Universidad Central "Marta Abreu" de Las Villas

Facultad Matemática, Física y Computación

Ciencias de la Computación

Trabajo de diploma

Algoritmos para la transformación de datos espacio-temporales mediante el uso de formatos de datos científicos en sistemas de información geográfica.

Autor: Carlos Guillermo Flores Martínez

Tutor: Dr. Romel Vázquez Rodríguez

Santa Clara

2017

Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

_____

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____ _____

Firma del tutor Firma del Jefe de Laboratorio

Dedicatoria

Le dedico este trabajo a mis padres, hermanos, a mi abuelo aunque ya no este conmigo y a toda mi familia.

Agradecimientos

Le agradezco a toda mi familia, a los compañeros de aula que estuvieron siempre presente, a todo aquel que aportó un granito de arena para que yo terminara mis estudios.

Resumen

Un Sistema de Información Geográfica (SIG) contribuye al manejo de grandes volúmenes de información, realizan eficientemente la representación, análisis y el procesamiento, por lo que son muy utilizados en distintas áreas del saber. Los campos de aplicación para modelar fenómenos que requieren componentes espaciales y temporales son cada vez más diversos y amplios. La información que se genera diariamente es almacenada en diversos formatos de datos científicos, los cuales se crean en necesidad de la comunidad que los necesite.

Este trabajo de diploma resume los resultados de una amplia investigación, donde se diseñó y crearon herramientas para la manipulación, conversión y generación de formatos de datos científicos espacio-temporales. Estas herramientas fueron incorporadas a una biblioteca que se integró a gvSIG y puede ser integrada a otro SIG que esté escrito en el mismo lenguaje. Para el proceso de validación y presentación de los resultados se presentó un caso de estudio con datos climáticos sobre la República de Cuba, que contiene información sobre diez variables climatológicas correspondientes al periodo 1901-2012. Los datos fueron suministrados por la Unidad de Investigación Climática (CRU) de la Universidad del este de Anglia en el Reino Unido. Con este caso de estudio se crearon archivos en formatos de datos científicos espacio-temporales que pueden ser objetos de estudio por entes especializados en climatología.

Abstract

A Geographic Information System (GIS) contributes to the management of large volumes of information, efficiently perform representation, analysis and processing, so they are widely used in different areas of knowledge. The fields of application for modeling phenomena that require spatial and temporal components are increasingly diverse and broad. The information that is generated daily is stored in various formats of scientific data, which are created in need of the community that needs them.

This diploma paper summarizes the results of extensive research, where tools for the manipulation, conversion and generation of space-time scientific data formats were designed and created. These tools were incorporated into a library that was integrated into gvSIG and can be integrated into another GIS that is written in the same language. For the validation process and presentation of the results, a case study with climatic data on the Republic of Cuba was presented, containing information on ten climatological variables corresponding to the period 1901-2012. The data were supplied by the Climate Research Unit (CRU) of the University of East Anglia in the United Kingdom. With this case study files were created in formats of scientific data space-time that can be objects of study by specialized entities in climatology.

Tabla de Contenido

Tabla de Contenido

Tabla de contenidos

Lista de Figuras

Introducción

En diferentes áreas de conocimiento los Sistemas de Información Geográfica (SIG) son muy importantes. Entre estas áreas se encuentran la cartografía, geografía, ciencias naturales, climatología y oceanografía. Actualmente existe dependencia de los modelos dinámicos que adicionen el tiempo como variable para observar el comportamiento de eventos geográficos, donde se genera datos espacio temporales(Durango 2013).

La creciente ola de información que se genera diariamente, es almacenada en diversos formatos de datos. Muchos de estos formatos han sido creados por instituciones o comunidades en dependencia de la necesidad de almacenamiento que estos posean para sus grandes volúmenes de datos. Gran parte de esta información almacenada está relacionada con variables climáticas y datos asociados a otros eventos que ocurren cotidianamente, con el objetivo de analizarlos y poder arribar a conclusiones sobre el tema.

La mayoría de estos datos están relacionados con el medio ambiente, muchos tienen componente espacial y temporal. La componente espacial significa que los datos pueden ser georreferenciados o lo que es lo mismo tienen una localización geográfica en el espacio, mientras que el componente temporal significa que los datos son tomados en varios instantes de tiempo(Vázquez Rodríguez, 2015).

Los SIG se desarrollan con el objetivo de incorporar nuevos métodos de análisis para examinar estos datos de forma efectiva y extraer de ellos información y determinados conocimientos. En general un SIG es una unión estructurada de hardware, software y datos geográficos diseñado para almacenar, manipular, analizar, capturar y desplegar en todas sus formas la información geográficamente referenciada, con el fin de resolver determinadas problemáticas de planificación y gestión. Los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de diferentes operaciones(BOLSTAD 2005).

La mayoría de los elementos que existen en la naturaleza pueden ser representados mediante formas geométricas (puntos, líneas o polígonos), esto es reconocido en el mundo de los SIG como formato de datos de tipo vector. La representación de información mediante celdillas

regulares es conocida como el formato *raster*. Se suele seleccionar una estructura de datos vectorial cuando hay que reflejar más de un atributo en un mismo espacio. Usar un formato *raster* permite crear una capa distinta para cada atributo. Los datos espaciales que manipulan los SIG pueden ser de tipo *raster* y tipo vector y su uso se relaciona con la captura, almacenamiento, transformación y análisis de datos espaciales. Los SIG pueden ser libres y propietarios (Murphy 1995; GUTIERREZ PUEBLA, J. G. 1997).

Antecedentes

SEXTANTE es un conjunto de algoritmos de análisis geoespacial de código libre desarrollado para la Junta de Extremadura, al cual se le pueden implementar y añadir nuevos algoritmos, este desarrollo se lleva a cabo apoyándose de algún software ya existente, donde se implementa las capacidades requeridas.

SEXTANTE está integrado en algunos de los SIG más importantes escritos en Java y también se puede incorporar a otros no escritos en Java, lo cual demuestra su importancia dentro de las comunidades SIG.

Anteriormente en (Roque 2014) se realizó un análisis de los formatos de datos científicos más utilizados hasta el momento para su posterior incorporación al módulo SEXTANTE de gvSIG, implementando un conjunto de algoritmos para la manipulación de los mismos.

En tesis anteriores se realizaron algoritmos para la transformación de formatos de datos científicos, con los mismos no se logró una correcta conversión de estos formatos de datos científicos investigados.

Planteamiento del problema

En estudios realizados con el módulo de SEXTANTE para la manipulación de formatos de datos científicos se detectaron algoritmos que pueden ser mejorados para la transformación correcta de algunos formatos de datos científicos. Además no se han incluido otros formatos que pueden ser de utilidad para la manipulación de datos espacio-temporales.

Son varios los sistemas de información geográfica y otras herramientas que incorporan estos formatos y no existen herramientas que permitan la transformación efectiva de datos en estos

formatos a otros tipos de formatos de datos científicos. Se necesita desarrollar herramientas que permitan la conversión de estos datos utilizando nuevos formatos.

Objetivo general

Implementar nuevos algoritmos para transformar datos espacio-temporales con formatos de datos científicos en SEXTANTE.

Objetivos específicos

1. Seleccionar de los principales formatos de datos científicos espacio-temporales los más adecuados para integrarlos a SEXTANTE.
2. Modificar los algoritmos que no tienen un correcto funcionamiento en SEXTANTE.
3. Implementar herramientas que permitan el trabajo con nuevos formatos de datos científicos espacio-temporales en SEXTANTE.
4. Validar las herramientas elaboradas mediante un caso de prueba.

Para darle solución a la problemática planteada y responder los objetivos fueron formuladas las siguientes preguntas de investigación.

Preguntas de investigación

1. ¿Cuáles son los principales formatos de datos científicos espacio-temporales utilizados para el almacenamiento de grandes volúmenes de datos en el área de las geociencias?

2. ¿Qué ventajas pueden brindar la integración de formatos de datos científicos espacio-temporales a gvSIG?

Justificación de la investigación

Actualmente existen diversos formatos de datos científicos para almacenar grandes volúmenes de información, lo cual nos obliga a realizar una investigación para ver sus ventajas y desventajas y seleccionar entre todos los que más se adapten a nuestro propósito. Además sería de gran utilidad dotar a los SIG libres de funcionalidades que permitan el trabajo con los

formatos de datos científicos, teniendo en cuenta la gran utilización de estos formatos por muchas instituciones científicas.

1. Sistemas de información geográfica(SIG) y Formatos de Datos Científicos Espacio-temporales

En este capítulo se dará una visión general del comportamiento de un SIG, abordaremos en que consiste gvSIG así como el módulo SEXTANTE. Posteriormente se realizará un estudio sobre algunos formatos de datos científicos espacio-temporales para conocer las potencialidades de cada uno de ellos, así como sus principales usos, características y aplicaciones.

1.1 Sistemas de Información Geográfica

En cualquier tipo de disciplina gran parte de la información que manejamos esta georreferenciada, acercándose al 70 %. Se trata de información que puede asignársele una determinada ubicación geográfica, y por tanto esta viene acompañada de información relativa a su localización. El creciente auge de las aplicaciones que trabajan con esta información ha evidenciado que la situación es favorable para el desarrollo de las mismas. Entre los pilares fundamentales de la sociedad moderna, la información y la tecnología son fundamentales, la tecnología rectora para el manejo de la información geográfica es sin duda alguna los SIG; así como los elementos básicos que conlleva la gestión de todo aquello que presente una componente geográfica que pueda ser aprovechada.

¿Qué es un SIG?

Varios son los conceptos y definiciones referentes a los SIG.(BOLSTAD 2005) define un SIG como "un sistema computacional que ayuda en la recolección, mantenimiento, almacenamiento, análisis, visualización y distribución de información y datos espaciales". Una definición clásica es la de (TOMLIN 1990), para quien un SIG es un elemento que permite "analizar, representar e interpretar hechos relativos a la superficie terrestre". Este autor sin embargo plantea, que "esta es una definición muy amplia, y habitualmente se emplea una más concreta. Concretamente, un SIG es un conjunto de *software* y *hardware* diseñado específicamente para la adquisición, mantenimiento y uso de datos cartográficos". De manera similar,(STAR, J. & ESTES 1990) define un SIG como un "sistema de información diseñado para trabajar con datos referenciados mediante coordenadas espaciales o geográficas. En otras palabras, un SIG es tanto un sistema de base de datos con capacidades específicas para datos georreferenciados, como un conjunto de

operaciones para trabajar con esos datos. En cierto modo, un SIG es un mapa de orden superior". Todas estas definiciones recogen el concepto fundamental de los SIG en el momento en que fueron escritas, pero hoy en día se hace necesario recoger otras ideas, la definición actual de un SIG debe fundamentarse sobre todo en el concepto *sistema*, como elemento integrador que engloba un conjunto de componentes interrelacionados. Por lo antes dicho, una definición más precisa es decir que un SIG es un "sistema que integra tecnología informática, personas e información geográfica y cuya principal función es capturar, analizar, almacenar, editar y representar datos georreferenciados" (Olaya 2009).

Funcionamiento de un SIG

Algo imprescindible para comprender todo SIG es el concepto de capa, que es una de las grandes virtudes inherentes a los SIG. En el caso de un SIG, los distintos tipos de información se pueden combinar de forma sencilla y limpia. Esto es así debido a que la idea de capa permite dividir la información espacial referida a una zona de estudio en varios niveles, de tal forma que, pese a coincidir sobre un mismo emplazamiento, la información sobre distintas variables se encuentra recogida de forma independiente. Es decir, en función de la componente temática se establecen distintos bloques de datos espaciales. Para comprender mejor el concepto de capa, pensemos en un mapa topográfico clásico. En él vamos a encontrar elementos como curvas de nivel, carreteras, núcleos urbanos, o simbología relativa a edificios y puntos singulares (iglesias, monumentos, etc.) Todos estos elementos en su conjunto componen el mapa, y aparecen en una misma hoja como una unidad coherente de información geográfica. No obstante, cada uno de estos grupos de información recogida: elevaciones, red vial, núcleos urbanos y puntos de interés arquitectónico, pueden recogerse de forma independiente y combinarse al componer el mapa según las necesidades del momento, o bien combinarse de modo distinto o emplearse individualmente (Figura 1).

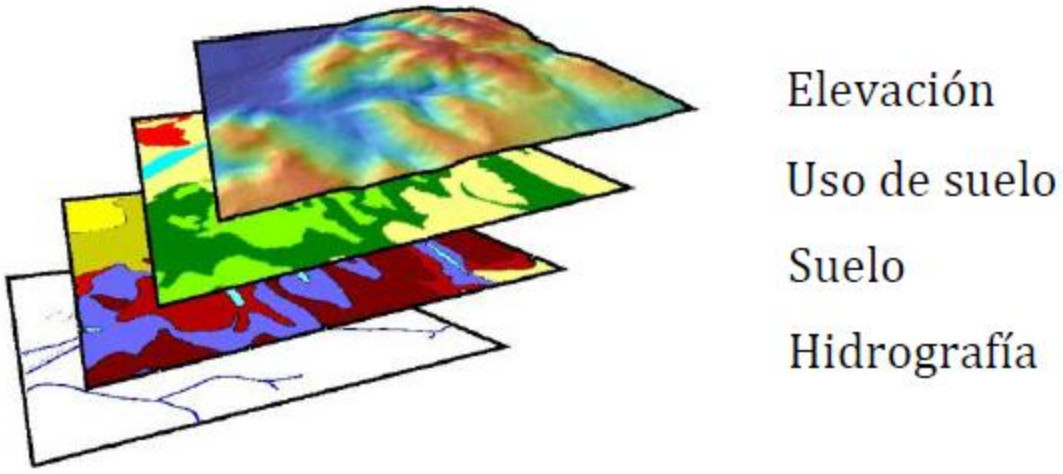


Figura 1: Concepto de capa de información geográfica dentro de un SIG. Fuente http://volaya.github.io/libro-sig/chapters/introduccion_datos.html

La figura es lo suficientemente clara como para entender el concepto de capa, ya que de ella resulta una serie de diferentes niveles que se pueden superponer según el criterio particular de cada usuario del SIG.

Toda la información geográfica con que trabajemos en un SIG va a ser en forma de capas. El SIG funciona como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, si se señala un objeto se conocen sus atributos, e inversamente, si se pregunta por un registro de la base de datos se puede saber su localización en la cartografía. La razón fundamental para utilizar un SIG es la gestión de información espacial. El sistema permite separar la información en diferentes capas temáticas y la almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, lo que facilita la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no podríamos obtener de otra forma. Las principales cuestiones que puede resolver un SIG, ordenadas de menor a mayor complejidad, son:

1. **Localización**: preguntar por las características de un lugar concreto.
2. **Condición**: el cumplimiento o no de unas condiciones impuestas al sistema.

3. **Tendencia**: comparación entre situaciones temporales o espaciales distintas de alguna característica.
4. **Rutas**: cálculo de rutas óptimas entre dos o más puntos.
5. **Pautas**: detección de pautas espaciales.
6. **Modelos**: generación de modelos a partir de fenómenos o actuaciones simuladas.

Los SIG han diversificado y aumentado su campo de aplicación, por lo que son utilizados en la mayoría de las actividades con componente espacial. Por su gran importancia (GUTIERREZ PUEBLA, J. G. 1997) divide los SIG en propietarios y no propietarios.

A continuación analizaremos el SIG gvSIG que se clasifica dentro del grupo de SIG no propietarios.

gvSIG

gvSIG (Generalitat Valenciana SIG) surge como un proyecto amparado por la Generalitat Valenciana de España que, a finales de 2003, promocionó un concurso para el desarrollo de un SIG con una serie de características propias como: multiplataforma, de código abierto, modular, con licencia GPLv2, interoperable, con formatos de otros programas (Autocad, Microstation, Arcview) y sujeto a estándares de la OGC (Open Geospatial Consortium) (Anguix and Carrión, 2005). El resultado ha sido una aplicación que ya tiene disponibles varias versiones al público y gran parte de las funcionalidades propias de los SIG cubiertas, aunque se desarrolla constantemente.

Las funciones básicas que cualquier usuario desearía como diseño de impresión o soporte de formatos de imagen típicos, están incorporadas sin necesidad de ningún módulo adicional. gvSIG posee una jerarquía de clases bien estructurada para la incorporación de nuevas funcionalidades. Permite la lectura de varios formatos de datos geográficos y no geográficos en forma de tablas, así como la conexión con varias bases de datos. Este SIG posee las aplicaciones traducidas a veinte idiomas; toda la documentación está disponible en 5 idiomas, incluyendo español e inglés, por lo que se ha convertido en un SIG muy popular en el mundo hispano (Anguix, 2009). Se ha reportado su utilización en varios países europeos como Francia, Italia, Suiza, Austria, Reino Unido y Alemania, donde se encuentra la mayor comunidad de usuarios de gvSIG no

hispanohablantes. Varias instituciones y universidades prestigiosas han utilizado esta aplicación, tal es el caso de la Agencia Espacial Europea y Oxford Archaeology. Varios países africanos también han realizado trabajos con gvSIG, pero su mayor uso se ha reportado en Iberoamérica.

La simplicidad para la incorporación de nuevas funcionalidades, la disponibilidad de toda su documentación, el soporte técnico suministrado por la lista de distribución de sus desarrolladores y el apoyo institucional del proyecto, hace que se considere el SIG más adecuado para la incorporación de un módulo de visualización científica para el trabajo con formatos de datos científicos.

Entre las funcionalidades que encontramos en gvSIG están:

- Acceso a formatos vectoriales, *raster*, servicios remotos, bases de datos y tablas.
- Consultas.
- Geoprocesos.
- Representación vectorial y *raster*.
- Redes.

1.2 SEXTANTE

Conjunto de algoritmos de análisis geoespacial de código libre desarrollado para la Junta de Extremadura, al cual se le pueden implementar y añadir nuevos algoritmos, este desarrollo se lleva a cabo apoyándose de algún software ya existente, e implementando las capacidades requeridas. SEXTANTE está integrado en algunos de los SIG más importantes escritos en Java y también se puede incorporar a otros no escritos en Java, lo cual demuestra su importancia dentro de las comunidades SIG (Olaya 2011). En sus comienzos SEXTANTE tenía como base el SIG Alemán SAGA, Al cual se le incorporaron un amplio número de funcionalidades y modificaciones en su núcleo. Desde la versión 1.10 de gvSIG, este ha sustituido a SAGA como software base, principalmente por contar con una estructura de apoyo más sólida y con mayor potencial futuro.

Figura 2: Elementos básicos de SEXTANTE.

1-Caja de Herramientas.

2-Modelador Gráfico.

3-Linea de Comandos.

4-Historial.

5-Resultados.

Caja de Herramientas

Es el elemento principal para el control de las extensiones o módulos. Este gestor conforma un conjunto de herramientas con todas las extensiones de SEXTANTE que pueden ejecutarse desde el mismo. A su vez, estas extensiones se agrupan en bloques de acuerdo con el tipo de análisis que se lleva a cabo, para así facilitar su empleo y manejo (Figura 3: Caja de herramientas).

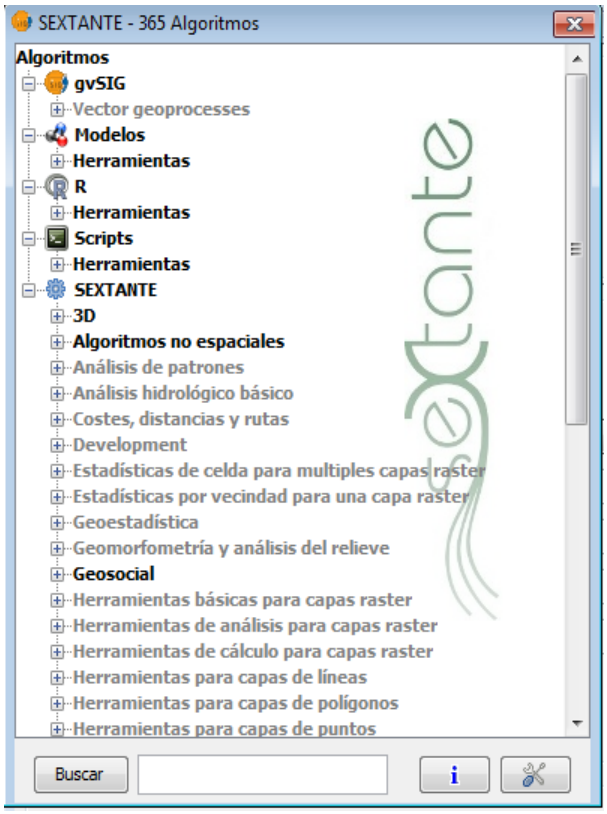


Figura 3: Caja de herramientas.

Modelador Gráfico

Para la creación de modelos complejos mediante una interfaz sencilla lo utilizado es el modelador gráfico, logrando así la simplificación de los procesos que impliquen el uso de varias extensiones de SEXTANTE de forma encadenada. Mediante el modelador puede diseñarse de forma sencilla una nueva extensión que tome datos del usuario y mediante ellos alimente a una serie de extensiones, de forma que las salidas generadas por estas puedan ser empleadas como entradas en otras distintas. De esta manera los procesos que implican varios pasos pueden reducirse así a uno único, definiendo el flujo de datos entre las distintas extensiones involucradas.

El modelador cuenta con un lienzo de trabajo donde se ve la estructura del modelo planteado, y en la parte izquierda un conjunto de elementos que se pueden añadir al modelo para ir conformándolo progresivamente (Figura 4: Modelador Grafico).

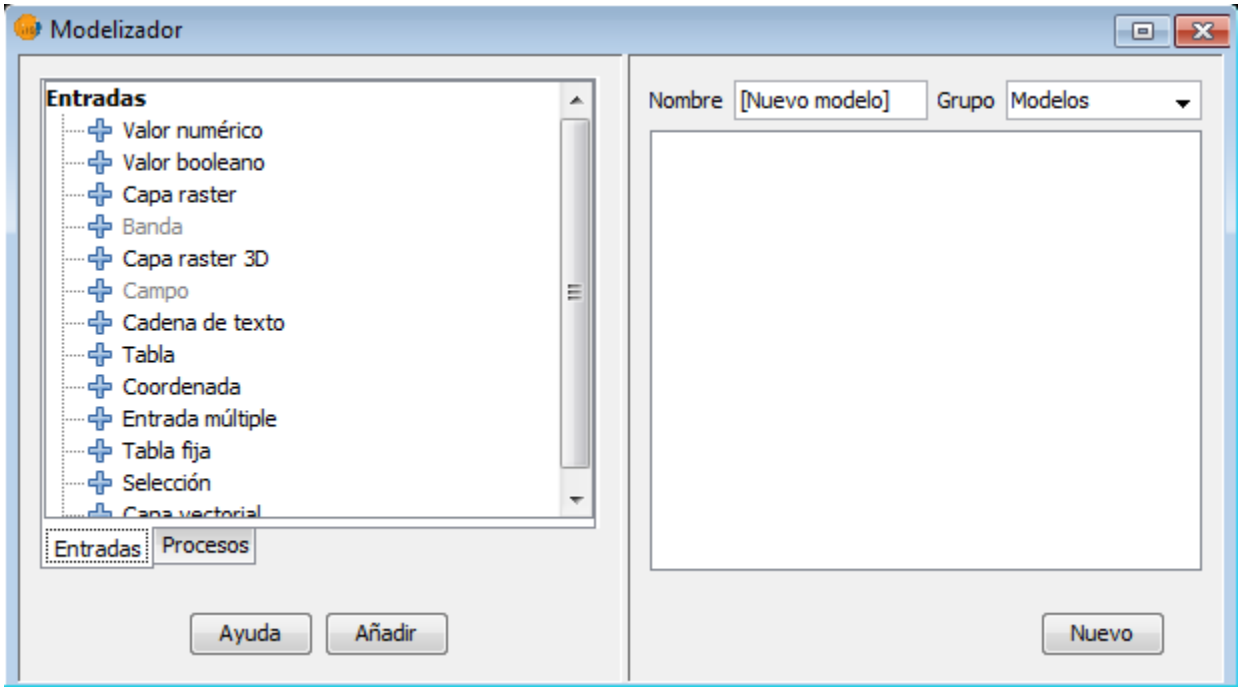


Figura 4: Modelador Grafico

Línea de Comandos

La consola de SEXTANTE permite a los usuarios más avanzados hacer un uso más ágil del programa y automatizar tareas mediante la creación de sencillos scripts (Figura 5: Línea de Comandos).

Figura 5: Línea de Comandos

Historial

Registra los distintos procesos que se han ejecutado con SEXTANTE, sin importar el lugar de donde fueron ejecutados, ya sea desde la línea de comandos o desde el gestor de extensiones (Figura 6: Historial).

Figura 6: Historial

1.3 Datos espacio-temporales

Los modelos espacio-temporales se crean cuando los datos se agrupan por tiempo, así como por espacio. Muestra clara de esto sería el de una secuencia de estudios desarrollados a contaminantes atmosféricas u otras variables de determinado interés para alguna comunidad científica, donde los datos son capturados a intervalos regulares de tiempo, pudieran ser durante el día a día, por semanas o durante algún intervalo de tiempo. Así el análisis tiene que tomar cuenta de dependencia espacial entre los seguimientos o monitoreos, pero también las observaciones a cada monitoreo no son típicamente independientes, pero forman una serie de tiempo.

En otras palabras, se debe tomar cuenta de correlaciones temporales, así como las correlaciones espaciales. Hasta el momento no había existido una teoría de los procesos espacio-temporales alejada de la ya bien establecida teoría de la estadística espacial y el análisis de las series de datos(Durango 2013). El creciente aumento de los datos espacio-temporales en los últimos años ha aumentado las investigaciones y el estudio de las series de tiempo unido al estudio de los datos espacio-temporales.

Los datos espaciales son una representación del mundo real o cotidiano. Dicha representación trae consigo la incorporación de modelos dinámicos, dependientes de las variables temporales y espaciales. Una gran cantidad de estudios han permitido acrecentar las capacidades y condiciones de desarrollo para herramientas que realicen análisis de variables temporales. Para modelar y entender la evolución de los fenómenos geográficos el tiempo es determinante, se considera que las transformaciones del espacio geográfico dependen de las actividades y los fenómenos espacio-temporales (Durango 2013).

Actualmente los estudios que examinan el espacio y el tiempo, dan por hecho que cada elemento geográfico desempeña múltiples roles en un momento específico en el tiempo; esto condujo a sugerir que no existe un área en el espacio geográfico que se pueda separar del flujo del tiempo y el espacio. Una entidad geográfica tiene una ruta espacio-temporal, que comienza en el instante que se confecciona y termina cuando se destruye. Durante diferentes etapas del ciclo de vida sufre una serie de cambios en su localización y en los eventos que la afectan en el tiempo. Por lo

tanto, el tiempo y el espacio están estrechamente relacionados al punto de no poderse analizar por separados.

Las entidades geográficas presentan una ruta espacio-temporal (Figura 7), que inicia en el momento de la toma de los geodatos y termina en el momento que se destruyen los geodatos. Partiendo de la anterior descripción, los datos espacio-temporales se representan como una inserción de la dimensión tiempo en entidades geográficas concebidas, donde el espacio geográfico se organiza en capas temáticas que incluyen la información de toma de los datos en un tiempo determinado(Vázquez Rodríguez 2015; Durango 2013).

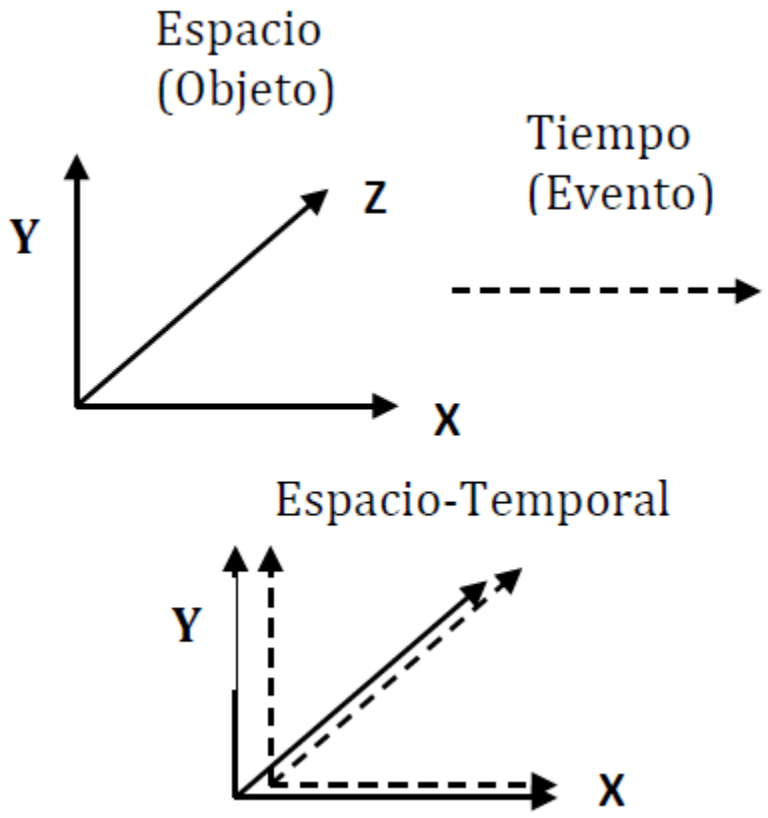


Figura 7: Representación de la entidad Espacio-temporal. Fuente (Vázquez Rodríguez 2015).

Para el almacenamiento de los datos espacio-temporales existen diferentes formatos digitales. A lo largo de los años, a medida que han surgido problemas de almacenamiento, diferentes instituciones han generado nuevos formatos de datos que se ajusten a sus necesidades específicas. Para atender la compactibilidad, portabilidad y otras características las comunidades SIG requieren de la creación de estos formatos(Ramachandran et al. 2004).

1.4 Formato de datos espacio temporales

Los formatos de datos espacio-temporales que utilizan las comunidades e instituciones científicas para intercambiar y almacenar los grandes volúmenes de información son diversos (McGrath 2003). Las comunidades SIG que poseen estos formatos son las encargadas de determinar su nivel de uso dependiendo del dominio que posea. Varios de estos formatos son analizados en este epígrafe con el fin de saber sus principales características y aplicaciones.

1.4.1 Hierarchical Data Format (HDF)

Hierarchical Data Format, más conocido por **HDF,** fue desarrollado por el Centro Nacional de Aplicaciones de Supercómputo (*National Center for Supercomputing Applications*, NCSA) en el año 1988. En la actualidad, su soporte corre a cargo de *HDF Group* de la universidad de Illinois. Es un formato de datos de propósito general, flexible, portable y eficiente para el almacenamiento y recuperación de datos científicos(Ullman & Denning 2012; Poinot 2010). HDF es también un conjunto de bibliotecas escritas en Java, eficiente en las operaciones de E/S debido al uso de paralelismo (Gray et al., 2005), es libre, de código abierto y multiplataforma. Dado su excelente desempeño y simplicidad ha sido ampliamente usada por muchas comunidades de científicos (Long et al. 2013). Este formato es muy usado por entidades que producen y gestionan información de carácter ambiental y otras entidades de observación territorial, como es el caso de la NASA (Vanegas, 2013).

La distribución consiste en la biblioteca, utilidades de línea de comando, una suite de prueba, interfaces con Java y HDFView que es un potente visor basado en Java, mediante el cual los usuarios pueden observar fácilmente detalles interesantes de los datos. HDF presenta cuatro niveles de interacción. En su nivel más bajo es un archivo para el almacenamiento de datos científicos. En su nivel más alto es una colección de utilidades y aplicaciones para manipular, ver y analizar datos en los ficheros HDF, y en los niveles intermedios se encuentra una biblioteca de programas que provee APIs de alto nivel y una interfaz de datos de bajo nivel (Figura 8).

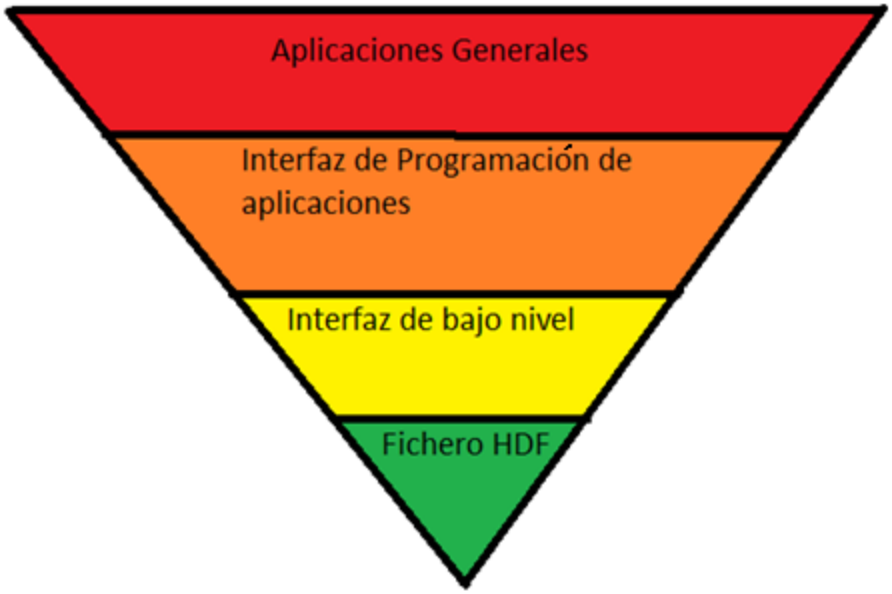


Figura 8: Niveles de interacción de HDF

Aplicaciones generales

En el nivel más alto hay utilidades de línea de comandos de HDF, aplicaciones de NCSA que soportan visualización de datos, análisis y variedad de aplicaciones de terceros desarrolladores.

Existen utilidades de línea de comandos de HDF para:

- Convertir de un formato a otro (por ejemplo, desde y hacia JPEG/HDF)
- Analizar y ver ficheros HDF (siendo *hdp* una de las herramientas más útiles)
- Manipular los ficheros HDF.

De las utilidades HDF, la "*hdp*" es una de las más importantes, su función está dada en que provee información rápida sobre los contenidos y objetos de datos en un archivo HDF (Long et al., 2013), puede listar los contenidos de los archivos HDF en varios niveles con diferentes detalles, además puede también vaciar los datos de uno o más archivos, en formato binario o ASCII.

Compresión de datos

HDF4 (y versiones posteriores) soportan una interfaz de compresión de bajo nivel, la cual permite que cualquier objeto de dato sea comprimido utilizando una variedad de algoritmos. Actualmente solo tres algoritmos de compresión están soportados: Run-lengthEncoding (RLE), Huffman adaptativo, y un codificador de diccionario LZ-77 (el algoritmo de decodificación de gzip)(Yeh et al. 2002). Los planes para algoritmos futuros incluyen un codificador de diccionario Lempel/Ziv-78, un codificador aritmético y un algoritmo rápido de Huffman. HDF4 (y entregas posteriores) soportan compresión de n-bit para SDS, RLE (Run- LengthEncoding), IMCOMP (compresión con perdida), y compresión JPEG para imágenes de puntos.

El archivo HDF

HDF puede almacenar varios tipos de objetos de datos dentro de un archivo, como imágenes de mapa de bits, paletas, texto y estilos de tablas de datos. Cada "objeto" en un archivo HDF tiene una etiqueta predefinida que indica el tipo de datos y un número de referencia que identifica la instancia. Hay un número de etiquetas disponibles para la definición que hace el usuario sobre los tipos de datos. Sin embargo, solamente las personas que tienen acceso al software del usuario pueden acceder adecuadamente a los datos. Los usuarios de HDF no necesitan conocer el formato físico, pues el único método práctico de acceso y manipulación de los datos es mediante interfaces de software. Existen diversas formas de acceder a los datos contenidos en un archivo HDF. Una de ellas es el acceso a nivel binario, conociendo la estructura del archivo. A continuación, se describe la estructura básica del formato HDF(Group & others 2011):

- ***File Header***: Es el primer componente de un archivo HDF que incluye los primeros cuatro bytes en el archivo HDF. El *File Header* es una firma de los archivos HDF, con valores hexadecimales, por ejemplo, 0x0E, 0x03, 0x13, 0x01.
- ***RootGroup***: es una analogía de un directorio de sistema de archivos. Un grupo tiene cero o más objetos y cada objeto debe ser miembro de al menos un grupo. El grupo raíz es un caso especial, pues no puede ser miembro de ningún grupo. Contiene los siguientes campos: *Name* y *Attributes*. Los *Attributes* tienen asociados la siguiente información: *Name*, *Value*, *Type*, que puede tomar los siguientes valores: *string*, *byte* (8-bit), *short* (16-bit), *int*(32-bit), *unsigned byte* (8-bit), *unsigned byte* (16-bit), *unsigned byte* (32-bit),

long(64-bit), *float*, *double*, *object reference*; tipo de almacenamiento: *Max string length* para tipos de datos *string* y *Arraysize* para todos los otros tipos de datos.

- **Named Object**: Hay tres clases de *Named Objects*: *Group*, *Dataset* y *Named Datatype*. Cada uno de esos objetos es miembro de al menos un *Group* y un *Link*.
- ***Group***: similar al *Root Group*.
- ***Dataset***: es un arreglo multidimensional (rectangular) de *Data Elements*. El *DataspaceObject* define la forma de la matriz (número de dimensión y tamaño de cada dimensión). Se asocia con la siguiente información: *name* y *parentgroup*.
- ***Datatype***: describe las características de un único *Data Element*. Los *Datatype* son de dos tipos: *Atomic* y *Composite*. Un *Atomic Datatype* representa un objeto simple como: cadena de caracteres, carácter, tiempo, entero y doble, entre otros. Un *Composite Datatype* se compone de múltiples elementos de tipo *Atomic datatype*, como: *array*, *enumeration*, *variable length* y *compound*. Se asocia con la siguiente información: *Datatype class*, *size* (*bits*) y *byte ordering*.
- ***Dataspace***: describe el diseño de los elementos de una matriz multidimensional. Un *Dataspace* describe el hiper-rectángulo como una lista de dimensiones con actuales y máximos (o ilimitado) tamaños. Se asocia con la dimensión y tamaño de la matriz.
- ***Attribute***: se usan para documentar los objetos. Los atributos de un objeto son el nombre y el dato. Un atributo es similar a un *Dataset* pero con las siguientes limitaciones:
  •Sólo se puede acceder por medio del objeto
  • Los nombre de los atributos tienen significado sólo sin el objeto.
  •Un atributo debe ser un objeto pequeño.
  • Un acceso simple debe leer y escribir los datos de un atributo.
  • Los atributos no tienen atributos.

1.4.2 HDF-EOS

Earth Observing System (EOS) es un programa iniciado por la *Earth Science Enterprise* (ESE) de la NASA para el estudio del cambio climático global. EOS usa principalmente instrumentos aerotransportados para obtener información sobre la tierra, la atmósfera, los océanos y los fenómenos físicos-químicos que ocurren sobre el sistema terrestre (Yang & Di 2004). En la actualidad el programa EOS genera más de 2 terabytes de datos por día. HDF-EOS es un formato

estándar para los datos EOS producidos por la NASA. El formato HDF fue seleccionado como sistema base para almacenar los datos EOS, sin embargo, este carece de mecanismos para guardar información geolocalizada, que es vital para los datos geoespaciales. El proyecto EOS amplió HDF a HDF-EOS para añadirle tres nuevos modelos de datos (*point*, *swath* y *grid*).

De los tres modelos de datos añadidos *swath* y *grid* pueden tener dos o múltiples dimensiones espaciales, mientras que *point* es un modelo de dato discreto. El modelo *grid* es usado para datos georectificados, donde los elementos de datos espaciales (píxeles) son regularmente organizados y todos los píxeles tienen la misma forma y tamaño. Las coordenadas espaciales de cualquier píxel pueden ser obtenidas fácilmente, dando las coordenadas de un píxel de referencia, usualmente la esquina superior o inferior izquierda y el tamaño del píxel. Los datos *grid* pueden ser fácilmente añadidos a otros conjuntos de datos espaciales para ser analizados por un SIG y sensores remotos(Yang & Di 2004) .

El modelo de datos *swath* es para datos que van a ser georectificados, donde diferentes píxeles tienen diferentes formas y tamaños. Sin embargo, sin ser georectificados los datos *swath* no pueden ser combinados directamente con otros datos espaciales georectificados. Actualmente muchos SIG y sensores remotos no son capaces de procesar los HDF-EOS con modelos de datos *swath* debido a no poder geo-rectificar estos datos. Esto sin duda ha significado una limitante para muchos usuarios de EOS, por lo que se están realizando estudios y trabajos con el objetivo de crear herramientas que ayuden a los usuarios a georectificar sus datos.

1.4.3 Network Common Data Form (NetCDF)

El formato de archivo NetCDF (*Network Common Data Form*) fue implementado por el programa Unidata, uno de los ocho programas de la UCAR, y establecido como formato estándar para la comunidad científica para el almacenamiento de todo tipo de datos oceanográficos y atmosféricos desde el año 1989(Borrell González 2012). La fortaleza de NetCDF está dada por la sencillez de su modelo de datos subyacente, su flexibilidad y su eficiente acceso a los datos (Hankin et al. 2010). NetCDF contiene información que ayuda a identificar qué clase de datos se encuentra en el archivo, como tipo de variable, unidad, dimensión y origen, entre otros. NetCDF, a diferencia de otros formatos, no requiere archivos adicionales para su interpretación. En relación con sus características, los datos NetCDF pueden ser (UNIDATA 2012):

- Autodescriptivos: contienen información acerca de los datos.
- Portables: un archivo NetCDF se puede acceder de diferentes maneras.
- Escalables: se puede acceder a un pequeño conjunto de un gran conjunto de datos eficientemente.
- Agregables: se pueden añadir datos a estructuras existentes sin copiar ni redefinir todo el conjunto de datos.
- Compartibles: un proceso escritor y varios lectores pueden acceder a los datos de un NetCDF simultáneamente.
- Archivables: soporta el acceso a versiones antiguas de NetCDF.

Estándares que cumple NetCDF

- OGC (*Open Geospatial Consortium*) persigue acuerdos entre las diferentes empresas del sector que posibiliten la interoperación de sus sistemas de geoprocesamiento y facilitar el intercambio de la información geográfica en beneficio de los usuarios.
- CDI (*Common Data Index*) su objetivo es dar a los usuarios una visualización muy detallada y gran difusión geográfica de información oceanográfica a través de los datos obtenidos de diferentes instituciones.
- ISO 19115 (*International Organization for Standardization*), es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación. Su función principal es buscar la estandarización de normas de productos para organizaciones a nivel internacional. NetCDF tiene una interfaz de programación de aplicaciones (API) bien diseñada que permite el desarrollo de aplicaciones en varios lenguajes de programación. Esta interfaz es usada también para una biblioteca de funciones de acceso a datos, que se almacenan y recuperan en forma de matrices(Rew et al. 2011).

Los valores de las matrices se pueden acceder directamente, sin conocer detalles del almacenamiento de los datos. El desarrollo del formato NetCDF mejora la accesibilidad y la reutilización de los datos en las matrices. Recientemente los desarrolladores de NetCDF y HDF han colaborado para crear un software que combina las mejores características de cada uno, permitiendo así, una mayor interoperabilidad entre las dos comunidades (Hankin et al. 2010). El

software resultante fue NetCDF-4 compatible con versiones anteriores de programas y datos para NetCDF-3, pero mejora el rendimiento e incrementa las capacidades de codificación de colecciones de datos.

El archivo NetCDF

Los archivos NetCDF tienen una estructura interna definida por el CDM (*Common Data Model*). El CDM es un modelo de datos abstractos para conjuntos de datos científicos, este modelo fusiona los modelos NetCDF, OPeNDAP y HDF5 para crear una API común para muchos tipos de datos científicos. El CDM consta de varias capas, donde cada una se va superponiendo en la parte superior de la anterior, para ir añadiendo cada vez una semántica más compleja:

- La capa de **acceso de datos**, se encarga de los datos de lectura y escritura.
- La capa de **sistemas de coordenadas**, identifica las coordenadas de las matrices de datos.
- La capa de **atributo estándar** es la que conoce algunos de los significados que los humanos utilizamos para hacer referencia a los datos científicos tales como unidades, sistemas de coordenadas, topología de datos, etc.…
- La capa de **características de tipo de dato científico**, identifica los tipos de datos añadiendo métodos especializados para cada tipo de datos. Según(Rew et al. 2011)los tipos de datos abstractos más importantes que componen un archivo NetCDF son:
    - **Conjunto de Datos (*Dataset*)**: Puede ser un NetCDF, HDF5, OPeNDAP o cualquier otro tipo de documento al que se pueda acceder a través de la API de NetCDF.
    - **Grupo**: Un grupo contiene variables, dimensiones, atributos, etc.… En definitiva, los grupos son el contenedor de todo aquello que contiene nuestro *dataset* formando un árbol jerárquico. Por eso, como mínimo siempre habrá un grupo definido en nuestro archivo.
    - ***ProtoVariable**:* En los archivos NetCDF una *ProtoVariable* es un contenedor de datos. Está compuesta por varias variables que contienen su información, como por ejemplo el tipo de datos que contiene, su dimensión o dimensiones que definen su matriz y, opcionalmente, se pueden complementar añadiéndole un conjunto de atributos que la describan o definan aún con mayor exactitud.

- **Dimensión**: Una dimensión se utiliza para definir en función de cómo varían los valores de nuestra variable. Puede ser compartida entre las demás variables, lo que proporciona una manera simple y eficaz de asociarlas.
- **Atributo**: Un atributo tiene un nombre y un valor que se utilizan para asociar metadatos con una variable o grupo. Por ejemplo, si se trata de un atributo de una variable estaríamos hablando de unidades o nombre estándar y solamente afectaría a esta variable. Si nos referimos a un atributo de un grupo, este estaría afectando a todo el conjunto de datos y el atributo sería, por ejemplo, para definir qué convención se ha utilizado o el nombre de la organización que ha generado estos datos.
- **Estructura**: Es un tipo de variable que contiene otras variables. De esta manera los datos almacenados en una estructura se encuentran físicamente muy juntos en el disco, lo que hace que sea eficaz la recuperación de estos datos al mismo tiempo.
- ***Array***: Un *array*(o matríz) contiene los datos reales de una variable obtenidos del disco, red, base de datos, etc. Podemos decir que un *array* es el contenedor de las series de datos.

1.4.4 Common Data Format (CDF)

CDF es un formato de datos para almacenar conjuntos de datos multidimensionales. Sus orígenes datan del desarrollo del Sistema de Datos Climáticos de la NASA en el *National Space Science Data Center* (NSSDC). Según(NASA 2013) una de las características más importante de CDF es que puede manipular conjuntos de datos inherentemente dimensionales además de los escalares. Para lograr esto, CDF agrupa los datos por variables cuyos valores son organizados conceptualmente en arreglos. Las variables CDF son nombres genéricos u objetos que representan datos. Por ejemplo, una variable puede ser datos representando una variable independiente, una variable dependiente, tiempo, o un valor de fecha, etc. En otras palabras, las variables no contienen ningún significado aparte de los datos en sí mismos. La dimensionalidad de una variable depende de cómo sean especificados los datos por los usuarios. Para un dato escalar, por ejemplo, el arreglo de valores debe ser 0-dimensional; mientras que para los datos de una imagen sería de 2 dimensiones. CDF permite a los usuarios especificar arreglos hasta de diez

dimensiones. Los arreglos para una variable en particular son llamados "*variable record*". Una colección de arreglos, uno para cada variable es nombrada como un "*CDF record*".

Un CDF puede contener múltiples "CDF records". Estos son útiles para los datos que son observados en diferentes instantes de tiempo. Mientras que las variables son un mecanismo para representar los datos, los atributos en un CDF son un mecanismo para describir el archivo CDF y sus variables. Existen dos tipos de atributos en un CDF: los atributos globales para describir el archivo CDF y los atributos de variables para describir cada variable. Ejemplos de atributos globales, pueden ser: fecha de creación del archivo, autor, documentación de los datos, etc. Ejemplos de atributos de variables son: valor máximo y mínimo, unidades con la cual son almacenados los datos, etc. Es importante destacar que no existe una solo forma correcta de almacenar los datos en un CDF, dado que los usuarios tienen total control de como guardar los datos. Esta es una de las ventajas de CDF.

Los datos pueden ser organizados de cualquier forma según crea el usuario. CDF es también una biblioteca flexible y extensible que brinda muchas facilidades para la creación y el acceso a un CDF (NASA 2013). Esta permite crear de dos formas los formatos de archivos para almacenar los datos y metadatos. El primero es el formato tradicional de multi-archivos CDF, de esta manera se crea un archivo ".CDF" con toda la información y metadatos, además un archivo para cada variable v#, donde # es el número de la variable. La segunda opción es crear un único archivo ".cdf" que contiene toda la información, metadatos y los valores de los datos para cada variable en el CDF. Ambos formatos permiten el acceso directo a los datos. La ventaja de un único archivo es que minimiza el número de archivos a manejar y facilita su distribución a través de redes. Esta biblioteca también permite la compresión de datos, pero solo para un CDF creado a partir de un único archivo.

1.4.5 GML

Geography Markup Language(GML) es un estándar de codificación basado en XML para información geográfica, que desarrolló el Open GIS Consortium (OGC 2012; W3C 2012). GML sirve para modelar, transportar y almacenar información geográfica, relacionada con la posición, localización y extensión, entre otros; incluye propiedades espaciales y no espaciales de las entidades geográficas(OGC 2012).

Debido a que GML se basa en XML, éste hereda todas las ventajas de XML: se puede manipular con todas las herramientas estándar de XML y ambos son llamados lenguajes de marcado "*markup*" o metalenguajes. GML, al igual que XML, separa el contenido de la presentación y se enfoca sólo en la captura del contenido geográfico. Así, consigue ser una herramienta para la descripción de datos geográficos y la administración de documentos que contienen información geográfica (OGC 2012).

GML describe el mundo en términos de entidades geográficas llamadas *features*. Un *feature* es una lista de propiedades y su geometría geográfica. Las propiedades son, usualmente, el nombre, tipo, valor y descripción. Las geometrías geográficas son los elementos geográficos básicos: puntos, líneas, curvas, superficies y polígonos (W3C 2012).

El formato GML requiere un esquema de aplicación que representa un esquema XML escrito acorde con los roles de GML y define un vocabulario de objetos geográficos para un dominio particular. GML es un estándar de codificación basado en XML para información geográfica por lo que algunos de sus componentes se explicaron con anterioridad en 1.2.4. GML utiliza el prefijo "gml:" en las líneas de las etiquetas para identificar que es un documento de tipo GML. A continuación, se identifica la estructura básica del formato GML para datos espacio-temporales(OGC 2012):

- *Namespace*: al igual que el formato XML es un contenedor abstracto del grupo de datos, el *namespace* se asocia con una serie de atributos de un mismo tipo de elementos representados en los esquemas GML. Los esquemas GML se deben importar y describir en el archivo GML, se consideran principalmente los siguientes:
    - *Feature*: para modelar *features* gráficos.
    - *basicType, measure* y *value Object:* para usar unidades de medida.
    - *GeometryBasicOd1d* y *geometryBasic2d*: para geometrías simples de una o dos dimensiones.
    - *Topology*: para propiedades topológicas.
    - *CoordinateReferenceSystem*: para construir un diccionario de sistemas de coordenadas de referencia.
    - *Temporal*: para propiedades dependientes de tiempo o *features* dinámicos.

    - *Coverage*: para construir coberturas, ej. imágenes de sensores remotos, fotografías aéreas y modelos digitales de elevación, entre otras.

- *Header:* es un método que se usa para escribir la declaración de XML. El método entrega el número de la versión y el tipo de codificación como parámetros.
- *Annotation*: es una nota que se hace sobre un dato o información.
- Tipos de elementos: al igual que el formato XML, un elemento puede ser de dos tipos: *simpleType* y *complexType*, pero la estructura se enfoca más en la representación de elementos geográficos. El ejemplo siguiente se refiere a un elemento *complexType*:

  <complexType name="LanguageStringType">
  <simpleContent>
  <extension base="xs: string">
  <attribute ref="xml: lang"/>
  </extension>
  </simpleContent>
  </complexType>
- Nombre de las cadenas localizables (*Name*): para el caso del ejemplo anterior *LanguageStringType* proporciona el tipo base para el texto lingüístico, para su uso dentro de los esquemas GML y en esquemas de aplicación GML.
- *Element*: representa el elemento geográfico que contiene. Su representación por medio del *name* y *type* (*ej. string, boolean, float, date*).
- *Data*: contiene información relacionada con la observación. Internamente se asocia con etiquetas de los datos.
- *DateCreated:* representa la fecha y en algunos casos la hora de toma de los datos. Ejemplo *22/11/2007, 20/11/2007, 30/11/2007,* entre otros. Algunas etiquetas utilizadas para el tiempo son: *date, dateTime, duration, gDay (DD), gMonth (MM), gMonthDay (MM-DD), gYear (YYYY), gYearMonth (YYYY-MM)* y *time.*

1.4.6 KML

Keyhole Markup Language (KML) es un lenguaje de marcado descriptivo, basado en la sintaxis del formato XML, para representar datos geográficos en tres dimensiones. KML permite describir y almacenar información geográfica (puntos, líneas, superficies, polígonos) y utiliza

como referencia el estándar GML. KML permite la representación de relaciones topológicas en modelos geográficos, optimiza la descripción de elemento y describe la información geográfica mediante etiquetas con elementos anidados (Ying-jun, 2009).

KML utiliza el prefijo "kml:" en las líneas de las etiquetas para identificar que es un documento de tipo kml. A continuación se especifica la estructura básica del formato kml para datos espacio temporales (OGC 2012).

- *Namespace*: al igual que el formato XML es un contenedor abstracto del grupo de datos, el *namespace* se asocia con una serie de atributos de un mismo tipo de elementos representados en los esquemas KML.
- *Header*: es un método que se usa para escribir la declaración de XML. El método entrega el número de la versión y el tipo de codificación como parámetros.
- *Name*: representa el nombre de la variable o elemento a representar.
- Tipos de elementos: al igual que el formato XML, un elemento puede ser de dos tipos *simpleType* y *complexType*, pero la estructura se enfoca más en la representación de elementos geográficos.
- Objetos de tipo arreglo de elementos: <*SimpleArrayData*> la representación de los datos puede presentarse por medio de arreglos como vectores o matrices, para ello se requiere el *name* de arreglo y el *type* (*ej. string, boolean, float y date*).
- Objetos de indicación geográfica: son estructuras jerárquicas de etiquetas que representan los elementos de los objetos geográficos. Por ejemplo: *Document, Placemark, altitudeMode, LinearRing, coordinates y date,* entre otras.

1.4.7 Excel

Excel es un programa informático desarrollado y distribuido por *Microsoft Corp*. Se trata de un software que permite realizar tareas contables y financieras, gracias a sus aplicaciones para crear y trabajar con hojas de cálculo. La primera incursión de Microsoft con las hojas de cálculo (que permiten manipular datos numéricos en tablas formadas por la unión de filas y columnas) tuvo lugar en 1982, con la presentación de Multiplan. Tres años más tarde llegaría la primera versión de Excel.

Tipos de datos

Excel nos permite trabajar con distintos tipos de datos, como números, fechas y texto, entre otros. Si bien al ingresarlos en las celdas, estos son detectados de manera automática, el programa no siempre interpreta de modo correcto lo que deseamos, y por eso es necesario saber cómo lograr que cada dato se corresponda con su tipo específico. A continuación, centraremos nuestra atención en identificar tres tipos de datos principales:

Números

El tipo de datos numérico es el que más utilizaremos en nuestras plantillas de cálculo. Al ingresar un número en una celda, el programa reconoce el formato numérico y el valor aparece alineado a la derecha. Los principales aspectos que debemos tener en cuenta al ingresar números son los siguientes:

•Se admite como número a los caracteres del 0 al 9, la coma decimal (,) y el separador de miles (.). Las funciones de estos últimos elementos pueden variar según la configuración regional del sistema operativo de la computadora en la que se trabaja.

•Se utiliza el punto (.) del teclado numérico, este se toma como separador decimal, al igual que la coma (,) del teclado alfanumérico.

•El punto (.) que se encuentra en el teclado alfanumérico será considerado como separador de miles.

Texto

El tipo de datos texto abarca las cadenas de caracteres alfanuméricos, es decir, conjuntos de letras, símbolos y números. Se utiliza, principalmente, para escribir nombres, rotular información y describir características, entre otras posibilidades. Cuando se ingresa texto en una celda, Excel reconocerá de manera automática este tipo de dato y, de manera predeterminada, lo alineará a la izquierda, al igual que si ingresamos una combinación de números y letras más símbolos. Esto puede cambiar si el texto comienza con el signo igual (=) o con el signo más (+) o menos (-). El programa interpreta que lo que sigue es una fórmula de cálculo; por lo tanto, si queremos que sea interpretado como texto, debemos anteponer el carácter comilla simple (‘).

Fecha y hora

Excel utiliza como elementos separadores de fecha la barra (/) y el guion (-). Al ingresar dos o tres números separados con alguno de estos caracteres, el programa los interpreta como fechas. Por ejemplo, si ingresamos 3/5, aparecerá 03-may del año en curso. En el caso de que sea 25-2-13, veremos 25/02/2013. El formato de fecha que se interpreta de modo predeterminado depende de la configuración regional establecida en el sistema operativo. La configuración regional por defecto, en la mayoría de los países de habla hispana, es dd/mm/aaaa (día/mes/año); y en los Estados Unidos es mm/dd/aaaa (mes/día/año). Además, una fecha como 15/2/2013 tendrá formato de Fecha corta, mientras que viernes, 15 de febrero de 2013 tendrá formato de Fecha larga. Ahora, si queremos ingresar un dato en formato de hora, lo hacemos escribiendo dos números (hora y minutos) separados por dos puntos (:). Así se interpretará el formato de hora hh:mm (hora: minuto). En cambio, si ingresamos tres números separados con dos puntos (:) se toman como hh:mm:ss (hora: minuto: segundo).

1.5 Conclusiones parciales

Debido a la necesidad de determinadas comunidades científicas de crear formatos de datos espacio-temporales existen algunos problemas para su distribución. Entre los principales problemas se encuentra que no todos los formatos se rigen por los mismos principios o estándares de ahí la necesidad de algunas comunidades de usuarios de manipular y convertir datos a formatos que admitan sus aplicaciones.

Entre todos los formatos de datos espacio temporales analizados se seleccionaron HDF y NetCDF como formatos para trabajar con ellos en el módulo SEXTANTE de gvSIG por la amplia librería con la que cuenta para su trabajo en java. Así como el trabajo con datos representados en tablas Excel por su gran utilidad para compartir datos científicos actualmente. Lograr una mayor área de trabajo para la biblioteca SEXTANTE y sus algoritmos, es de vital importancia por el gran uso que tiene esta entre las diferentes comunidades SIG.

2 Algoritmos para la Manipulación de archivos HDF y NetCDF en SEXTANTE

En este capítulo se definen los aspectos y características fundamentales a tener en cuenta para el diseño y la implementación de los algoritmos. Se presentan pasos a seguir para una posterior incorporación de nuevos algoritmos al módulo, así como una descripción de los algoritmos implementados para dar solución al objetivo planteado.

2.1 Diseño de los algoritmos realizados

El lenguaje UML (*Unified Modeling Lenguage*) fue el seleccionado para realizar el diseño de los algoritmos, que tiene como objetivos principales la especificación, visualización, construcción y documentación de los productos de un sistema de software. Este lenguaje es utilizado por el RUP (*Rational Unified Process*) como lenguaje de modelado para lo cual se basa en todos sus tipos de diagramas, que constituyen diferentes vistas del modelo del producto.

Los diagramas UML utilizados en el diseño son: diagramas de casos de uso, diagramas de actividades y el diagrama de clases.

La herramienta utilizada para el modelado de todos los diagramas es el *Visual Parading* para UML versión 10.1.

2.1.1 Análisis de actores y casos de usos

Modelar un sistema desde el punto de vista de un usuario es el trabajo de los casos de uso. El caso de uso es una vía mediante la cual se describe la forma en que lucirá un sistema para los usuarios de la misma. Es una colección de escenarios iniciados por una entidad llamada actor (persona, componente de hardware, etc.). Los algoritmos implementados están destinados a un actor en específico, el cual debe ser un especialista o investigador de cualquier rama que necesite trabajar con los formatos de datos científicos HDF o NetCDF y sean capaces de interpretar de forma correcta los resultados que brindan las herramientas (Figura 9**¡Error! No se encuentra el origen de la referencia.**).

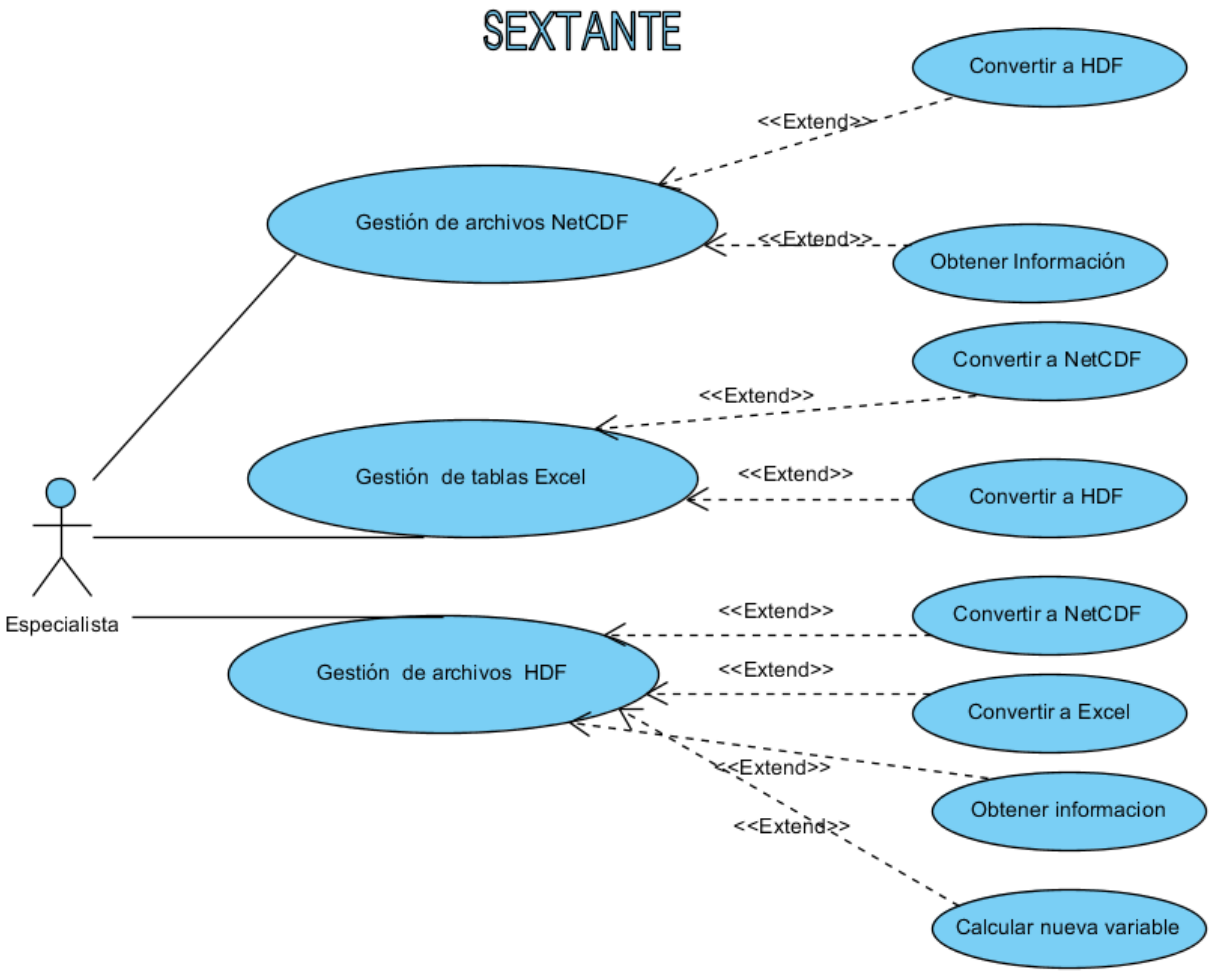


Figura 9: Diagrama de Casos de Usos.

Los casos de usos con los que cuenta la aplicación son gestión de archivos NetCDF, gestión de tablas Excel y gestión de archivos HDF. A continuación, se realizará un análisis de cada caso de uso.

Gestión de archivos NetCDF: Este caso de uso incluye las opciones de Convertir a HDF y obtener información sobre las variables, dimensiones, atributos y tipos de datos de los archivos con los que se trabaja.

Gestión de archivos Excel: En este caso de uso las opciones que se pueden realizar son solo dos; convertir a NetCDF y convertir a HDF.

Gestión de archivos HDF: Cuando trabajamos con archivos HDF podemos convertirlo a archivos NetCDF, convertirlo a tablas Excel, obtener información y calcular nuevas variables a partir de otras existentes para incorporárselas al mismo archivo con el que se trabaja.

2.1.2 Diagrama de actividades

El diagrama de actividades es útil para representar las operaciones de un objeto y los procesos de negocio. Este tipo de diagrama muestra los pasos (actividades), puntos de decisión y bifurcaciones.

Para ganar en claridad y precisión seleccionaremos para cada caso de uso un algoritmo en específico para realizarle el diagrama de actividades.

Convertir de NetCDF a HDF

Para la ejecución de este algoritmo el primer paso es seleccionar el archivo NetCDF de entrada y el HDF de salida por parte del usuario, luego se seleccionan los nombres de todos los *datasets* del NetCDF que será convertido a un solo HDF. Para cada uno de los *datasets* involucrados en el proceso de transformación obtenemos sus dimensiones, creamos un *dataset* en un archivo HDF con igual nombre y con iguales dimensiones que en el archivo NetCDF, procedemos a leer del NetCDF y después escribir en el *dataset* creado en el archivo HDF. Por último cerramos el HDF para que todos los cambios sean registrados correctamente y así obtenemos un archivo con la misma cantidad de *datasets*, el mismo tipo de datos y las mismas dimensiones que el original (Figura 10).

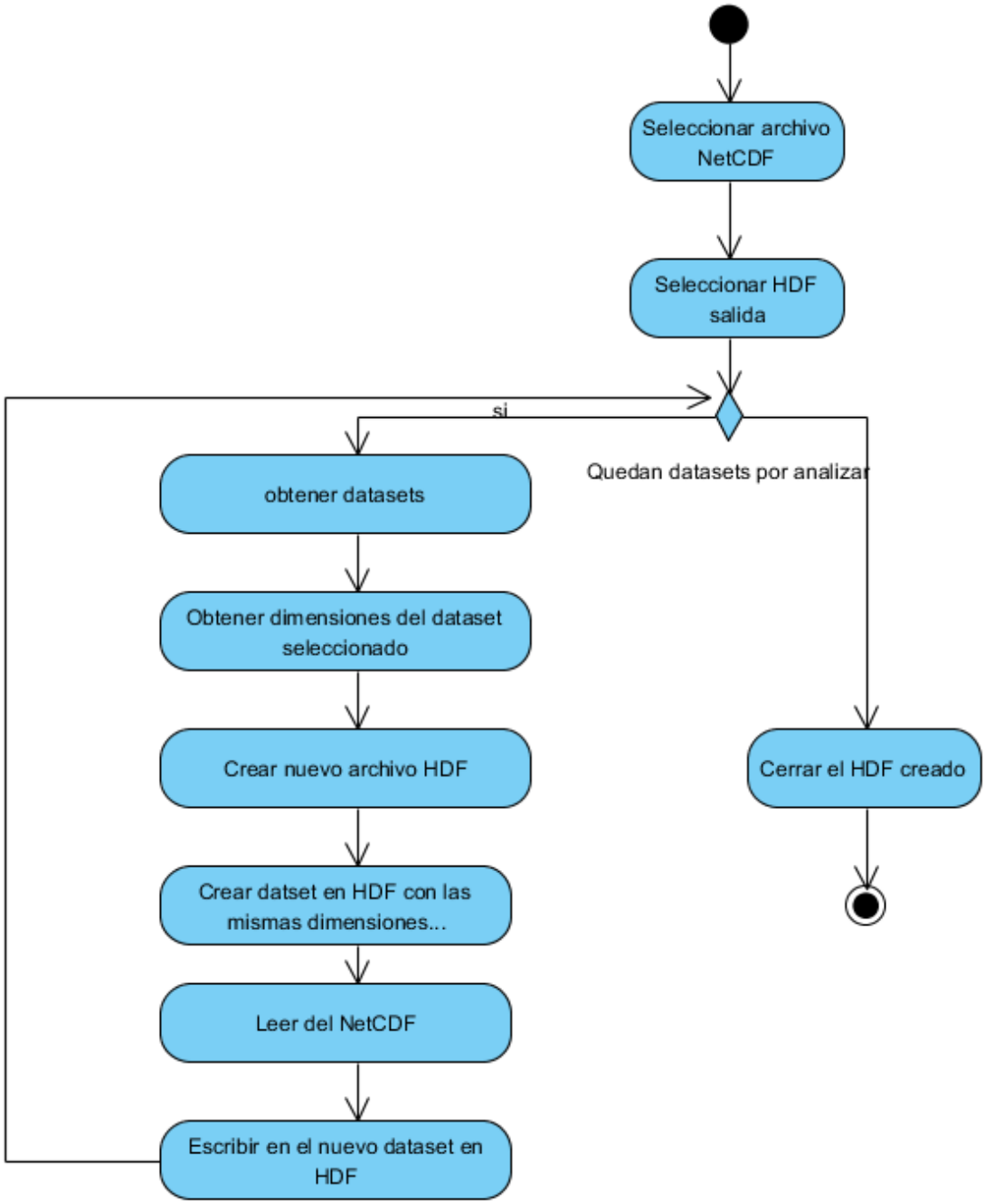


Figura 10: Diagrama de Actividad para el caso de uso Convertir a HDF.

Calcular nueva variable

En este algoritmo el primer paso es seleccionar el HDF, introducir el nombre del nuevo dataset e introducir la fórmula con la que vamos a trabajar. Una vez definidos todos los parámetros de entrada comprobamos que el nombre del nuevo dataset sea correcto, si este nombre no resulta

correcto el programa muestra un mensaje de error y termina, de lo contrario obtenemos los datasets que contiene el HDF y procedemos a comprobar que la fórmula esté correctamente escrita, si la fórmula no está escrita correctamente el programa muestra un mensaje de error. Si la fórmula está bien escrita lo siguiente a realizarse es obtener las dimensiones de alguno de los datasets que contienen el HDF, pues entre las principales características de este tipo de archivos está que todos sus datasets poseen las mismas dimensiones. Después de obtener las dimensiones del dataset creamos dentro del propio archivo HDF un nuevo dataset con el nombre definido por el usuario y las dimensiones seleccionadas anteriormente. La fórmula es ejecutada como siguiente paso y su resultado será escrito en el nuevo dataset creado hasta que este sea llenado completamente; el próximo paso es la creación del dataset extremo dentro del archivo HDF pues este se llama igual que el creado recientemente lo que su nombre culmina con una letra mayúscula E y contiene los valores extremos del dataset (máximo y mínimo).Para concluir con la ejecución del algoritmo el próximo paso es verificar que los dos nuevos datasets han sido incorporados correctamente en el archivo y después cerramos el mismo para que los cambios sean guardados correctamente (Figura 11).

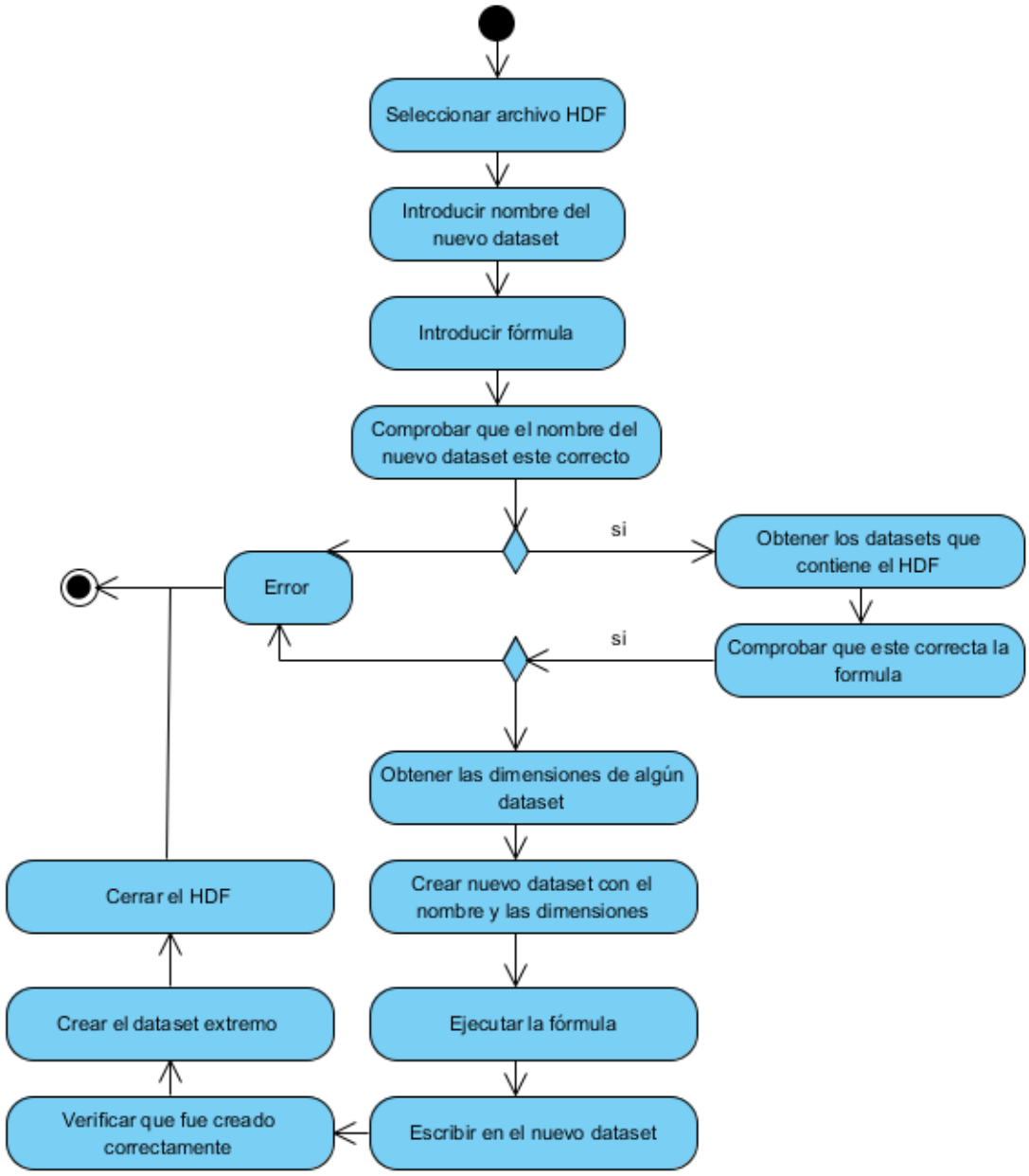


Figura 11: Diagrama de Actividad para el caso de uso calcular nueva variable

Convertir de Excel a NetCDF

Este algoritmo comienza con la selección del archivo Excel y el NetCDF de destino, después de definidos estos parámetros por el usuario se toma el nombre de la hoja Excel que vamos a leer, se toman las dimensiones de la misma y se crea un NetCDF que contendrá un dataset con el mismo nombre de la hoja y las dimensiones de esta. El próximo paso a ejecutarse es leer del

Excel y después esto será escrito en el dataset creado en el NetCDF, por último procedemos a cerrar el archivo NetCDF para que estos cambios sean guardados (Figura 12).

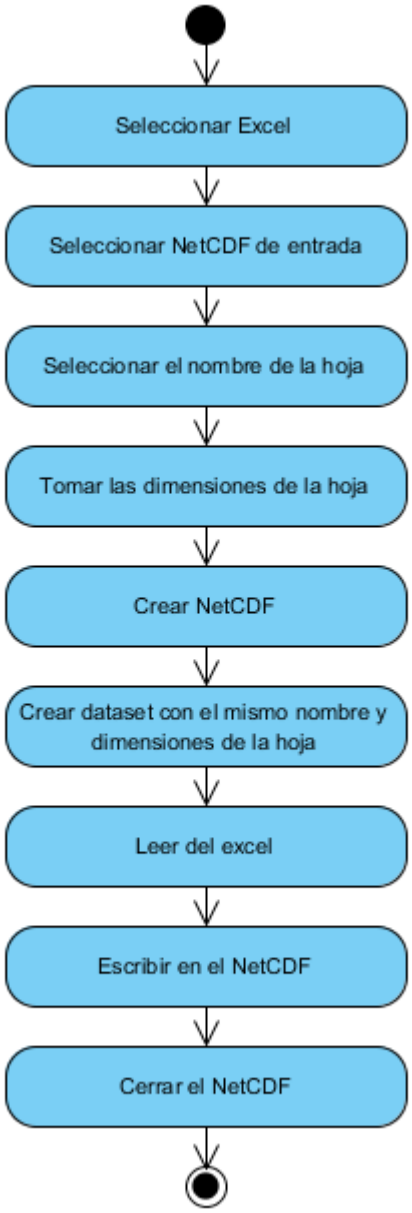


Figura 12: Diagrama de actividad para el caso de uso convertir de Excel a NetCDF.

2.1.3 Diagrama de clases

El diagrama de clases es una forma de estimular al cliente a que diga más respecto a su área y que ponga en evidencia cierta información adicional. Las relaciones muestran como se conectan los términos del vocabulario entre sí para dar una idea de la sección del mundo que se modela.

El módulo de SEXTANTE está dividido en dos paquetes fundamentales: el paquete *Algorithm* donde se encuentran todos los algoritmos implementados y el paquete *FileController* donde están todas las clases encargadas de manipular los archivos HDF, NetCDF y Excel. En la Figura 13**¡Error! No se encuentra el origen de la referencia.**, para ganar en claridad, solo se ha mostrado el diagrama de clases para el algoritmo *NetCDFtoHDF*. De forma similar a las relaciones entre clases que se observan en este diagrama sucede con los demás algoritmos. El paquete *Sextante* es el núcleo de SEXTANTE, en el están implementadas todas las clases de las que dependen los algoritmos incluidos a esta plataforma. Los algoritmos implementados que leen o escriben en archivos NetCDF hacen uso de la clase *NetCDFController* en donde radican todos los métodos necesarios para la manipulación de estos archivos. Para leer o escribir en archivos HDF es necesario utilizar la clase *HDFController* donde existen métodos que ayudan a manipular estos archivos. Así mismo ocurre con los archivos Excel que utilizan la clase *ExcelController*. En el diagrama se puede observar la relación de herencia que existe entre la clase *NetCDFtoHDFAlgorithm* y la clase abstracta *GeoAlgorithm* de SEXTANTE. La relación de herencia está presente también entre las clases *NetCDFtoHDFParametersPanel* donde se define la interfaz visual del algoritmo para ser utilizado desde la Caja de Herramientas y la clase *GeoAlgorithmPareametersPanel* de SEXTANTE. Otra relación que está presente es la de asociación entre las clases *GeoAlgorithm* y la clase *NetCDFtoHDFParametersPanel*. La clase *GeoAlgorithmParametersPanel* utiliza la clase *NetCDFtoHDFAlgorithm* para crear la *GUI* según las especificaciones definidas en el método *defineCharacteristics ()*. Es importante resaltar, que aunque no se defina una *GUI*, SEXTANTE se encarga de definir una *GUI*, la cual utiliza los parámetros que se definen en el método *defineCharacteristics ()*.

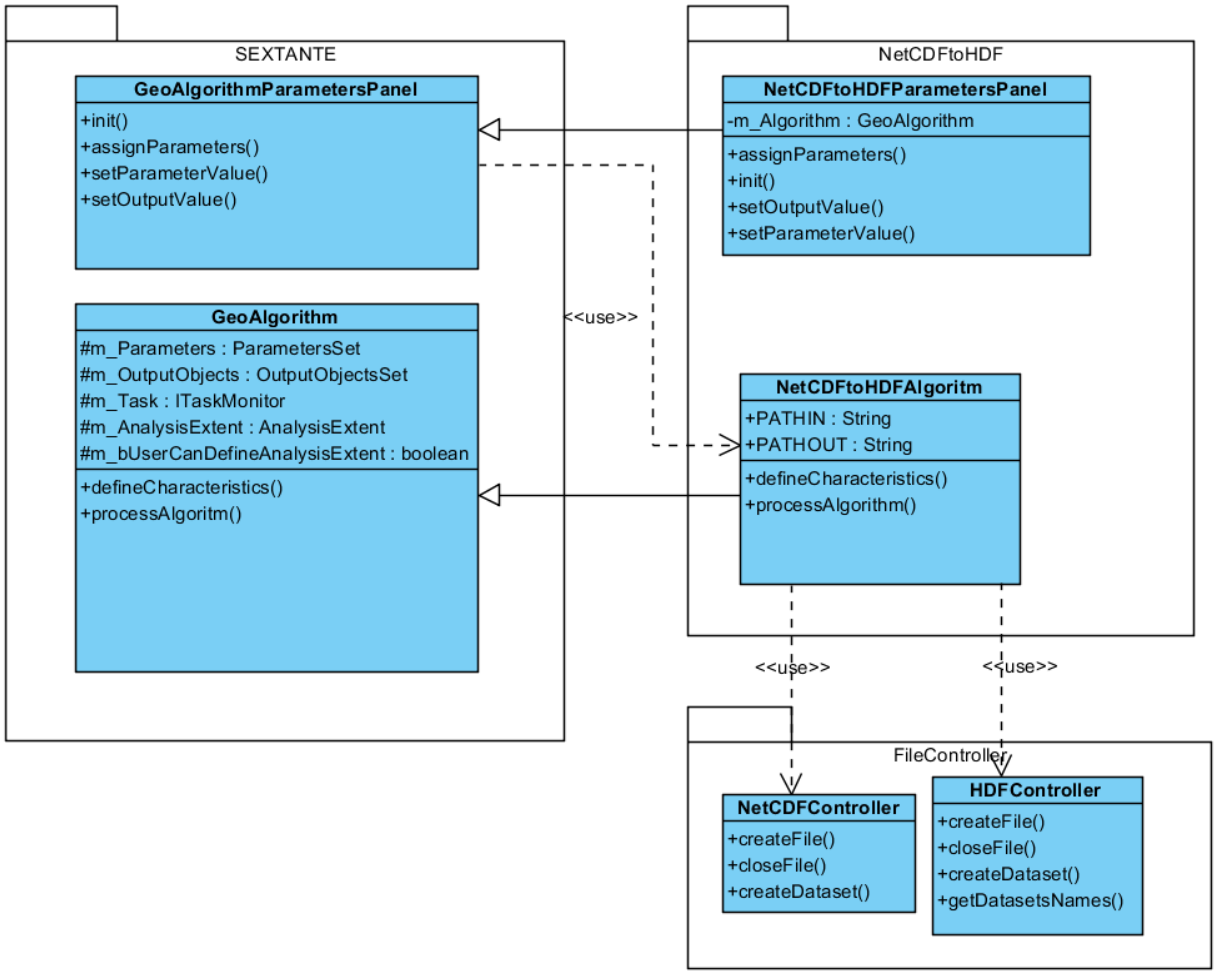


Figura 13: Diagrama de Clases.

2.2 Herramientas utilizadas en la implementación de los algoritmos

Tanto SEXTANTE como gvSIG están implementados en Java, por eso se seleccionó este lenguaje para mantener una mayor compatibilidad. Súmese a eso que se utilizan las bibliotecas *HDF-java* y *NetCDF 4.2* ambas desarrolladas en Java, las cuales tienen múltiples funciones para la manipulación de formatos de datos científicos espacio-temporales. Para la manipulación de archivos con formato Excel se utiliza el conjunto de bibliotecas *poi-3.15* implementadas en Java con varias funcionalidades para trabajar con estos archivos. En el proceso de extracción de la información de los archivos NetCDF y HDF se utiliza la biblioteca *antlr-4.2-complete* que permite la creación de *String Templates* (plantillas) donde escribiremos toda esta información.

Existe un conjunto de Entornos de Desarrollo Integrado (IDE, de sus siglas en inglés) que permiten el desarrollo de proyectos en java. El IDE Eclipse Kepler o 4.3 fue el seleccionado

como ambiente de programación para la realización de los algoritmos por sus facilidades para la depuración de programas y por la estructura de los proyectos de gvSIG que está diseñada para ser compilada con mayor facilidad en este IDE.

Una amplia comunidad de desarrolladores hace posible que regularmente existan versiones de gvSIG disponibles. De las versiones existentes de gvSIG se seleccionó como sistema base la versión 1.12 para trabajar con ella.

2.3 Pasos para agregar un nuevo algoritmo a SEXTANTE

El primer paso para agregar un algoritmo en SEXTANTE es crear un proyecto, una vez creado este proyecto creamos un nuevo paquete que contendrá el algoritmo y todos los archivos necesarios para su implementación. Se recomienda crear un paquete diferente por cada algoritmo para ganar en claridad y organización(Olaya 2011).

Para que nuestro algoritmo sea publicado en SEXTANTE la clase que contendrá el algoritmo que queremos incorporar debe terminar con el sufijo *Algorithm* y debe heredar de la clase GeoAlgorithm que tiene todas las operaciones necesarias para efectuar cualquier tipo de análisis. Además, el archivo *jar* que contiene los algoritmos debe ser nombrado con el prefijo *sextante_*.

Después de heredar de la clase GeoAlgorithm el algoritmo debe implementar los siguientes métodos:

- defineCharacteristics ()
- processAlgorithm ()

El primero de los métodos anteriores debe ser utilizado para definir las características del algoritmo, por lo que este método debe responder las siguientes preguntas:

- ¿Cuáles entradas son necesarias para la ejecución del algoritmo?
- ¿Qué salida generará el algoritmo?
- Características básicas del algoritmo

Este método *defineCharacteristics ()* es el utilizado por SEXTANTE para crear la Interfaz Visual del algoritmo implementado, se utiliza para esto los parámetros de entrada y salida definidos por el desarrollador.

Para implementar el algoritmo con los cálculos y procesos necesarios, esto se realiza en el método *processAlgorithm ()*.

Existen tres tareas fundamentales que deben implementarse para que el algoritmo cumpla con lo deseado y se llegue al objetivo:

- Leer los parámetros de entrada. Los valores deben ser leídos y guardados en variables para su posterior uso.
- Trabajar con las variables.
- En un algoritmo los datos de salida son lo más importante, por lo que los datos de salida deben ser correctamente creados para poder ser utilizados por otros algoritmos de SEXTANTE.

SEXTANTE proporciona un mecanismo sencillo para internacionalizar los algoritmos, solo hay que utilizar el método getText(String) de la clase estática *Sextante* el cual devuelve la clave dada en idioma actual de la aplicación SIG desde la cual se ejecuta SEXTANTE, como pasa con los algoritmos se debe seguir un convenio para que SEXTANTE sea capaz de encontrar los archivos que contienen las traducciones de las claves. Se deben crear archivos *Java properties* con las claves en los diferentes idiomas y deben ser agregados al archivo *jar* y colocados dentro de un directorio nombrado *i18n*.

Personalizar la interfaz visual de los algoritmos

Diseñar la *GUI* no es necesario para la creación de un algoritmo. Una vez creado el algoritmo se puede utilizar desde otro algoritmo mediante programación y también desde cualquiera de los componentes de SEXTANTE tales como la caja de herramienta o el modelador gráfico. En este último caso la interfaz gráfica de usuario se crea automáticamente sobre los parámetros del algoritmo, este tiene disimiles ventajas, pero en ocasiones los elementos de SEXTANTE no son suficientes por lo que hay que realizar una interfaz diferente.

En diversas ocasiones la interfaz de usuario que nos proporciona SEXTANTE puede resultar un poco difícil de comprender y utilizar, si queremos realizar una interfaz de usuario propia para nuestro algoritmo, estos son los pasos a seguir:

- En el mismo paquete debemos agregar una nueva clase con el mismo nombre pero sustituyendo el prefijo *Algorithm* por *ParameterPanel.*
- La nueva clase incorporada debe heredar de la clase *GeoAlgorithmParametersPanel*, que a su vez hereda de la clase *JPanel*. Cuando SEXTANTE crea el diálogo para el algoritmo este busca una clase con el mismo nombre y con el sufijo *ParametersPanel*, si la clase existe y hereda de *GeoAlgorithmParametersPanel* se añadirá el panel representado por esta clase a la pestaña Parámetros. Si no se genera el panel predeterminado.
- Esta clase debe tener un constructor sin argumentos y debe implementar el método init (GeoAlgorithm) el cual será el encargado de crear la *GUI*. Este método va ser llamado por SEXTANTE justo después de instanciar la clase. Otro método que se implementa es el nombrado *boolean assignParameters()*, este método se encarga de asignar los valores entrados por el usuario a los correspondientes parámetros del algoritmo, además debe chequear que sean correctos, de ser así retorna *true* en otro caso retorna *false*.

De esta manera queda modificada la *GUI* para el algoritmo, pero también se pueden implementar los métodos *SetParameterValue(String, String)* y *setOutputValue(String, String)* estos dos métodos se encargan de mostrar los parámetros con los que se ejecutó el algoritmo por última vez, esto obliga a los usuarios a llenar los valores de entrada para el algoritmo.

2.4 Esquemas generales de algunos algoritmos

La Figura 14**¡Error! No se encuentra el origen de la referencia.** muestra el seudocódigo para la realización de uno de los algoritmos, donde se necesita como parámetros el archivo NetCDF de entrada, el archivo HDF de salida, y el nivel de compresión. Durante el proceso de transformación, se le extraen al *dataset* de origen las dimensiones y se crea uno en el nuevo formato con las mismas dimensiones y tipo de datos. Este proceso se repite hasta que concluyan todos los *dataset*. De manera análoga la **¡Error! No se encuentra el origen de la referencia.**muestra un seudocódigo para la realización de otro algoritmo.

```
FIN: ARCHIVO DE ENTRADA
FOUT: ARCHIVO DE SALIDA
N:NIVEL DE COMPRESIÓN
PARA CADA UNO DE LOS DATASETS DE FIN {
SDFIN-TOSDFOUT(FIN , FOUT, N)
    1 T=DATA-TYPE(FIN)
    2 DIM=GET-DIMENSION(FIN)
    3 CREATE-DATASET-VARIABLE(FOUT, T, DIM , N)
    4  NUM-DIM=SIZE(DIM)
    5 READ-BLOCK(NUMDIM , FIN)
    6 WRITE-BLOCK(NUMDIM , FOUT)
}
```

Figura 14: Algoritmo 1.

fIN: Archivo de entrada NetCDF.
fOUT: Archivo de salida HDF.
n: Nivel de compresión.

```
NETCDFCRU-TO-HDF(fIN, fOUT, n)
 1  t ← DATA-TYPE(fIN)
 2  L ← INFORMATION-POINTS(fIN)
 3  time ← TIME(fIN)
 4  dim ← {SIZE(L), time}
 5  CREATE-DATASET-VARIABLE(fOUT, t, dim, n)
 6  minmax ← {MaxValue, MinValue}
 7  for i ← 1 to SIZE(L)
 8       do point ← L[i]
 9          line ← READ-TIME(fIN, point)
10          WRITE-LINE(fOUT, i, line)
11          for j ← 1 to SIZE(line)
12               do minmax[0] ← MIN(minmax[0], line[j])
13                  minmax[1] ← MAX(minmax[1], line[j])
```

Figura 15: Algoritmo 2.

2.5 Principales funcionalidades del módulo.

El módulo SEXTANTE de gvSIG cuenta actualmente con 19 algoritmos a los cuales se le incorporan 8 nuevos algoritmos.

A continuación, se brinda una breve descripción de cada uno de los algoritmos implementados.

HDFFieldCalculator: Este algoritmo puede ser ejecutado desde que iniciamos SEXTANTE, en este algoritmo los parámetros de entrada son el archivo HDF, nombre del nuevo *dataset*, diferente a los existentes y la fórmula a aplicar. De esta manera se crea un nuevo *dataset* dentro del fichero HDF de entrada con la misma estructura de los demás *datasets*.

```
m_Parameters.addFilepath(PATHIN, Sextante.getText("hdf_in"), false, true, ".h5");
m_Parameters.addString(DATASET, Sextante.getText("new_dataset"));
m_Parameters.addString(FORMULA, Sextante.getText("formula"));
```

New NetCDF to HDF: Este algoritmo también puede ser ejecutado desde que iniciamos SEXTANTE sin la necesidad de tener ningún objeto cargado previamente en gvSIG. Los parámetros necesarios para ejecutar el algoritmo son el archivo NetCDF que vamos a transformar, el archivo HDF de salida y el nivel de compresión con el que vamos a guardar el archivo HDF. Es importante resaltar en este algoritmo que el nivel de compresión tiene gran importancia para guardar archivos HDF de gran tamaño, aunque este proceso demora significativamente la ejecución del algoritmo.

```
m_Parameters.addFilepath(PATHIN, Sextante.getText("netcdf_in"),false, true, new
String[] { "nc" });
m_Parameters.addFilepath(PATHOUT, Sextante.getText("hdf_out"),false, false, "h5");
m_Parameters.addString(COMPRES, "");
```

New HDF to NetCDF: Al igual que el anterior este algoritmo se puede ejecutar directamente desde que iniciamos SEXTANTE , solo tenemos que especificar los parámetros necesarios para la ejecución del algoritmo tales como el archivo HDF de entrada y el archivo NetCDF de salida, obteniendo un NetCDF que contiene todos los *datasets* del fichero de entrada.

```
m_Parameters.addFilepath(PATHIN, Sextante.getText("hdf_in"), false,true, new String[]
{ ".h5" });
m_Parameters.addFilepath(PATHOUT, Sextante.getText("netcdf_out"),false, false, "nc");
```

HDF Information: Este algoritmo para su ejecución solo necesita del archivo HDF de entrada y la dirección donde el usuario desea guardar el archivo de salida, en el cual se guardará toda la información referente al archivo HDF.

```
m_Parameters.addFilepath(PATHIN, Sextante.getText("hdf_in"), false,true, ".h5");
m_Parameters.addFilepath(PATHOUT, Sextante.getText("txt_out"), false,true, ".txt");
```

NetCDF Information: Al igual que el anterior algoritmo este también necesita el archivo NetCDF de entrada y la ubicación del archivo de salida definido por el usuario donde se guarda toda la información del NetCDF.

```
m_Parameters.addFilepath(PATHIN, Sextante.getText("netcdf_in"),false, true, new
String[] { "nc" });
m_Parameters.addFilepath(PATHOUT, Sextante.getText("txt_out"), false,true, ".txt");
```

Excel Table to NetCDF: Este algoritmo toma los datos guardados en una tabla Excel y crea un archivo NetCDF con las mismas dimensiones que el Excel de entrada.

```
m_Parameters.addFilepath(PATHIN, Sextante.getText("excel_in"), false,true, new
String[]{".xlsx",".xls"});
 m_Parameters.addFilepath(PATHOUT, Sextante.getText("NetCDF_out"), false, true,
".nc");
```

Excel Table to HDF: Este algoritmo toma los datos guardados en una tabla Excel y crea un archivo HDF con las mismas dimensiones que el Excel de entrada.

```
m_Parameters.addFilepath(PATHIN, Sextante.getText("excel_in"), false, true, new
String[]{".xlsx",".xls"});
 m_Parameters.addFilepath(PATHOUT, Sextante.getText("HDF_out"), false, true, ".nc");
```

HDF Dataset to Excel: Este algoritmo transforma en una tabla Excel un *dataset* guardado en un archivo HDF. Los parámetros necesarios para la ejecución del algoritmo son el archivo HDF de entrada, el archivo Excel de salida y el nombre del *dataset* que voy a transformar.

```
m_Parameters.addFilepath(PATHIN, Sextante.getText("hdf_in"), false, true, ".h5");
m_Parameters.addFilepath(PATHOUT, Sextante.getText("excel_out"), false, true,
".xlsx");
m_Parameters.addString(DATASET, Sextante.getText("new_dataset"));
```

2.6 Conclusiones Parciales

La realización de algoritmos para el trabajo con formatos de datos científicos contribuye al análisis de grandes volúmenes de información mediante la conversión de estos formatos a NetCDF, HDF y a tablas Excel. En este capítulo se presentó el análisis de actores y casos de uso del sistema, se mostró la arquitectura general de sistema realizado. Se mostraron las acciones o pasos a seguir para la incorporación de nuevos algoritmos al módulo de SEXTANTE.

3 Presentación, uso y validación del sistema

En este capítulo se realiza una presentación para el usuario respecto a las facilidades y funcionalidades del módulo. Se muestra un manual de usuario con las principales funcionalidades de los algoritmos, un caso de estudio sobre datos climáticos de la República de Cuba y una breve validación del sistema.

3.1 Manual de usuario

Este manual de usuario contribuye a la correcta utilización de los algoritmos con los que cuenta el módulo para la transformación de formatos de datos científicos en gvSIG.

3.1.1 Acceso al módulo de SEXTANTE y a los algoritmos

Este paso se puede realizar de dos vías diferentes

- Como primera opción debemos ir al menú SEXTANTE / Caja de herramientas (Figura 16).

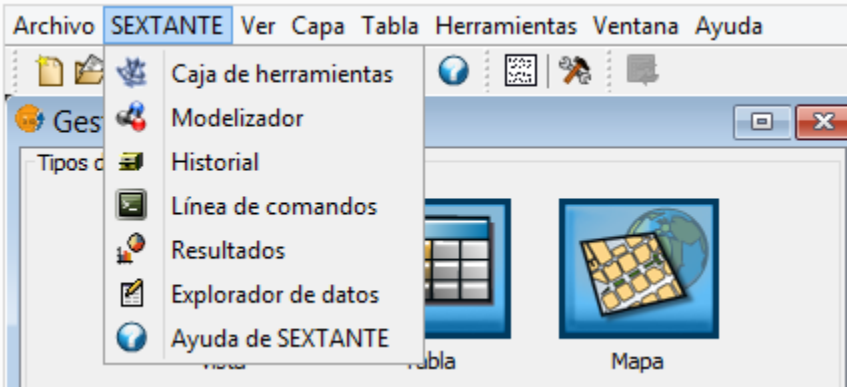


Figura 16: Primera vía de acceso a SEXTANTE

- La segunda opción es la más utilizada por todos y es ir directamente al icono de la Caja de herramientas que se muestra como un menú en la ventana de inicio de gvSIG (Figura 17).

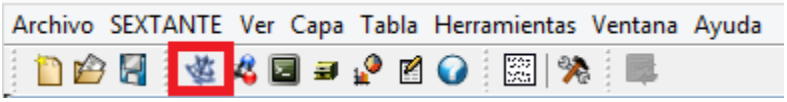


Figura 17: Segunda vía de acceso a SEXTANTE.

Después de haber utilizado cualquiera de las dos vías anteriores es necesario proceder a seleccionar la opción con la que vamos a trabajar, dado que SEXTANTE es una biblioteca muy

amplia con muchas funcionalidades, en nuestro caso accederemos a Herramientas para Formatos de Datos Científicos (Figura 18).

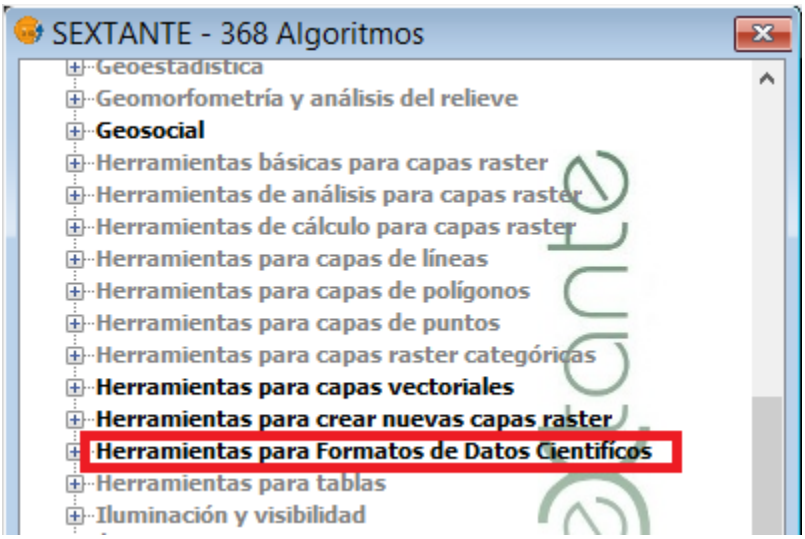


Figura 18: Herramientas para Formatos de Datos Cietíficos en SEXTANTE.

3.1.2 Elementos generales para la ejecución de los algoritmos

Para empezar con la utilización de los algoritmos es importante tener en cuenta que cada algoritmo que usted seleccione muestra una interfaz propia que puede guardar similitud con otras vistas en SEXTANTE. En la **¡Error! No se encuentra el origen de la referencia.** se muestran los pasos necesarios para ejecución del algoritmo HDFFieldCalculator, por ser el que muestra una *GUI* de mayor complejidad para su entendimiento. Se han señalado los principales elementos con los que puede contar una *GUI* para un determinado algoritmo.

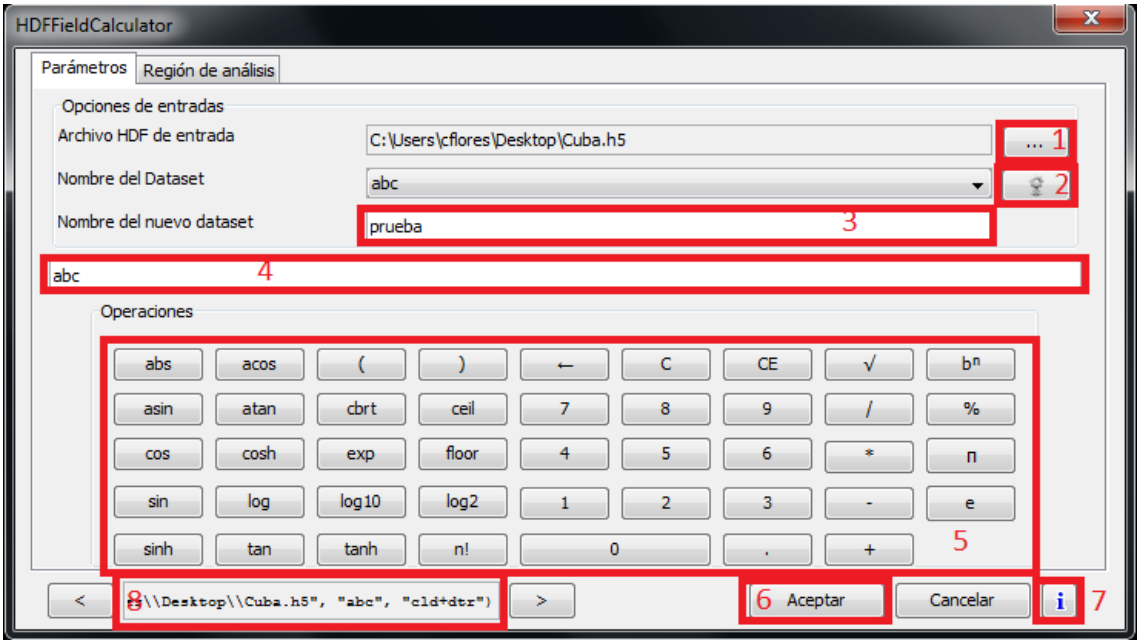


Figura 19: Elementos significativos de los algoritmos.

1. Opción para incorporar el archivo de entrada
2. Botón para incorporar el *dataset* o los *datasets* que intervienen en la fórmula, estos se incorporan de uno a la vez
3. Área para escribir el nombre del nuevo *dataset* que se incorpora al archivo
4. Área donde se incorpora automáticamente o se puede introducir manualmente la fórmula que se va aplicar para guardar el resultado en el nuevo *dataset*
5. Operaciones matemáticas aplicables para la formula
6. Botón de aceptar
7. Información referente al algoritmo, muestra los argumentos para ejecutar el algoritmo desde la línea de comandos
8. Información referente a la última ejecución del algoritmo

Es importante saber que una de las características de gvSIG es que siempre que usted vaya a utilizar un algoritmo este aparecerá automáticamente con los últimos parámetros con los que se corrió. En la **¡Error! No se encuentra el origen de la referencia.**se muestra otro de los algoritmos con menos nivel de detalles, pues casi todas las *GUI* con las que cuentan los algoritmos tienen la misma similitud, por lo que será muy fácil entender los componentes que conforman la ventana.

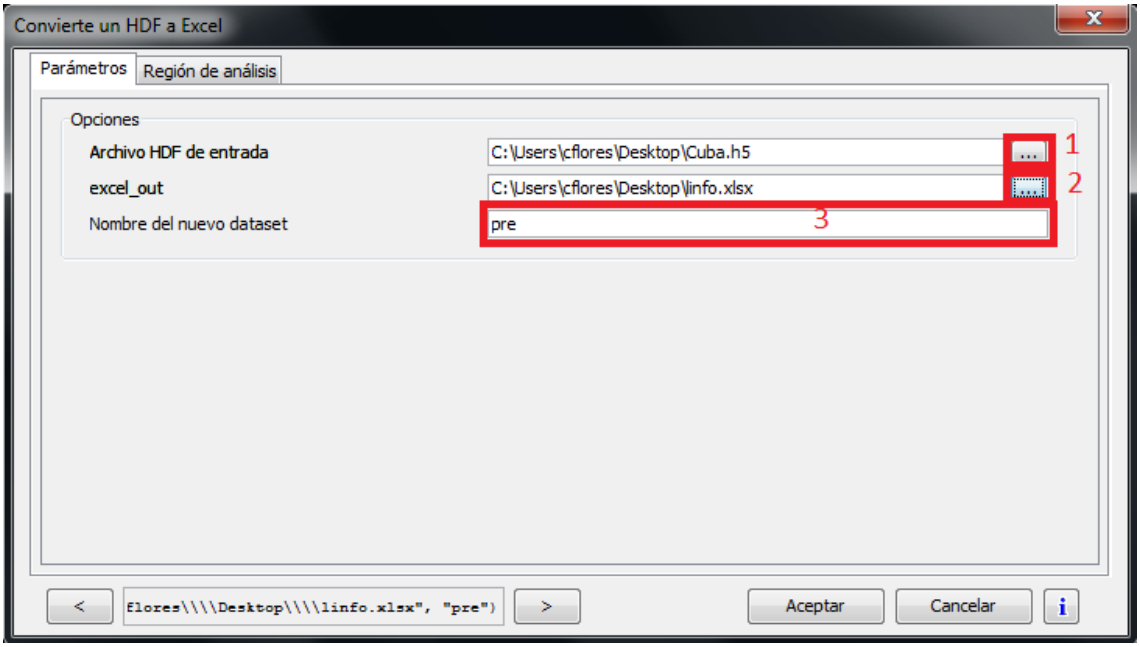


Figura 20: Otros elementos importantes de los algoritmos.

3.2 Caso de estudio: Datos climáticos sobre la República de Cuba.

En este epígrafe utilizaremos las herramientas implementadas para calcular una nueva variable dentro de un archivo HDF, crearemos a partir de este un archivo NetCDF.

3.2.1 Procedencia de los datos.

Los datos utilizados para este caso de estudio fueron tomados por el instituto *Climatic Research Unit* (CRU) de la universidad del Este de Anglia en Reino Unido. Este instituto se ha reafirmado como uno de los líderes mundiales en el estudio del cambio climático.

El objetivo fundamental de la unidad de investigaciones climáticas es contribuir al mejoramiento de los científicos en tres áreas fundamentalmente:

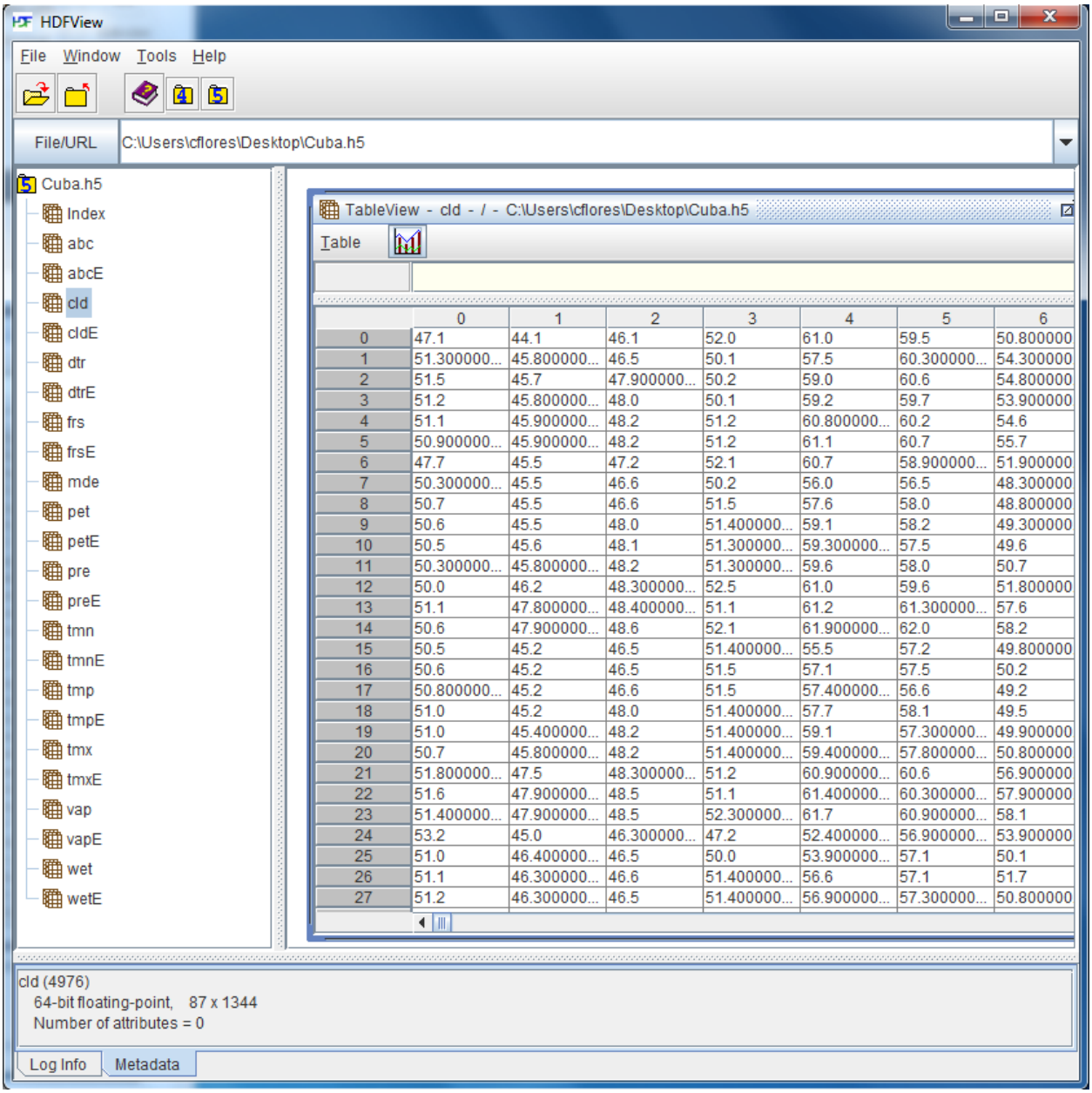
- Historia del clima pasado y su impacto sobre la humanidad
- El curso y las causas del cambio climático durante el siglo actual
- Perspectivas para el futuro

3.2.2 Características principales de los datos.

Varios son los *datasets* con los que cuenta el CRU, los cuales poseen disimiles características. Para este caso de estudio se cuenta con el *dataset* Cuba que almacena los datos en cuadricula de alta resolución. Este tipo de *dataset* puede estar compuestos por varias variables, las cuales permiten comparar las variaciones del clima con las variaciones de otros fenómenos (Figura 21). Entre las variables que podemos encontrar dentro del *dataset* Cuba se encuentran:

- cld: Cobertura nubosa
- dtr: Rango de temperatura diurna
- frs: Frecuencia de días con escarcha
- pet: Potencial de precipitación
- pre: Precipitación
- tmp: Temperatura media diaria
- tmn: Promedio mensual de la temperatura mínima diaria
- tmx: Promedio mensual de la temperatura máxima diaria
- vap: Presión de vapor
- wet: Frecuencia de días húmedos

Como se puede observar este *dataset* contiene diez variables climáticas. Para un mejor análisis en la **¡Error! No se encuentra el origen de la referencia.**9 se ha mostrado todas las variables que contiene el archivo Cuba.h5.



	0	1	2	3	4	5	6
0	47.1	44.1	46.1	52.0	61.0	59.5	50.800000
1	51.300000...	45.800000...	46.5	50.1	57.5	60.300000...	54.300000
2	51.5	45.7	47.900000...	50.2	59.0	60.6	54.800000
3	51.2	45.800000...	48.0	50.1	59.2	59.7	53.900000
4	51.1	45.900000...	48.2	51.2	60.800000...	60.2	54.6
5	50.900000...	45.900000...	48.2	51.2	61.1	60.7	55.7
6	47.7	45.5	47.2	52.1	60.7	58.900000...	51.900000
7	50.300000...	45.5	46.6	50.2	56.0	56.5	48.300000
8	50.7	45.5	46.6	51.5	57.6	58.0	48.800000
9	50.6	45.5	48.0	51.400000...	59.1	58.2	49.300000
10	50.5	45.6	48.1	51.300000...	59.300000...	57.5	49.6
11	50.300000...	45.800000...	48.2	51.300000...	59.6	58.0	50.7
12	50.0	46.2	48.300000...	52.5	61.0	59.6	51.800000
13	51.1	47.800000...	48.400000...	51.1	61.2	61.300000...	57.6
14	50.6	47.900000...	48.6	52.1	61.900000...	62.0	58.2
15	50.5	45.2	46.5	51.400000...	55.5	57.2	49.800000
16	50.6	45.2	46.5	51.5	57.1	57.5	50.2
17	50.800000...	45.2	46.6	51.5	57.400000...	56.6	49.2
18	51.0	45.2	48.0	51.400000...	57.7	58.1	49.5
19	51.0	45.400000...	48.2	51.400000...	59.1	57.300000...	49.900000
20	50.7	45.800000...	48.2	51.400000...	59.400000...	57.800000...	50.800000
21	51.800000...	47.5	48.300000...	51.2	60.900000...	60.6	56.900000
22	51.6	47.900000...	48.5	51.1	61.400000...	60.300000...	57.900000
23	51.400000...	47.900000...	48.5	52.300000...	61.7	60.900000...	58.1
24	53.2	45.0	46.300000...	47.2	52.400000...	56.900000...	53.900000
25	51.0	46.400000...	46.5	50.0	53.900000...	57.1	50.1
26	51.1	46.300000...	46.6	51.400000...	56.6	57.1	51.7
27	51.2	46.300000...	46.5	51.400000...	56.900000...	57.300000...	50.800000

Figura 21: Estructura de la variable climática cld en HDF.

La estructura del archivo HDF es la siguiente: un *dataset* índice que tiene las mismas dimensiones que el mapa de fondo o modelo digital de elevaciones (matriz de M fila por N columnas) con el número de celda si esta contiene información o cero si la celda no tiene información. Tiene un *dataset* de P fila por Q columnas para cada variable V que contiene el archivo HDF, donde P es el número de celdas con información y Q es el número de registros temporales para cada variable. Además, otro *dataset* de dos filas por una columna con el fin de almacenar los valores extremos de los datos por cada variable. En la Figura 21 se puede ver el

resultado del flujo de trabajo y el archivo con la estructura descrita anteriormente. Para este caso M es igual 10 y N a 24 mientras que P es igual 87 que son las celdas que almacenan información y Q igual 1344 momentos de tiempos.

3.2.3 Ejecución del flujo de trabajo

Después de describirse los datos con los que se realiza este caso de estudio, veremos cómo se han utilizado los algoritmos incorporados al módulo SEXTANTE de gvSIG para la conversión de formatos de datos científicos.

Generación de un archivo NetCDF

Para la generación de este archivo utilizaremos el algoritmo HDF to NetCDF con el cual generaremos un archivo NetCDF con la misma cantidad de variables que conforman el archivo HDF, además de poseer la misma estructura y dimensiones de las variables originales (Figura 22).

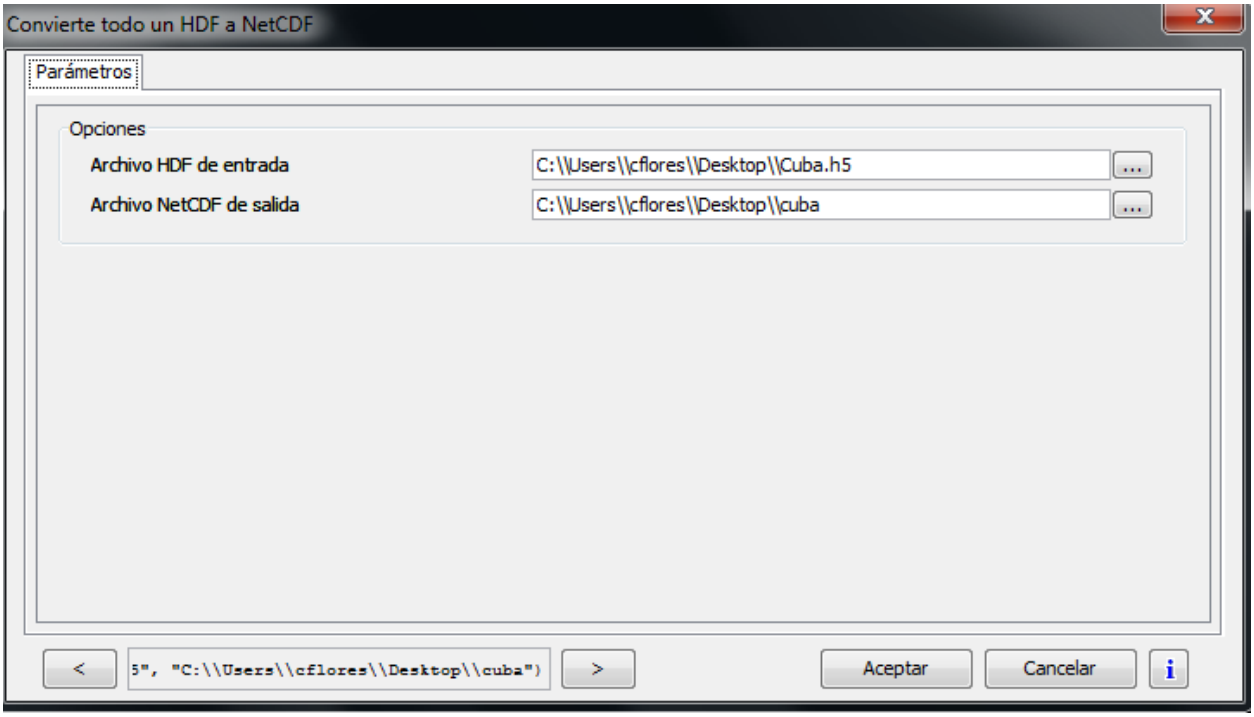


Figura 22: Parámetros de entrada para el algoritmo.

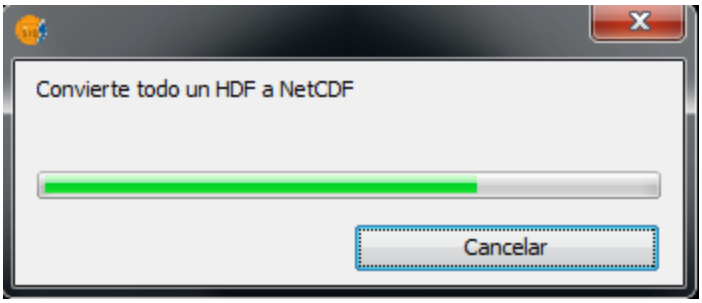


Figura 23: Barra de progreso que muestra la ejecución del algoritmo.

La ejecución del algoritmo se realizó en una computadora con las siguientes características: Procesados Intel(R) Core(TM)2 Duo CPU E7300 @ 2.66GHz y memoria del sistema 4096 MB. El tiempo de ejecución para el algoritmo fue de aproximadamente 48 segundos para este archivo HDF, valorándose como buenos los resultados obtenidos teniendo en cuenta la cantidad de información procesada. En cuanto a la capacidad ocupada en disco se redujo de 14.5 a 9.81 MB.

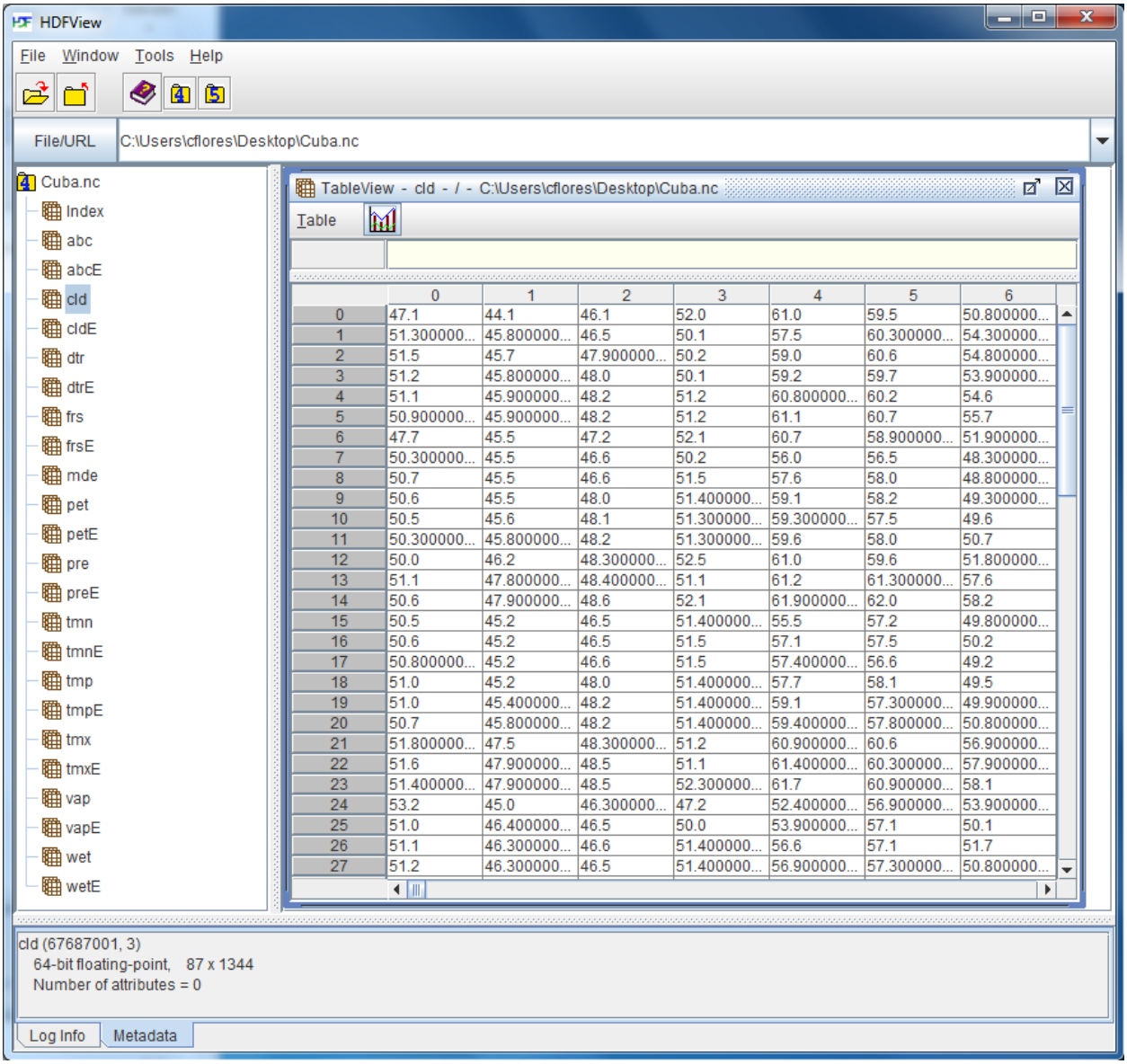


Figura 24: Variables del archivo NetCDF generado.

Calcular una nueva variable

El cálculo de nuevas variables puede ser de gran importancia para ver el comportamiento de una variable x que es resultado de la interacción de una o más variables. Para realizar este proceso se procede a ejecutar el algoritmo *HDFFieldCalculator.*

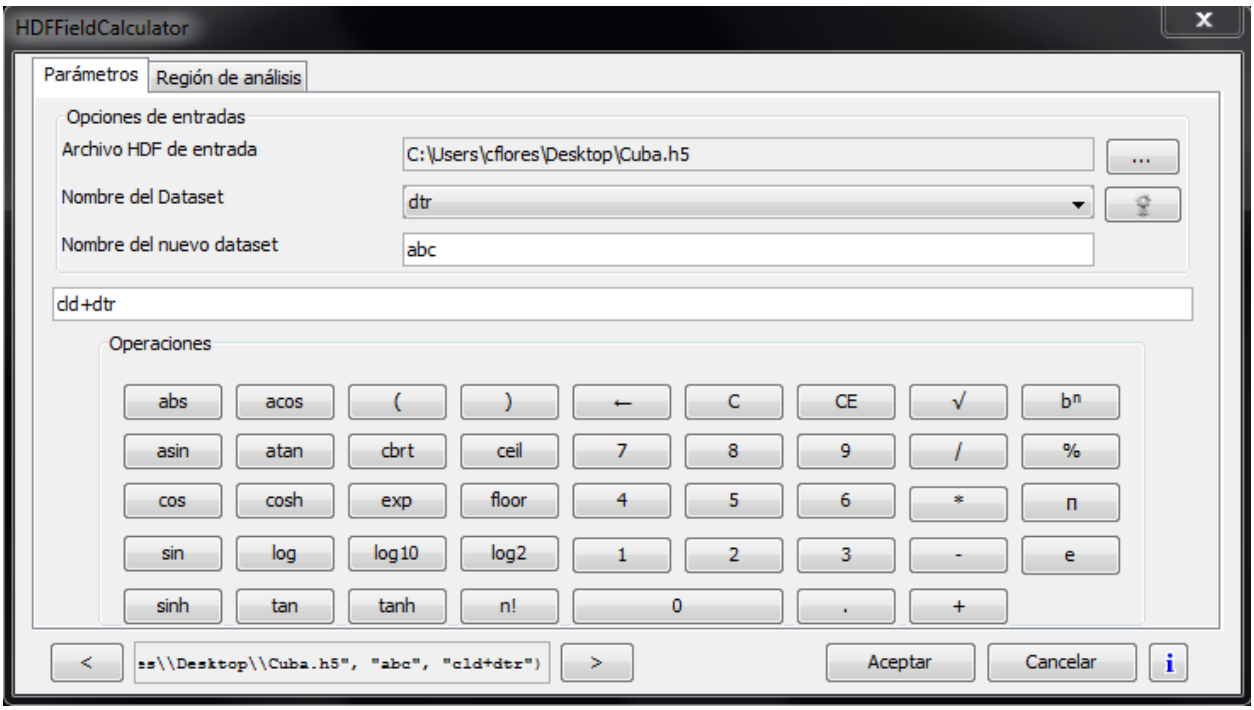


Figura 25: Parámetros de entrada para el algortimo.

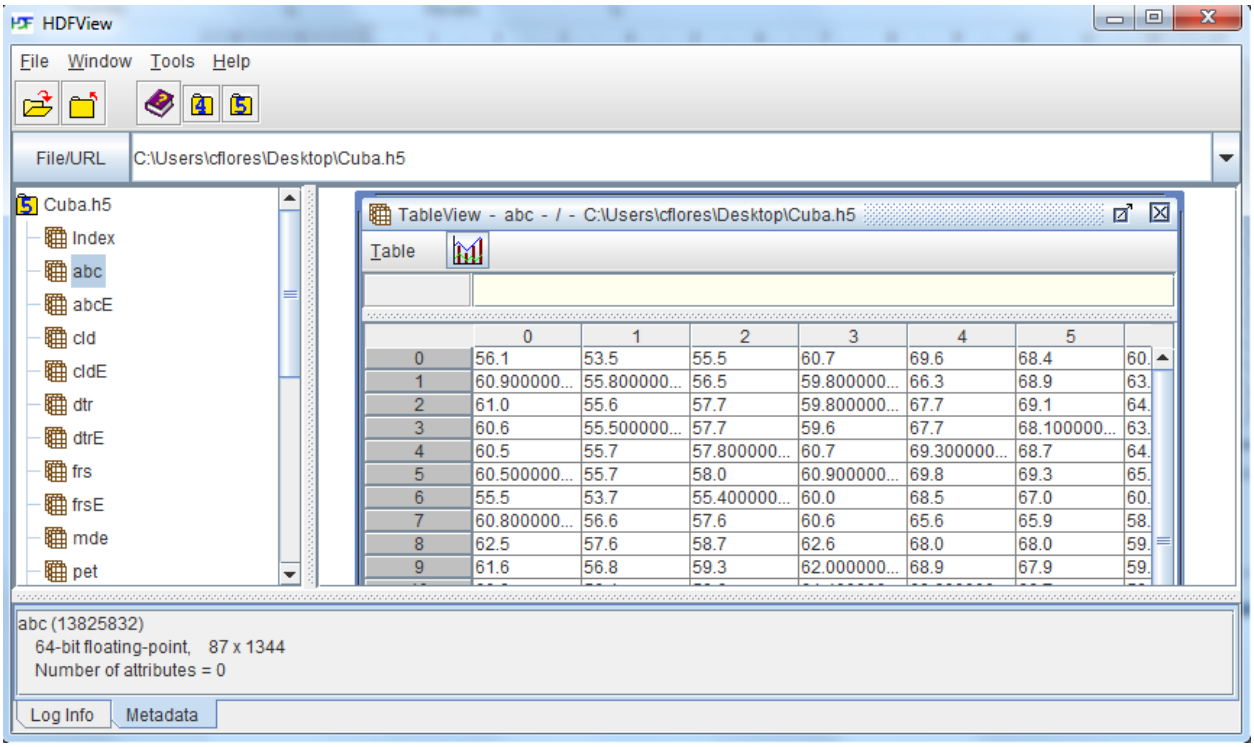


Figura 26: Fichero HDF con nueva variable creada.

Como se puedo apreciar en la Figura 25 la definición de los parámetros de entrada del algoritmo no se necesita definir un archivo de salida pues la nueva variable calculada se incorporará

automáticamente al archivo HDF de entrada, manteniendo las dimensiones de las variables con las que se calculó. La Figura 26 muestra el nuevo archivo HDF después de haberse incorporado la variable *abc* como resultado de sumar las variables *cld* y *dtr*.

3.3 Validación de los resultados

Durante el proceso de validación se verifica que el sistema producido cumpla con las especificaciones y que logra su cometido.

El archivo NetCDF resultante del proceso de conversión de HDF a NetCDF mantiene la misma cantidad de variables con todas sus dimensiones idénticas a las del archivo original. Mediante la **¡Error! No se encuentra el origen de la referencia.** y la **¡Error! No se encuentra el origen de la referencia.**donde se muestra la variable cld original y la resultante, se puede verificar que en el proceso de transformación del archivo se mantuvo el mismo tipo de datos, así como las dimensiones y los mismos valores para las variables.

La variable creada después de aplicar el algoritmo HDFFieldCalculator, cumple con lo esperado por el usuario.

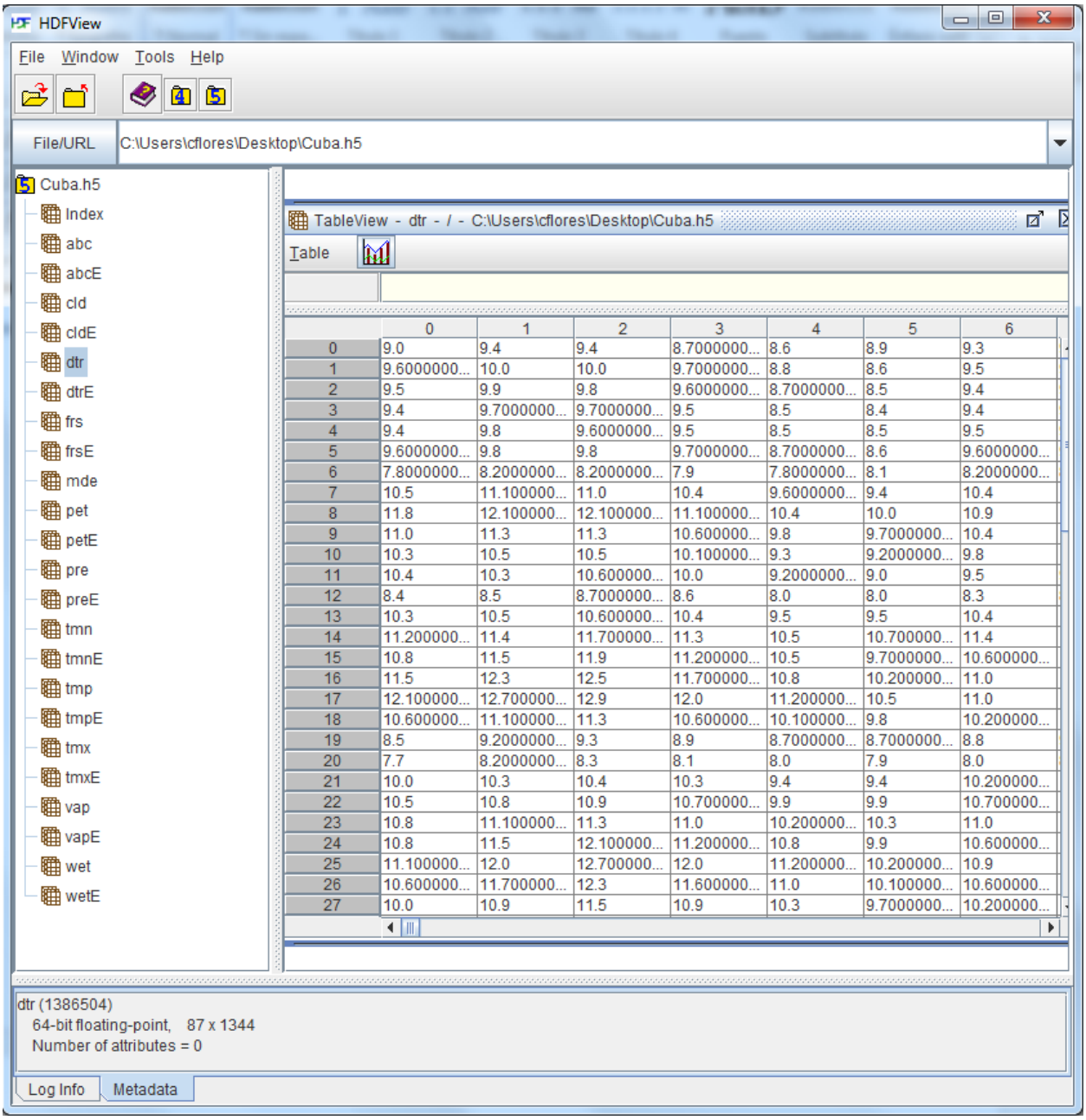


Figura 27: Estructura de la variable dtr creada.

Manteniendo 87 celdas que almacenan información, tomados en 1344 momentos de tiempos. La **¡Error! No se encuentra el origen de la referencia.**muestra los valores que posee la variable dtr y la **¡Error! No se encuentra el origen de la referencia.**los valores de la variable cld, a simple vista se

puede observar como los valores que muestra la Figura 26 se corresponden con el proceso de sumar las variables cld + dtr.

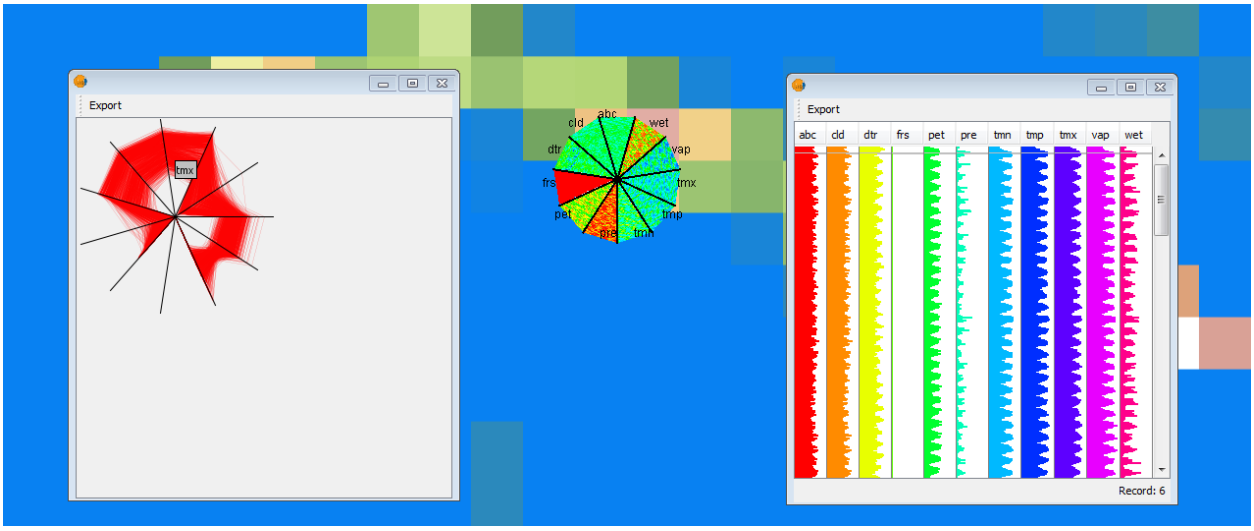


Figura 28: Fichero HDF visualizado con Tácnicas de Visualización científica.

La Figura 28 muestra el fichero Cuba.h5 visualizado en gvSIG y con técnicas de visualización científica aplicadas, donde se observa como la nueva variable creada dentro del archivo se integró totalmente y se representa muy bien.

3.4 Conclusiones Parciales

A lo largo de este capítulo se han probado diferentes algoritmos para el desarrollo del caso de estudio planteado, se verifica de esta manera la eficiencia y el correcto funcionamiento de los algoritmos utilizados. Este caso de estudio sirve como guía para la conversión de *datasets* más actualizados con la misma estructura descrita en este capítulo. Todos los archivos creados pueden ser objeto de estudio por parte de especialistas para la observación y la toma de decisiones respecto al clima.

Conclusiones

Como resultado de esta investigación se implementaron nuevas herramientas para la manipulación de cuantiosos volúmenes de datos espacio-temporales, cumpliendo de esta manera el objetivo propuesto y una serie de tareas que se describen a continuación:

- Se realizó un amplio estudio sobre los principales formatos de datos espacio-temporales utilizados en el área de las geociencias, se describieron y mostraron algunas de sus principales características, siendo seleccionados los archivos HDF, NetCDF y los datos representados en Excel como los más importantes para su trabajo en el módulo SEXTANTE de gvSIG.
- Se modificaron varios de los algoritmos con los que contaba el módulo, para lograr una mayor efectividad y funcionamiento del mismo.
- Se diseñaron e implementaron un conjunto de algoritmos, los cuales fueron incorporados a la biblioteca de análisis geoespacial SEXTANTE, se logra que los algoritmos implementados puedan ser utilizados desde cualquier SIG al que se le pueda incorporar la plataforma SEXTANTE. Se definió el esquema general de algunos de los algoritmos con los que cuenta el módulo resultante, lo que permite que este trabajo pueda ser ampliado para otros formatos de datos.
- Se presentó un caso de estudio donde se pudo apreciar el correcto funcionamiento de los algoritmos desarrollados.

Recomendaciones

Luego de concluir la presente investigación, solo resta hacer algunas recomendaciones tales como:

- Implementar otras funcionalidades que aumente el uso de esta herramienta por usuarios especializados.
- Incorporar nuevos formatos de datos espacio-temporales a la herramienta para que pueda usarse en otras áreas científicas.
- Probar la efectividad de los algoritmos implementados con otros conjuntos de datos en otro dominio de aplicación.

Bibliografía

BOLSTAD, P., 2005. GIS Fundamentals: A first text on Geographic Information Systems, White Bear Lake, Minnesota, Eider Press.

Borrell González, A., 2012. Aplicaci{ó}n inform{á}tica para la integraci{ó}n de los datos generados por el observatorio OBSEA en las redes de sistemes de observaci{ó}n.

Durango, C., 2013. Caracterización de Datos Espacio - Temporales en Sistemas de Información Geográfica.

Group, H.D.F. & others, 2011. HDF5 User's Guide.

GUTIERREZ PUEBLA, J. G., M., 1997. Sistemas de Información Geográfica.

Hankin, S.C. et al., 2010. NetCDF-CF-OPeNDAP: Standards for ocean data interoperability and object lessons for community data standards processes. In *Oceanobs 2009, Venice Convention Centre, 21-25 septembre 2009, Venise*.

Long, Q. et al., 2013. JAWAMix5: an out-of-core HDF5-based java implementation of whole-genome association studies using mixed models. *Bioinformatics*, 29(9), pp.1220–1222.

McGrath, R.E., 2003. XML and Scientific File Formats. In *2003 Seattle Annual Meeting*.

Murphy, L.D., 1995. Geographic information systems: are they decision support systems? In *System Sciences, 1995. Proceedings of the Twenty-Eighth Hawaii International Conference on*. pp. 131–140.

NASA, 2013. CDF User's Guide.

OGC, 2012. Open Geospatial Consortium. Available at: http://www.opengeospatial.org/.

Olaya, V., 2011. SEXTANTE User's manual (v1. 0). *Pdf online.[http//www1. unex. es/eweb/sext antegis/IntroductionToSEXTANTE. pdf.]*.

Olaya, V., 2009. Sistemas de informaci{ó}n geogr{á}fica. *Cuadernos Internacionales de Tecnolog{í}a para el Desarrollo Humano, 2009, n{ú}m. 8*.

Poinot, M., 2010. Five good reasons to use the hierarchical data format. *Computing in Science & Engineering*, 12(5), pp.84–90.

Ramachandran, R. et al., 2004. Earth Science Markup Language (ESML):: a solution for scientific data-application interoperability problem. *Computers & Geosciences*, 30(1), pp.117–124.

Rew, R. et al., 2011. The NetCDF users guide-data model, programming interfaces, and format for self-describing, portable data-NetCDF version 4. *Unidata Program Center*.

Roque, D.C., 2014. Herramientas para la manipulacion de formatos de datos cientificos en SIG.

STAR, J. & ESTES, J., 1990. Geographic information systems, prentice-Hall Englewood Cliffs.

TOMLIN, D.C., 1990. Geographic information systems and cartographic modeling.

Ullman, R. & Denning, M., 2012. HDF5 for NPP sensor and environmental data records. In *2012 IEEE International Geoscience and Remote Sensing Symposium*. pp. 1100–1103.

UNIDATA, 2012. No Title. Available at: http://www.unidata.ucar.edu.

Vázquez Rodríguez, R., 2015. *Nuevos m{é}todos para el procesamiento y an{á}lisis de informaci{ó}n geogr{á}fica* Universidad de Granada. Tesis Doctorales, ed., Granada: Universidad de Granada. Available at: http://hdl.handle.net/10481/41303.

W3C, 2012. W3C Standards. Available at: http://www.w3.org.

Yang, W. & Di, L., 2004. An accurate and automated approach to georectification of HDF-EOS swath data. *Photogrammetric Engineering & Remote Sensing*, 70(4), pp.397–404.

Yeh, P.-S. et al., 2002. Implementation of CCSDS Lossless Data Compression in HDF.

Bibliografía