	
9	La información deberá protegerse a los diferentes niveles de procesos y usuarios.							X	
10	Integración del desarrollo raster en el actual GeoCVC								X
11	Actualización del Sistema a nivel de información raster								X

Fig. 41. Requerimientos obtenidos

Las aerofotografías y los ORIs de radar se recopilaron y estructuraron en carpetas en el servidor, el cual cuenta con una partición con almacenamiento de 5Tb. Los datos se encontraban en formato ECW y fue necesario cambiarlos al formato JP2 debido a que la extensión image de ArcGIS no cuenta con la licencia para la lectura de dicho formato. Se eligió el formato JP2 debido a que posee un buen radio de compresión, similar al de ECW; la organización de los datos se puede apreciar en la figura 42.

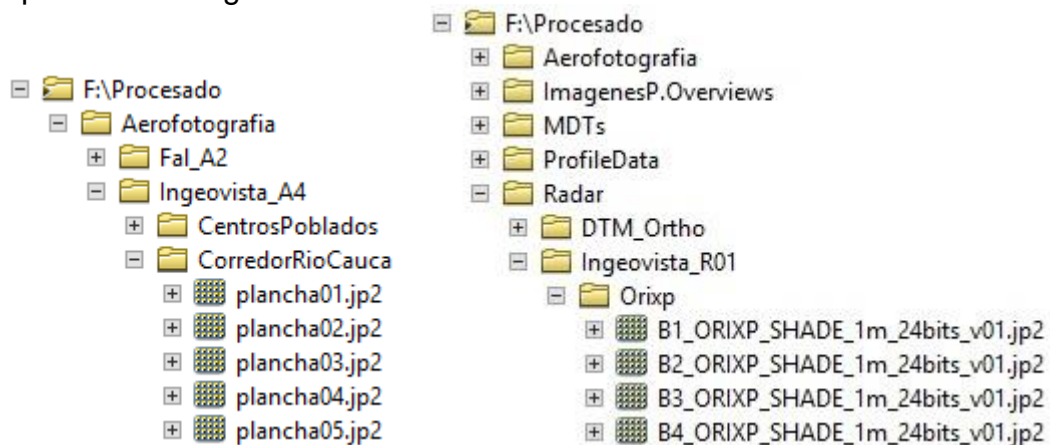


Fig. 42. Estructura de carpetas en el servidor

Se crea una carpeta en el servidor denominada *Profiledata* y se modifican los permisos para que pueda acceder el usuario que ejecuta el servicio de ArcGIS Server; se crean las GDB descritas en el punto 5.7.1; los MDTs son almacenados en la GDB *demdata*, la cual contiene un *mosaic dataset* con los modelos de resolución de 1m (corredor y centros poblados) y dos raster dataset, el MDT de radar de 2,5m y el MDT de 30m SRTM. Como se puede apreciar en la figura 43

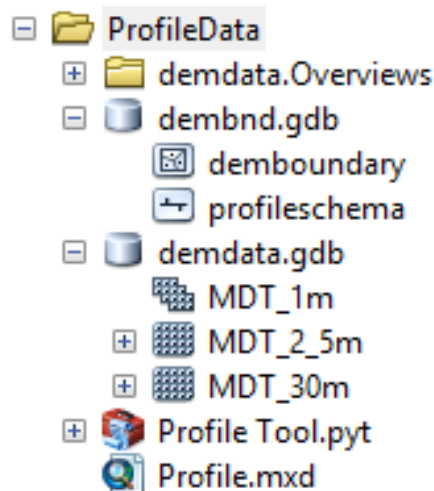


Fig. 43. Estructura de los datos que emplea el servicio

Se crearon los servicios dinámicos con la simbología descrita en el numeral 5.6.1 a partir de la estructura de datos de los MDTs que se presentó en la figura 43. obteniendo dos servicios con simbología dinámica de los MDT de 1m y 2,5m. Adicional se publicó un servicio sin la simbología dinámica para propósitos de análisis y obtención de modelos derivados de MDTs desde los equipos desktop que cuentan con licencia de ArcGIS desktop; se publican los modelos derivados de sombras y pendiente a partir de los DTMs originales MDTs de 1 y 2.5m mediante la utilización de plantillas dando como resultado 4 servicios más. Los servicios obtenidos se pueden apreciar en la figura 44.

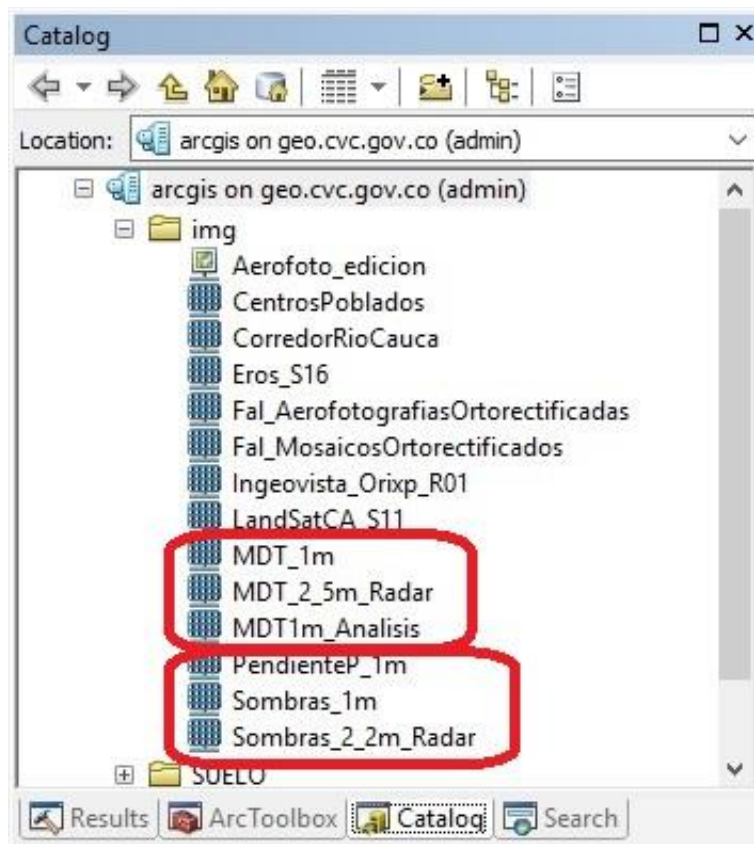


Fig. 44. Carpeta con los principales servicios de visor

Se logra una eficiencia en el almacenamiento al generar varios servicios a partir de un único MDT, como se observa en la figura 45.

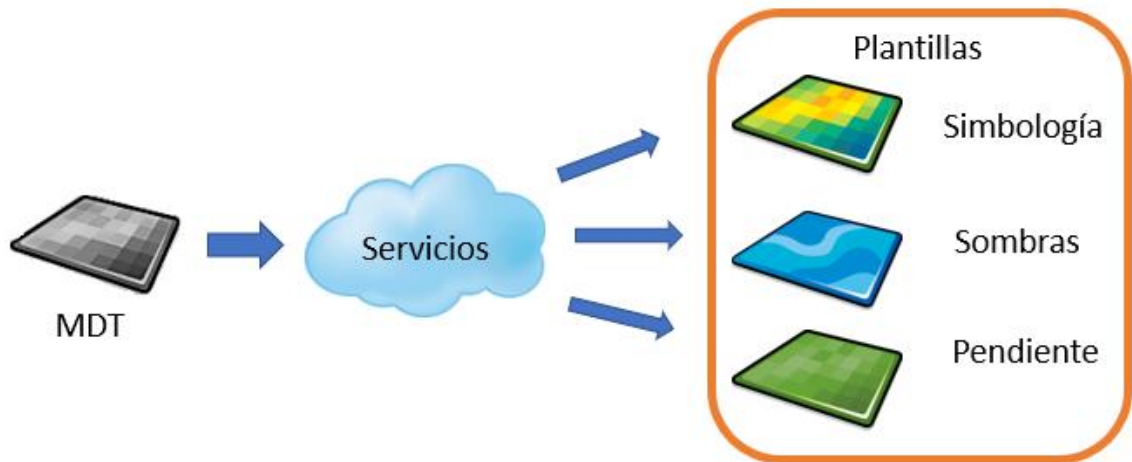


Fig. 45. Servicios de modelos derivados a partir de una fuente MDT única

Adicionalmente se crearon los servicios geográficos de imágenes, uno para el despliegue de las aerofotografías de los centros poblados, otro para el corredor del Río Cauca y un tercer servicio para la ORI a partir de sus correspondientes *mosaic dataset*, mediante el proceso descrito en el numeral 5.6.3. En la figura 52 se puede apreciar la GDB que contiene a las fuentes de datos descritas.

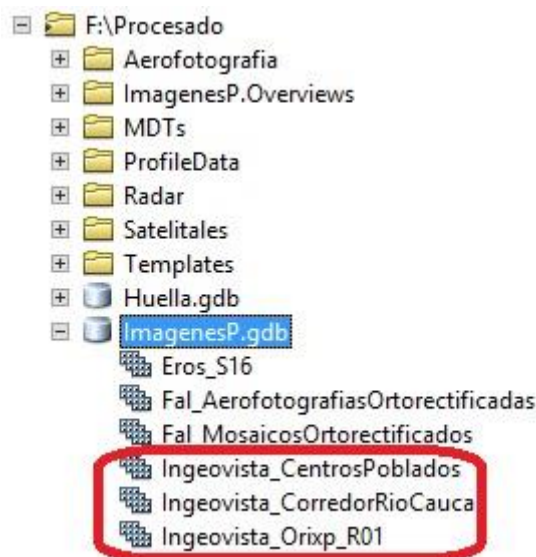


Fig. 46. Mosaic dataset de aerofotografías y Oris

Los servicios se obtuvieron siguiendo el procedimiento en el punto 5.6.3 en la figura 46 se pueden observar los servicios obtenidos.

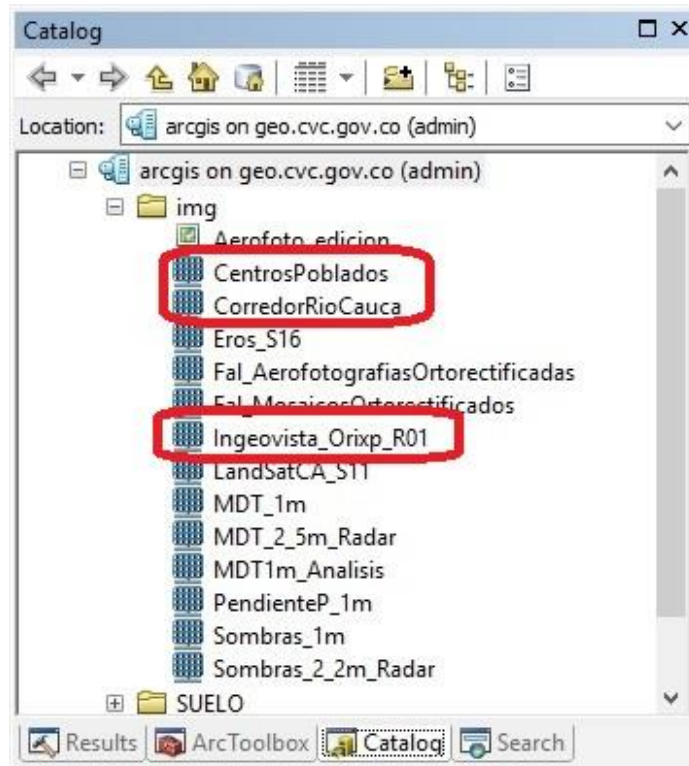


Fig. 47. Servicios de imágenes de aerofotografías y Oris

Se implementó el servicio geoprocresamiento que permite la creación de perfiles topográficos, el cual fue descrito en el numeral 5.7 del modelo de geoprocresamiento, el cual es utilizado por el visor y también puede ser utilizado desde ArcMap mediante el addin de perfil.



Fig. 48. Servicios de imágenes de aerofotografías y Oris

El visor geográfico se construyó empleando HTML y javascript y cuenta con los siguientes elementos: **1** panel lateral izquierdo donde se encuentra la tabla de contenido carga de shapes, herramientas de dibujo y ubicación por coordenadas, **2** una barra de herramientas en la parte superior, **3** Área de visualización geográfica, **4** herramienta de perfil y **5** Ventana de perfil topográfico.

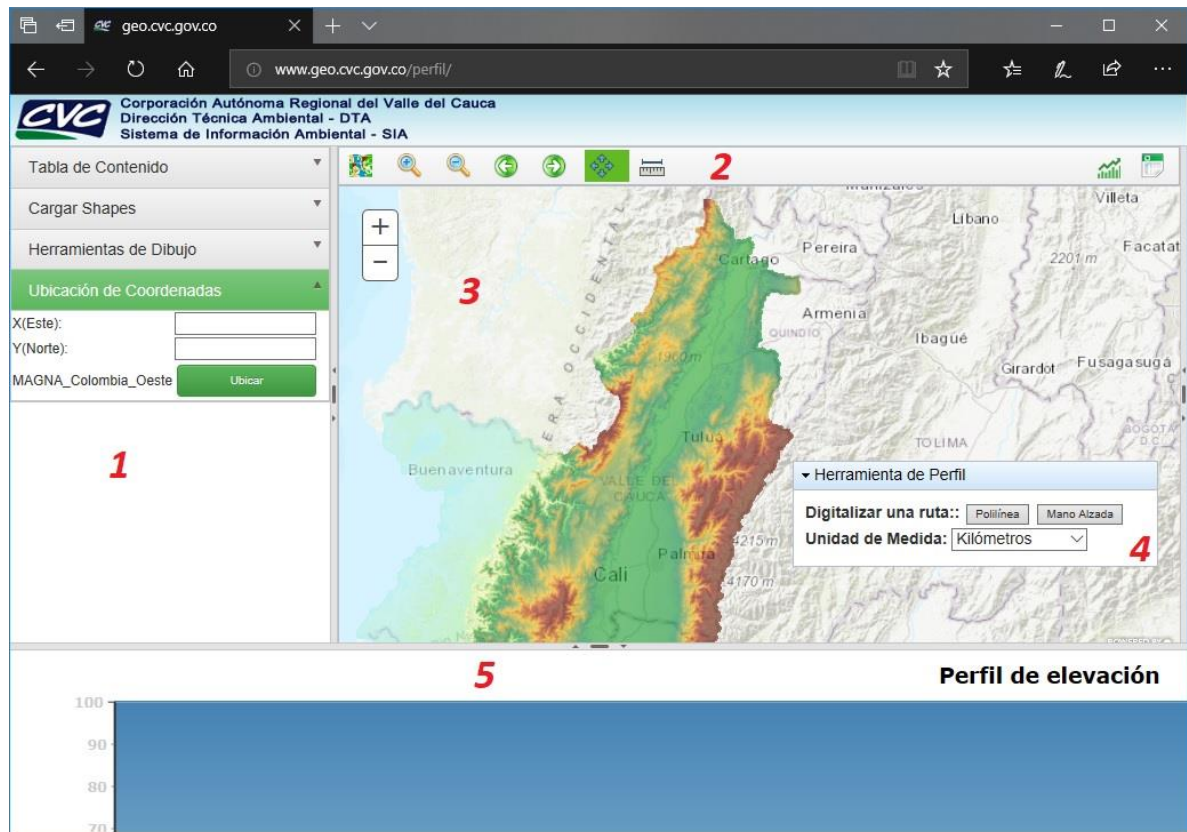


Fig. 49. Elementos del visor perfil

El panel lateral izquierdo está compuesto de la tabla de contenido, el cual permite apagar o encender los servicios geográficos, también cuenta con la opción de ajustar la transparencia del servicio; el panel de carga de shapes permite la carga de un shape file a visor. El panel de herramientas de dibujo permite realizar bosquejos en el área de visualización geográfica; el panel de Ubicación por coordenadas permite ingresar coordenadas planas y ubicarlas en el área de visualización.

La descripción de los botones de izquierda a derecha de la barra de herramientas en la parte superior del visor es la siguiente: los primeros 5 botones corresponden a la extensión completa, acercamiento, alejamiento, extensión geográfica anterior o siguiente realizadas durante la sección, desplazamiento (pan), herramienta de medición; los dos botones de la derecha funcionan una vez es generado el perfil

topográfico y corresponden a la herramienta de exportar perfil a imagen y exportar los datos del perfil abscisas y cotas a un archivo Excel.



Fig. 50. Barra de Herramientas

La herramienta de generación perfiles topográficos funciona a partir de un alineamiento de dos o más vértices, el cual debe ser dibujado interactivamente por el usuario sobre los modelos de elevación contenidos en la base de datos demdata.

Los pasos para generar un perfil topográfico se presentan a continuación:

- Se seleccionan las unidades en las que será dibujado el perfil topográfico.



Fig. 51. Herramienta de perfil

- Se selecciona el modo a digitalizar, polilínea (dibujar como mínimo dos vértices) o mano alzada; (Clic sostenido y mover el cursor).

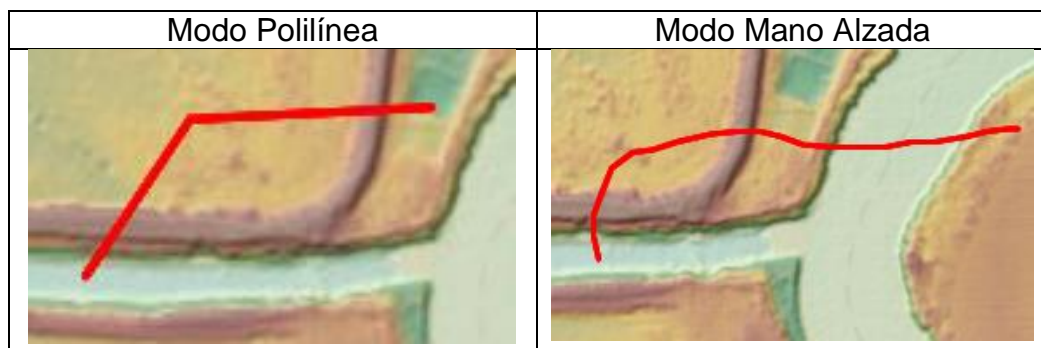


Fig. 52. Digitalización polilínea y mano alzada

- El perfil se puede exportar como un gráfico y como un archivo de Excel con estos botones.



Fig. 53. Botones de exportación

- Los archivos quedan almacenados en la carpeta de descargas en formato PNG y XLS.



Fig. 54. Perfil exportado en formato PNG

PuntoID	Abscisa (km)	Cota (mts)
0	0,00	959,794
1	11,116	960,132
2	22,232	959,708
3	33,348	959,886
4	44,464	960,025
5	55,58	959,887
6	66,696	959,979
7	77,812	959,999
8	88,928	960,014
9	100,044	960,109
10	111,159	960,097
11	122,276	960,002
12	133,391	959,724
13	144,508	959,636
14	155,623	959,803
15	166,739	960,423
16	177,855	960,477
17	188,971	960,35
18	200,087	960,441

Fig. 55. Datos del perfil exportados en formato XLS

7. CONCLUSIONES

Luego del desarrollo del proyecto se llegó a las siguientes conclusiones

- Se consolidaron los datos de forma centralizada, lo cual evita duplicidad de información o posible modificación de los datos de forma accidental, mediante la utilización de las plantillas en los servicios no se hace necesario crear los modelos derivados ahorrando espacio en disco.
- Los servicios de visualización dinámica permiten apreciar los cambios de las alturas en el área de visualización, esto es muy conveniente en las zonas planas, pues se pueden apreciar diques y terraplenes.
- Los servicios geográficos son consumidos por la aplicación web y pueden ser cargados en el software ArcGIS para su visualización y análisis, esto brinda versatilidad en el uso de los datos ráster.
- Al igual que los servicios de imágenes, el servicio de geoprocésamiento es consumido por la aplicación web y es posible utilizarlo en desde el software ArcGIS.
- Los servicios son más eficientes y se despliegan más rápido cuando se publican a partir de los datos ráster almacenados en carpetas en el propio servidor de imágenes, el desempeño es pobre si se cargan a la GDB corporativa Oracle, además conlleva a un problema de almacenamiento debido a que la totalidad de datos ráster superaría el espacio de almacenamiento de la base de los datos
- La aplicación geográfica cumple su cometido de brindar acceso a los datos tanto a funcionarios como al público en general, adicional a esto es posible construir un perfil topográfico a partir de los MDTs.

8. RECOMENDACIONES

- Se recomienda adquirir el licenciamiento del formato ECW para el software ArcGIS Server, para poderlo utilizar directamente sin necesidad de realizar conversiones a otros formatos.
- Se recomienda continuar con el uso de las plantillas para los modelos derivados y evitar creación de los ráster de estos modelos, también se recomienda cachear los servicios que no requieren visualización dinámica como el modelo de sombras o el de aspecto.
- Se recomienda implementar políticas de seguridad en los servicios geográficos.
- La Oficina de Tecnología de la Información debe ser la encargada de gestionar las políticas de software, sus respectivos certificados de firma digital de los programas con los que se realizan los portales geográficos.

BIBLIOGRAFÍA

- Carlos Gómez, J. &. (2008). Cartografía geomorfológica aplicada a un sector de interés arqueológico en el Macizo del Deseado, Santa Cruz (Patagonia Argentina). *Investigaciones Geográficas UNAM*.
- CARTO DB. (s.f.). *carto*. (carto) Recuperado el 18 de Octubre de 2018, de <https://carto.com/>
- Congreso de la Republica de Colombia. (6 de marzo de 2014). *LEY 1712 DE 2014*. (Secretaría del Senado) Recuperado el 18 de septiembre de 2018, de http://www.secretariassenado.gov.co/senado/basedoc/ley_1712_2014.html
- Congreso de la Republica de Colombia. (15 de octubre de 2018). *LEY 99 DE 1993*. (Secretaría del Senado) Recuperado el 20 de octubre de 2018, de http://www.secretariassenado.gov.co/senado/basedoc/ley_0099_1993.html
- Corporación Autónoma Regional del Valle del Cauca. (10 de mayo de 2018). *Acerca de la CVC*. (CVC) Recuperado el 20 de octubre de 2018, de <http://www.cvc.gov.co/acerca-de-cvc>
- Environmental Systems Research Institute. (2015). *The Arcgis Book*. Redlands, California: Esri Press.
- Environmental Systems Research Institute. (2016). *¿Qué es un dataset de mosaico?* (esri inc.) Recuperado el 20 de octubre de 2018, de <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/raster-and-images/what-is-a-mosaic-dataset.htm>
- Environmental Systems Research Institute. (2016). *¿Qué es una superficie TIN?* (esri inc.) Recuperado el 20 de Octubre de 2018, de <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/tin/fundamentals-of-tin-surfaces.htm>
- Environmental Systems Research Institute. (2016). *¿Qué son los datos LIDAR?* (esri) Recuperado el 20 de octubre de 2018, de <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/las-dataset/what-is-lidar-data-.htm>
- Environmental Systems Research Institute. (2016). *Cómo funciona Curvas de nivel*. (esri inc) Recuperado el 20 de Octubre de 2018, de <http://desktop.arcgis.com/es/arcmap/10.3/tools/spatial-analyst-toolbox/how-contouring-works.htm>
- Environmental Systems Research Institute. (2018). *¿Qué es ArcGIS Image Server?* (esri inc) Recuperado el 23 de octubre de 2018, de <http://enterprise.arcgis.com/es/server/latest/get-started/windows/what-is-arcgis-image-server-.htm>
- F, J. D. (1978). *Digital Terrain Model an Overview*.
- Felicitimo, A. M. (2004). *Pagina personal Angel M. Felicitimo*. Recuperado el 20 de Octubre de 2018, de <http://www6.uniovi.es/~feli/pdf/libromdt.pdf>
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. (University of California, Irvine) Recuperado el 12

- de octubre de 2018, de <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- Geoimage. (2018). *DEMS Overview*. (Geoimage) Recuperado el 20 de Octubre de 2018, de <https://www.geoimage.com.au/DEMS/dems-overview>
- Geoprocess SAS. (2018). *GISdata*. (Geoprocess SAS) Recuperado el 19 de Octubre de 2018, de <https://www.gisdata.co>
- Geovisor IDESC. (3 de Julio de 2018). (Alcaldia de Santiago de Cali) Recuperado el 18 de Octubre de 2018, de <http://idesc.cali.gov.co/geovisor.php>
- Inegi. (2018). *Instituto Nacional de Estadística y Geografía*. (INEGI) Recuperado el 19 de Octubre de 2018, de <http://www.beta.inegi.org.mx/default.html>
- Instituto Geográfico Nacional de Argentina. (s.f.). *IGN*. (Ministerio de la Defensa Argentino) Recuperado el 12 de octubre de 2018, de <http://www.ign.gob.ar/sig>
- Mantelo, R. D.-F.-J. (1979). *Tratado de topografía*. Madrid: Aguilar ediciones S.A.
- Noman, N. (26 de agosto de 2014). *Setting up an on-premise elevation Profile service*. Recuperado el 5 de octubre de 2018, de <https://www.arcgis.com/home/item.html?id=0d47b17d3b884b9d977a6f76fc6bfcaa>
- Observatorio de Territorios Etnicos y Campesinos. (s.f.). (Pontificia Universidad Javeriana) Recuperado el 17 de Octubre de 2018, de <http://etnoterritorios.org/sig.shtml>
- Olaya, V. (2014). Modelo Ráster. En *Sistemas de Información Geográfica* (pág. 88). Madrid: Creative Common Attribution.
- Python Software Foundation. (12 de octubre de 2018). *General Python FAQ*. (Python Software Foundation) Recuperado el 20 de octubre de 2018, de <https://docs.python.org/2.7/faq/general.html#why-was-python-created-in-the-first-place>
- Sun microsystems. (2018). (Sun Microsystem) Recuperado el 21 de Octubre de 2018, de <https://www.oracle.com/java/>
- Tait, M. G. (2005). Implementing geoportals: applications of distributed GIS. *Computers, Environment and Urban Systems*, 29, 33-47.
- Tomlin, C. D. (1990). *Geographic information systems and cartographic modelling*. Prentice Hall.
- W3C World Wide Web Consortium. (11 de octubre de 2016). *Extensible Markup Language (XML)*. (W3C) Recuperado el 12 de octubre de 2108, de <https://www.w3.org/XML/>
- ZENIT. (2018). *ZENIT Topografía y Cartografía*. (zenit) Recuperado el 20 de Octubre de 2018, de <https://zenitop.es/malla-modelo-digital-terreno-lidar/>

ANEXO A PASOS PARA MODIFICAR SCRIPT PYTHON PERFIL

A continuación, se citan textualmente los pasos contenidos en el manual (Noman, 2014) para el funcionamiento del Script en el servidor de la CVC.

1. DEM layer variables

- A. Create a string variable for each DEM layer. Assign the DEM layer names (as defined in Profile.mxd) to the string variables. The following is an example for three DEM layers. If you only have one DEM, then only one variable is needed.

```
#-----  
# Define layer variables  
#-----  
demLayer1 = "dem90m"  
demLayer2 = "dem30m"  
demLayer3 = "dem10m"  
#-----
```

- B. Wrap each layer variable in an *arcpy.Describe* statement. These statements are never executed, but they are mandatory for the packager to recognize the layers during publishing. By default, there are three *arcpy.Describe* statements, remove the redundant ones if you have less than three DEM layers, or add new ones if you have more than three.

```
#-----  
# Wrap each variable in an arcpy.Describe statement  
#-----  
if False:  
    arcpy.Describe(demLayer1)  
    arcpy.Describe(demLayer2)  
    arcpy.Describe(demLayer3)  
#-----
```

- C. Update the dictionary '*self.dictDEMs*'. This dictionary designates the DEM layer for each resolution. Replace the key-value pairs in this dictionary with your DEM resolutions and the layer variables. The DEM resolutions should be specified as integer values in meters.

```
#-----  
# Update the DEM layers dictionary  
#-----  
self.dictDEMs = {"90":demLayer1,  
                 "30":demLayer2,  
                 "10":demLayer3}  
#-----
```

2. DEM resolutions and the default resolution

Update the dictionary '*self.dictDEMResolutions*'. This dictionary provides the descriptions (the keys) for all the DEM resolutions. The keys will show up in the dropdown of the DEM resolution parameter on the tool dialog. Replace the key-value pairs in this dictionary with your DEM descriptions and resolutions. The DEM resolutions should be specified as integer values in the DEM XY units.

Update the variable 'self.defaultDEMResolution'. This variable defines the default resolution to use when the user omits the DEM resolution parameter.

```
#-----  
# Resolution Dictionary  
#-----  
self.dictDEMResolutions = {"90m":"90", "30m":"30", "10m":"10"}  
self.defaultDEMResolution = '90'  
#-----
```

3. Profile schema feature class

Update the variable 'profileSchm1'. Replace the pathname with your profile schema feature class.

```
#-----  
# Input profile schema  
#-----  
profileSchm1 = r"C:\Profile\ProfileData\dembnd.gdb\profileschema"  
self.profileSchema = profileSchm1  
#-----
```

4. DEM boundary feature layer

Update the variable 'boundaryLayer1'. Replace this string with the DEM boundary layer name you created in the Profile.mxd.

```
#-----  
# DEM boundary layer  
#-----  
boundaryLayer1 = "demboundary"  
if False:  
    arcpy.Describe(boundaryLayer1)  
self.demBoundary = boundaryLayer1  
#-----
```

5. Maximum number of elevation values for each profile

Optionally, you may change the maximum number of elevation values that can be returned for each profile. This number is defined in the variable 'self.maxNumVertices'. The default value is 2000. You can update it to a higher value if necessary.

```
#-----  
# Maximum number of vertices  
#-----  
self.maxNumVertices = 2000  
#-----
```

ANEXO B SCRIPT PERFIL

```
""" Tool name: Profile
Source name: Profile Tool.pyt
Description: Return an elevation profile for an input polyline.
Author: Environmental Systems Research Institute Inc.
Last updated: Aug. 05, 2014
"""

import os
import time
import arcpy

class Toolbox(object):
    def __init__(self):
        """Define the toolbox (the name of the toolbox is the name of the
        .pyt file)."""
        self.label = "Profile Tool"
        self.alias = ""

        # List of tool classes associated with this toolbox
        self.tools = [Profile]

class Profile(object):
    def __init__(self):
        """Define the tool (tool name is the name of the class)."""
        self.label = "Profile"
        self.description = "Return an elevation profile for an input polyline."
        self.canRunInBackground = False
        # custom properties
        self.debug = True
        self.outputToTable = False # set to True to direct the output to a table.
        self.idFieldName = "ID"
        self.glen_field1 = "proflen0"
        self.glen_field2 = "proflen1"
        self.metadataFieldName = "DEMResolution"
        self.geodesicLenFieldName = "ProfileLength"
        self.listLinearUnits = ["Meters", "Kilometers", "Feet", "Yards", "Miles"]
        #-----
        # Maximum number of vertices
        #-----
        self.maxNumVertices = 2000
        #-----
        # DEM boundary layer
        #-----
        boundaryLayer1 = "demboundary"
```

```

if False:
    arcpy.Describe(boundaryLayer1)
self.demBoundary = boundaryLayer1
#-----
# Profile schema feature class
#-----
profileSchm1 = r"C:\Profile\ProfileData\dembnd.gdb\profileschema"
self.profileSchema = profileSchm1
#-----
# DEM resolution dictionary
#-----
self.dictDEMResolutions = {"90m": "90", "30m": "30", "10m": "10"}
self.defaultDEMResolution = '90'
#-----
# DEM data layers
#-----
# Define layer variables
#-----
demLayer1 = "dem90m"
demLayer2 = "dem30m"
demLayer3 = "dem10m"
#-----
# Wrap each variable in an arcpy.Describe statement
#-----
if False:
    arcpy.Describe(demLayer1)
    arcpy.Describe(demLayer2)
    arcpy.Describe(demLayer3)
#-----
# Update the DEM layers dictionary
#-----
self.dictDEMs = {"90": demLayer1,
                 "30": demLayer2,
                 "10": demLayer3}
#-----
# DEM coordinate system
demSR = arcpy.Describe(self.dictDEMs.values()[0]).spatialReference
self.demCoordinateSystem = demSR
# DEM linear unit
lun = demSR.linearUnitName
if lun == "" or lun == None:
    lun = demSR.angularUnitName
    if 'degree' in lun.lower():
        lun = 'decimaldegrees'
if 'foot' in lun.lower() or 'feet' in lun.lower():

```

```

    lun = 'feet'
    self.demLinearUnit = lun
    # for adjusting length, change the zf here. eg, if the DEM linear unit is feet,
    then zf = 0.3048.
    # for meter, zf = 1.0; for decimal degrees, use zf = 1.0
    self.zFactor = self.getUnitConversionFactor(self.demLinearUnit)

    self.errorMessages = ["No input polyline features specified. The input needs
    to have at list one line feature.",
        "Input resolution is not supported. Select a different DEM source.",
        "The input profile line you requested falls outside of the data currently
    available in this service.",
        "Input parameter {0} is not valid.",
        "The input polyline contains too many vertices. Reduce the number of
    vertices.",
        "The specified sample distance results in more vertices than allowed.
    Increase sampling distance.",
        "Input feature contains too many vertices or the sample distance is too
    small. Specify a line with less than 1024 vertices, or increase the sampling
    distance.",
        "Input sample distance cannot be 0 or negative.",
        "Input feature id field does not exist. Change to another field or leave it
    as default.",
        "The number of input profile lines exceeds limit. Reduce the number of
    input profile lines to not more than 10."]

    def getLayerName(self, res):
        if not res in self.dictDEMs.keys():
            arcpy.AddError(self.errorMessages[1])
            raise
            return
        return self.dictDEMs[res]

    def getUnitConversionFactor(self, u1): # get conversion factor
        uFactor = 1
        inUnit = u1.strip().lower()
        if inUnit in ["meters", "meter"]:
            uFactor = 1
        if inUnit in ["centimeters", "centimeter"]:
            uFactor = 0.01
        if inUnit in ["decimaldegrees", "decimaldegree"]:
            uFactor = 1
        if inUnit in ["decimeters", "decimeter"]:
            uFactor = 0.1
        if inUnit in ["feet", "foot"]:

```

```

        uFactor = 0.3048
    if inUnit in ["foot_us", "feet_us"]:
        uFactor = 0.3048006096012192
    if inUnit in ["inches", "inch"]:
        uFactor = 0.0254
    if inUnit in ["kilometers", "kilometer"]:
        uFactor = 1000
    if inUnit in ["miles", "mile"]:
        uFactor = 1609.344
    if inUnit in ["millimeters", "millimeter"]:
        uFactor = 0.001
    if inUnit in ["nauticalmiles", "nauticalmile"]:
        uFactor = 1852
    if inUnit in ["points", "point"]:
        uFactor = 0.000352777778
    if inUnit in ["unknown", ""]:
        uFactor = 1
    if inUnit in ["yards", "yard"]:
        uFactor = 0.91440
    return uFactor

def lineFootprintTest(self, in_line_features):
    # Footprint polygon
    footPrt = self.demBoundary
    resList = []
    footPrtLayer = 'aFootPrtLyr'
    arcpy.MakeFeatureLayer_management(footPrt, footPrtLayer)
    arcpy.SelectLayerByLocation_management(footPrtLayer,
"COMPLETELY_CONTAINS",
                                         in_line_features)
    self.traceExecutionTime("")
    with arcpy.da.SearchCursor(footPrtLayer, "res") as cursor:
        for row in cursor:
            resList.append(row[0])

    return resList

def CountVerticesAndLength(self, in_polylines1):
    countL = 0
    countV = 0
    totalLen = 0
    individualLen = []

    list_oid = []
    list_vert = []

```

```

list_geodesiclen = []

with arcpy.da.SearchCursor(in_polylines1, ("Shape@", "Shape@Length",
"OID@", self.glen_field2)) as cur:
    for row in cur:
        countL += 1
        countV += row[0].getPart(0).count
        totalLen += row[1]
        individualLen.append(row[1])
        list_oid.append(row[2])
        list_vert.append(row[0].getPart(0).count)
        list_geodesiclen.append(row[3])

return (countL, countV, totalLen, individualLen, list_oid, list_vert,
list_geodesiclen)

def CountVerticesNoProjection(self, in_polylines1):
    countV = 0
    with arcpy.da.SearchCursor(in_polylines1, ("Shape@")) as cur:
        for row in cur:
            countV += row[0].getPart(0).count

    return countV

def getResolutionByLength(self, in_len):
    dem_res = []
    if in_len < 5000:
        dem_res = [10, 30, 90]
    if in_len >= 5000 and in_len < 15000:
        dem_res = [30, 90]
    if in_len >= 15000:
        dem_res = [90]
    return dem_res

def getResolutionByLengthFootprint(self, in_polylines, total_len):
    len_candidates = self.getResolutionByLength(total_len)
    foot_candidates = self.lineFootprintTest(in_polylines)
    self.traceExecutionTime("")
    foot_candidates_int = [int(x) for x in foot_candidates]
    res_list = [i for i in len_candidates if i in foot_candidates_int]
    res_list.sort()
    if len(res_list) == 0:
        arcpy.AddError(self.errorMessages[2])
        raise
    return res_list

```

```

def getResolutionByFootprint(self, in_polylines):
    foot_candidates = self.lineFootprintTest(in_polylines)
    self.traceExecutionTime("")
    foot_candidates_int = [int(x) for x in foot_candidates]
    foot_candidates_int.sort()
    if len(foot_candidates_int) == 0:
        arcpy.AddError(self.errorMessages[2])
        raise
    return foot_candidates_int

def getDefaultNumberVertices(self, in_number_vertices):
    out_num = None
    if in_number_vertices <= 50:
        out_num = 50
    if in_number_vertices > 50 and in_number_vertices <= 200:
        out_num = 200
    if in_number_vertices > 200:
        out_num = in_number_vertices
    return out_num

def densifyLine(self, in_line_features, distanceLU):
    if distanceLU != "": # only do it when not empty
        arcpy.Densify_edit(in_line_features, "DISTANCE", distanceLU)
        self.traceExecutionTime("")

def weedLine(self, in_line_features, in_tol):
    if in_tol <> 0: # only do it when not 0
        arcpy.Generalize_edit(in_line_features, in_tol)
        self.traceExecutionTime("")

def traceExecutionTime(self, msg):
    if self.debug:
        if msg <> None:
            arcpy.AddMessage(msg)
            arcpy.AddMessage(arcpy.GetMessages())

def printCoordinateSystem(self, in_dataset):
    des = arcpy.Describe(in_dataset)
    arcpy.AddMessage(des.SpatialReference.name)

def validateNumerical(self, inVal, paramStr):
    if inVal == None: # None is OK
        return
    elif inVal <= 0:

```



```

        arcpy.AddError(self.errorMessages[7].format(paramStr))
        raise

def validateDistanceUnits(self, inStr, paramStr):
    tempUnitsList = [s.lower() for s in self.listLinearUnits]
    tempUnitsList.extend(["#", ""])
    if inStr == None: # None is OK
        return
    elif not (inStr.lower() in tempUnitsList):
        arcpy.AddError(self.errorMessages[3].format(paramStr))
        raise

def validateInputDEMSource(self, inDEM):
    tempDEMList = [s.upper() for s in self.dictDEMResolutions.keys()]
    tempDEMList.extend(["", "FINEST", "#"])
    if inDEM == None: # None is OK
        return
    elif not (inDEM.strip().upper() in tempDEMList):
        arcpy.AddError(self.errorMessages[1].format(inDEM))
        raise

def validateFeatureIDField(self, inName, inFeature):
    fldList = arcpy.ListFields(inFeature)
    fldListLower = [f.name.lower() for f in fldList]
    if inName == None: # None is OK
        return
    elif not (inName.lower() in fldListLower):
        arcpy.AddError(self.errorMessages[8])
        raise

def formatInputDEMSource(self, inSource):
    tempDEMList = self.dictDEMResolutions.keys()
    tempDEMList.extend(["", "FINEST"])
    retVal = inSource
    for d in tempDEMList:
        if inSource.upper() == d.upper():
            retVal = d
            break
    return retVal

def createProfile(self, in_line_features, inputIsInOcean, line_id_field,
idFieldsTemp, inputSR,
dem_resolution, line_count, list_geodesiclen, out_profile):
    try:
        line_features_inputCS = os.path.join(r"in_memory", r"linetmpafterprj03")

```

```

route_temp = os.path.join(r"in_memory", "out_routetmp")
interp_line_temp = r"in_memory\interpouttmp"
out_vertices_temp = r"in_memory\verticestmp"
arcpy.env.workspace = "in_memory"

# get Z values from DEM

arcpy.InterpolateShape_3d(in_surface=self.getLayerName(dem_resolution),
                          in_feature_class=in_line_features,
                          out_feature_class=interp_line_temp,
                          vertices_only="VERTICES_ONLY")
self.traceExecutionTime("")
# Calculate M values using Create Routes tool
# By default, M is in meters. To change the M unit,
# change the unit in which glen_field2 is calculated (in the execute method)
arcpy.CreateRoutes_lr(in_line_features=interp_line_temp,
route_id_field=line_id_field,
                      out_feature_class=route_temp,
measure_source="TWO_FIELDS",
                  from_measure_field=self.glen_field1,
to_measure_field=self.glen_field2)
self.traceExecutionTime("")

if self.outputToTable: # out to table
    # project the line
    arcpy.env.outputCoordinateSystem = inputSR # convert to input
projection
    arcpy.CopyFeatures_management(route_temp, line_features_inputCS) #
project
    self.traceExecutionTime("")
    arcpy.env.outputCoordinateSystem = ""
    # extract X, Y, Z, M
    arcpy.CreateTable_management("in_memory",
os.path.basename(out_profile), os.path.join(os.path.dirname(__file__),
"profile_schema.dbf"))
    self.traceExecutionTime("")
    with arcpy.da.InsertCursor(out_profile, ("ID", "POINT_X", "POINT_Y",
"POINT_M", "POINT_Z")) as icur:
        with arcpy.da.SearchCursor(line_features_inputCS, ("Shape@",
line_id_field)) as scur:
            for row in scur:
                geo = row[0]
                id_val = row[1]
                for l1 in geo.getPart():
                    for pnt in l1:

```

```

        x = pnt.X
        y = pnt.Y
        m = pnt.M
        z = pnt.Z
        icur.insertRow((id_val, x, y, m, z))
    else: # out to line
        # project the line
        arcpy.env.outputCoordinateSystem = inputSR # convert to input
projection
        arcpy.CopyFeatures_management(route_temp, out_profile) # project
        self.traceExecutionTime("")
        arcpy.env.outputCoordinateSystem = ""
        # Add metadata info
        if inputIsInOcean:
            dem_source = ['1000m']
        else:
            dem_source = [k for k, v in self.dictDEMResolutions.iteritems() if v ==
str(dem_resolution)]
        arcpy.AddField_management(out_profile, self.metadataFieldName,
"TEXT", field_length=50, field_alias="DEM Resolution")
        arcpy.CalculateField_management(out_profile, self.metadataFieldName,
"" + dem_source[0] + "", "PYTHON")
        # Add geodesic length for profile
        arcpy.AddField_management(out_profile, self.geodesicLenFieldName,
"DOUBLE", field_alias="Length Meters")
        i = 0
        with arcpy.da.UpdateCursor(out_profile, self.geodesicLenFieldName) as
ucur:
            for row in ucur:
                row[0] = list_geodesiclen[i]
                i += 1
                ucur.updateRow(row)
        # remove tempid field
        if idFieldsTemp:
            arcpy.DeleteField_management(out_profile, line_id_field)

    except:
        msgs = arcpy.GetMessages(2)
        arcpy.AddError(msgs)
        raise

    def getParameterInfo(self):
        """Define parameter definitions"""
        param0 = arcpy.Parameter(name="InputLineFeatures",
            displayName="Input Line Features",

```

```

        direction="Input",
        parameterType="Required",
        datatype="GPFeatureRecordSetLayer")
# Feature set schema
param0.value = self.profileSchema

param1 = arcpy.Parameter(name="ProfileIDField",
    displayName="Profile ID Field",
    direction="Input",
    parameterType="Optional",
    datatype="Field")
param1.filter.list = ['OID', 'Short', 'Long']

param2 = arcpy.Parameter(name="DEMResolution",
    displayName="DEM Resolution",
    direction="Input",
    parameterType="Optional",
    datatype="GPString")
param2.filter.type = "ValueList"
list_dem = [" ", "FINEST"]
dem_keys = self.dictDEMResolutions.keys()
dem_keys.sort()
list_dem.extend(dem_keys)
param2.filter.list = list_dem

param3 = arcpy.Parameter(name="MaximumSampleDistance",
    displayName="Maximum Sample Distance",
    direction="Input",
    parameterType="Optional",
    datatype="GPDouble")

param4 = arcpy.Parameter(name="MaximumSampleDistanceUnits",
    displayName="Maximum Sample Distance Units",
    direction="Input",
    parameterType="Optional",
    datatype="GPString")

param4.filter.type = "ValueList"
param4.filter.list = self.listLinearUnits
param4.value = "Meters"

param5 = arcpy.Parameter(name="OutputProfile",
    displayName="Output Profile",
    direction="Output",
    parameterType="Derived",

```

```

        datatype="DEFeatureClass")

    params = [param0, param1, param2, param3, param4, param5]
    return params

def isLicensed(self):
    """Set whether tool is licensed to execute."""
    return True

def updateParameters(self, parameters):
    """Modify the values and properties of parameters before internal
    validation is performed. This method is called whenever a parameter
    has been changed."""
    return

def updateMessages(self, parameters):
    """Modify the messages created by internal validation for each tool
    parameter. This method is called after internal validation."""
    return

def execute(self, parameters, messages):
    """The source code of the tool."""
    try:
        startTime = time.time()
        self.debug = False
        in_polylines = parameters[0].value
        profile_id_field = parameters[1].valueAsText
        dem_resolution_p = parameters[2].valueAsText
        sample_distance_p = parameters[3].value
        sample_distance_units = parameters[4].valueAsText
        out_profile = os.path.join("in_memory", "profile1")

        arcpy.env.overwriteOutput = True
        maxInputLines = 100 # sync is 100, async is 1000
        if ("elevation_gpserver" in arcpy.env.scratchWorkspace):
            maxInputLines = 1000

        # Get input SR
        d0 = arcpy.Describe(in_polylines)
        inputSR = d0.spatialReference
        oidfld1 = d0.OIDFieldName

        # project first
        polylines_after_prj = os.path.join("in_memory", "inputlinetmp02")
        # project to raster coordinate system

```

```

        arcpy.env.outputCoordinateSystem = self.demCoordinateSystem

        arcpy.CopyFeatures_management(in_polylines,    polylines_after_prj)    #
project
        arcpy.env.outputCoordinateSystem = ""
        self.traceExecutionTime("")

        # Add and calculate geodesic length fields - from field and to field for
        # Create Routes tool to calculate the M values
        arcpy.AddField_management(polylines_after_prj,    self.glen_field1,
"DOUBLE")
        self.traceExecutionTime("")
        arcpy.CalculateField_management(polylines_after_prj, self.glen_field1,
            "0", "PYTHON_9.3")
        self.traceExecutionTime("")
        # The unit in which glen_field2 is calculated determines the M unit.
        # To change it other units, replace meters below with desired units
        arcpy.AddField_management(polylines_after_prj,    self.glen_field2,
"DOUBLE")
        self.traceExecutionTime("")
        arcpy.CalculateField_management(polylines_after_prj, self.glen_field2,
            "!shape.geodesiclength@meters!", "PYTHON_9.3")
        self.traceExecutionTime("")

        # validate profile id field
        time_a = time.time()
        self.validateFeatureIDField(profile_id_field, in_polylines)
        time_b = time.time()
        if self.debug:
            arcpy.AddMessage("ValidateFeatureIDField    execution    time:    "    +
str(time_b - time_a))

        # make temp id field
        idFieldsTemp = False
        temp_id_field = "tmpprflid_"
        if profile_id_field == None:
            idFieldsTemp = True # needed for field removal later
            fieldSp = 0
            fieldsObjID = 1
            profile_id_field = temp_id_field
        elif profile_id_field.lower() in ["oid", "fid", "objectid"]:
            idFieldsTemp = True # needed for field removal later
            fieldSp = 1
            fieldsObjID = 0
            profile_id_field = temp_id_field

```

```

        if profile_id_field == temp_id_field: # default
            arcpy.AddField_management(polylines_after_prj, profile_id_field,
"LONG")
            self.traceExecutionTime("")
            arcpy.CalculateField_management(polylines_after_prj, profile_id_field,
"!"+oidfld1+"!", "PYTHON_9.3")
            self.traceExecutionTime("")

        # var for metering
        fieldSp = 1
        fieldIsObjID = 1
        samplingDistSp = 0

        # now find the line length and number of vertices
        time_a = time.time()
        lineFact = self.CountVerticesAndLength(polylines_after_prj)
        time_b = time.time()
        if self.debug:
            arcpy.AddMessage("CountVerticesAndLength execution time: " +
str(time_b - time_a))

        line_counts = lineFact[0]
        total_num_vert = lineFact[1]
        total_len = lineFact[2]
        indiv_len = lineFact[3]
        list_oid = lineFact[4]
        list_vert = lineFact[5]
        list_glen = lineFact[6]

        if line_counts < 1:
            arcpy.AddError(self.errorMessages[0])
            raise
        elif line_counts > maxInputLines:
            arcpy.AddError(self.errorMessages[9])
            raise

        self.validateNumerical(sample_distance_p, "Maximum Sample Distance")
        self.validateDistanceUnits(sample_distance_units, "Maximum Sample
Distance Units")
        self.validateInputDEMSource(dem_resolution_p)

        # trim dem_resolution_p
        if dem_resolution_p is not None and str(dem_resolution_p).upper() <>
"FINEST":

```



```

        if dem_resolution_p.strip() == "":
            dem_resolution_p = None
        if dem_resolution_p is not None:
            dem_resolution_p
self.dictDEMResolutions[self.formatInputDEMSource(dem_resolution_p)]

# determine whether input line is in ocean
inputIsInOcean = False
# determine resolution
if str(dem_resolution_p).upper() <> "FINEST":
    if dem_resolution_p is None: # case 1 blank (default)
        dem_resolution = self.defaultDEMResolution
        res_list = self.getResolutionByFootprint(polylines_after_prj)
        if not int(dem_resolution) in res_list:
            arcpy.AddError(self.errorMessages[2])
            raise
            return
    else: # case 2 specified
        dem_resolution = dem_resolution_p
        res_list = self.getResolutionByFootprint(polylines_after_prj)
        if not int(dem_resolution) in res_list:
            arcpy.AddError(self.errorMessages[2])
            raise
            return
else: # case 3 - FINEST:
    res_list = self.getResolutionByFootprint(polylines_after_prj)
    dem_resolution = str(int(res_list[0]))

if sample_distance_units == None:
    sample_distance_units = "meters"

outfeaturelayer1 = "tempfeaturelayer"
arcpy.MakeFeatureLayer_management(polylines_after_prj,
outfeaturelayer1)
self.traceExecutionTime("")

for oid_val in list_oid:
    query_exp = oidfld1 + "=" + str(oid_val)
    arcpy.SelectLayerByAttribute_management(outfeaturelayer1,
"NEW_SELECTION", query_exp)
    self.traceExecutionTime("")

in_len = indiv_len[list_oid.index(oid_val)] # individual line length
in_glen = list_glen[list_oid.index(oid_val)] # individual glength

```

=

```

ratio1 = in_len / (in_glen / self.zFactor) # ratio to convert to Mercator
in_num_vert = list_vert[list_oid.index(oid_val)] # individual line vertex
number

if sample_distance_p == None: # default
    samplingDistSp = 0 # metering
    out_num_vert = in_num_vert
    needDensify = False
    needWeed = False
    if in_num_vert < 50:
        out_num_vert = 50
        needDensify = True
    elif in_num_vert >= 50 and in_num_vert < 200:
        out_num_vert = 200
        needDensify = True
    elif in_num_vert >= 200 and in_num_vert <= self.maxNumVertices:
        out_num_vert = in_num_vert
        needDensify = False
    elif in_num_vert > self.maxNumVertices:
        out_num_vert = self.maxNumVertices
        needDensify = False
        needWeed = True

sample_distance_m = in_len / (out_num_vert - 1) # default sample
distance

if needDensify:
    # change the unit here to DEM linear unit, eg, feet, meters,
decimaldegrees
    self.densifyLine(outfeaturelayer1, str(sample_distance_m) + " " +
self.demLinearUnit)
    if needWeed:
        self.weedLine(outfeaturelayer1, str(int(dem_resolution) / 4.0) + " " +
self.demLinearUnit)
    else: # specified
        samplingDistSp = 1 # metering
        newSamplingDist = ratio1 * sample_distance_p # convert to GCS
distance
        sample_distance_m = newSamplingDist *
self.getUnitConversionFactor(sample_distance_units) / self.zFactor # convert to
Feet
        nVert = int((in_len / sample_distance_m) + 1)
        if nVert > self.maxNumVertices:
            arcpy.AddError(self.errorMessages[5])
            raise

```

```

        return
    else:
        self.densifyLine(outfeaturelayer1, str(sample_distance_m) + " " +
self.demLinearUnit)

        # final count of no. vertices
        nVert1 = self.CountVerticesNoProjection(outfeaturelayer1)
        if nVert1 > self.maxNumVertices * 2:
            arcpy.AddError(self.errorMessages[6])
            raise
            return


        # Execute the tool, line is already densified
        arcpy.AddMessage("DEM Resolution: " + dem_resolution + ", Sampling
Distance: "
                        + str(sample_distance_m))
        self.createProfile(polylines_after_prj, inputIsInOcean, profile_id_field,
idFieldIsTemp, inputSR,
                        dem_resolution, line_counts, list_glen, out_profile)

        arcpy.SetParameterAsText(5, out_profile)

except:
    msgs = arcpy.GetMessages(2)
    arcpy.AddError(msgs)

```

ANEXO C FORMULARIO LEVANTAMIENTO DE REQUERIMIENTOS

 Constructor de Formularios

Nombre

Nombre del funcionario

Cargo

Cargo o funcion que desempeña

Pertenece al grupo

- SIA
- Usuarios CVC
- Admin Carto CVC
- Seleccione la o las funcionalidades que crea conveniente para el visor web
- Desarrollo en Android para Tablet
- Seleccionar mediante el mouse un recuadro para consultar
- Seleccionar a partir de un layer que intersecte las aerofotos y las presente en el visor
- Seleccionar mediante un dibujo dentro del area de estudio para consultar las imagenes que coincidad con esa area, linea o punto dibujado
- Visualizar los metadatos de una imagen o grupo de imagenes dentro de un área de estudio
- Permitir una impresion fisica de la consulta realizada al area de estudio
- Permitir la interaccion con la Base de datos geografica y la Base de datos Raster
- Permitir la integracion del visor raster con nuevas aplicaciones
- Consultar la disponibilidad de las fotografías aéreas para solicitar préstamo en físico.
- Consultar disponibilidad de las fotografias aéreas para imprimir o descargar las fotografias aéreas crudas escaneadas, escaneadas georreferenciadas y ortorrectificadas
- Permitir la descarga de las imagenes seleccionadas dle area de estudio en los formatos mas conocidos TIFF,JPeg, etc,
- Permitir ver los diferentes estados de las imágenes satelitales, mosaicos de fotografías aéreas y fotografías aéreas individuales que se van a descargar del sistema
- Crear los niveles de autorización del uso de las imágenes satelitales, mosaicos de fotografías aéreas y fotografías aéreas individuales en la visualización, descarga e impresión
- Permitir imprimir en tamaño carta las imágenes seleccionadas o la composición vectorial raster que se construya